RECURSIVE PARAMETER ESTIMATION FOR CONTINUOUS-TIME BIVARIATE MARKOV CHAINS

by

Zhuxuan Li

A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Electrical and Computer Engineering

Committee:

_____	Dr. Yariv Ephraim, Thesis Director
_____	Dr. Brian L. Mark, Committee Member
_____	Dr. Shih-Chun Chang, Committee Member
_____	Dr. Andre Manitius, Department Chair
_____	Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering

Date: 05/12/2014

Spring Semester 2014
George Mason University
Fairfax, VA

Recursive Parameter Estimation for Continuous-time Bivariate Markov Chains

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Zhuxuan Li
Bachelor of Science
Tsinghua University, 2011

Director: Dr. Yariv Ephraim, Professor
Department of Electrical and Computer Engineering

Summer Semester 2014
George Mason University
Fairfax, VA

Dedication

I dedicate this thesis to my parents Jiquan and Qiusha, and to my husband, Yuchen.

Acknowledgments

I owe an enormous debt of gratitude to my thesis advisor Professor Yariv Ephraim. Without him, the thesis would not have been possible. He was always generous and patient to give me advice for my research during the last two years with his rich professional knowledge. I am really grateful for his help in revising my thesis, he checked my thesis word by word, and pointed out every tiny mistake I have made. He also gave me insightful advice and encouraged me with my study. His gentle personality and rigorous attitude toward the research will benefit me in my future career and my personal life.

I would like to thank my committee members Professor Brian L. Mark and Professor Shih-Chun Chang for reading my thesis and for their helpful comments.

I would like to express my appreciation to my academic advisor Professor Andre Manitius for his continuous support and encouragement throughout my studies at GMU. I would also like to thank the faculty and staff of the Electrical and Computer Engineering department for their generous help.

This work was supported in part by the U.S. National Science Foundation under Grant CCF-0916568. I thank NSF for their generous support.

I would also like to thank my colleagues, Yuandao Sun, Neshat Etemadi-Rad and Thao Nguyen, who generously gave me a lot of help. Without their understanding, help and encouragement, I would not be able to finish my thesis.

Table of Contents

Page

List of Tables vii

List of Figures viii

Abstract x

1 Introduction 1

1.1 Overview of bivariate Markov chains 1

1.2 Overview of the algorithms 3

1.3 Organization 4

2 Literature Review 6

2.1 Continuous-Time Markov Modulated Poisson Process 6

2.2 Continuous-Time Bivariate Markov Chains 7

3 Estimation Algorithms for Markov Modulated Poisson Process 12

3.1 Likelihood Function 12

3.2 EM Algorithm 14

3.2.1 Forward-Backward recursions 17

3.2.2 Implementation of EM Algorithm 19

3.3 Recursive Algorithm 19

3.4 Numerical Results 23

3.4.1 Numerical Results of EM Algorithm 23

3.4.2 Numerical Results of Recursive Algorithm 25

3.4.3 Summary of Numerical Results 27

4 Estimation Algorithms for Bivariate Markov Chains 32

4.1 Likelihood Function [1] 32

4.2 EM Algorithm 34

4.2.1 Forward-Backward Recursions 35

4.2.2 Implementation of EM algorithm 37

4.3 Recursive Algorithm 38

4.3.1 Forward recursion 39

4.3.2 Implementation of Recursive Algorithm 41

4.4 Numerical Results 43

4.4.1 Numerical Results for EM Algorithm 43
4.4.2 Numerical Results for Recursive Algorithm 54
4.4.3 Summary of Numerical Results 54
5 Conclusion and Future Directions 72
5.1 Conclusion 72
5.2 Future Directions 73
5.2.1 Increase Number of States 73
5.2.2 Estimation with Actual Data 74
Bibliography 75

List of Tables

Table		Page
3.1	Bias and MSE in EM estimating an MMPP of case A	24
3.2	Bias and MSE in EM estimating an MMPP of case B	24
3.3	Bias and MSE in EM estimating an MMPP of case C	24
3.4	Bias and MSE in EM estimating an MMPP of case D	25
3.5	Bias and MSE in recursive estimating an MMPP of case A	27
3.6	Bias and MSE in recursive estimating an MMPP of case B	28
3.7	Bias and MSE in recursive estimating an MMPP of case C	28
3.8	Bias and MSE in recursive estimating an MMPP of case D	29
3.9	Averaged result of bias and MSE in recursive estimating an MMPP of case A	29
3.10	Averaged result of bias and MSE in recursive estimating an MMPP of case B	30
3.11	Averaged result of bias and MSE in recursive estimating an MMPP of case C	30
3.12	Averaged result of bias and MSE in recursive estimating an MMPP of case D	31
4.1	Bias and MSE in EM estimating a bivariate Markov chain of case A	45
4.2	Bias and MSE in EM estimating a bivariate Markov chain of case B	45
4.3	Bias and MSE in EM estimating a bivariate Markov chain of case C	45
4.4	Bias and MSE in EM estimating a bivariate Markov chain of case D	54
4.5	Bias and MSE in recursive estimating a bivariate Markov chain of case A	55
4.6	Bias and MSE in recursive estimating a bivariate Markov chain of case B	56
4.7	Bias and MSE in recursive estimating a bivariate Markov chain of case C	57
4.8	Bias and MSE in recursive estimating a bivariate Markov chain of case D	58
4.9	Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case A	59
4.10	Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case B	60
4.11	Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case C	61
4.12	Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case D	62

List of Figures

Figure		Page
1.1	Example of hidden Markov model	2
1.2	An example model of Markov modulated Poisson process	3
2.1	Example of Markov modulated Poisson process	7
3.1	Example of Markov modulated Poisson process with notations	13
4.1	Estimation result versus number of iterations in EM estimating of case A part one	46
4.2	Estimation result versus number of iterations in EM estimating of case A part two	47
4.3	Estimation result versus number of iterations in EM estimating of case B part one	48
4.4	Estimation result versus number of iterations in EM estimating of case B part two	49
4.5	Estimation result versus number of iterations in EM estimating of case C part one	50
4.6	Estimation result versus number of iterations in EM estimating of case C part two	51
4.7	Estimation result versus number of iterations in EM estimating of case D part one	52
4.8	Estimation result versus number of iterations in EM estimating of case D part two	53
4.9	Estimation result versus number of blocks in recursive estimating of case A part one	63
4.10	Estimation result versus number of blocks in recursive estimating of case A part two	64
4.11	Estimation result versus number of blocks in recursive estimating of case B part one	65
4.12	Estimation result versus number of blocks in recursive estimating of case B part two	66

4.13 Estimation result versus number of blocks in recursive estimating of case C part one 67
4.14 Estimation result versus number of blocks in recursive estimating of case C part two 68
4.15 Estimation result versus number of blocks in recursive estimating of case D part one 69
4.16 Estimation result versus number of blocks in recursive estimating of case D part two 70

Abstract

RECURSIVE PARAMETER ESTIMATION FOR CONTINUOUS-TIME BIVARIATE MARKOV CHAINS

Zhuxuan Li

George Mason University, 2014

Thesis Director: Dr. Yariv Ephraim

A continuous-time bivariate Markov chain comprises a pair of continuous-time random processes which are jointly Markov. One of the two processes is an underlying process while the other is assumed observable. An important special bivariate Markov chain is given by the continuous-time Markov modulated Poisson process (MMPP). The underlying process of an MMPP is a Markov chain, and the observable process is conditionally Poisson. Discrete-time bivariate Markov chains may also be defined, but they shall not be studied in this thesis. Bivariate Markov chains are useful in modeling ion-channel currents in living cells, Internet traffic, and in other problems in queuing theory. In this thesis we focus on recursive estimation of the parameter of a bivariate Markov chain which comprises its infinitesimal generator. We study a stochastic approximation approach using a newly developed recursion for the gradient of the log-likelihood function of the observed signal. The recursive algorithm is compared with a batch expectation-maximization (EM) algorithm developed earlier. The bias and mean squared error obtained in estimating each component of the parameter using each algorithm are evaluated and compared. The recursive algorithm requires far more data to provide an estimate comparable to that obtained by the EM algorithm, but the EM algorithm iterates over the entire data multiple times. The main advantage of the

recursive estimator is its ability to adapt to slow changes in the underlying statistics of the model. The EM algorithm is a batch approach which must be re-applied whenever new data becomes available or there is a change in the underlying statistics of the model.

Chapter 1: Introduction

1.1 Overview of bivariate Markov chains

A bivariate Markov chain is composed of two random processes, one of which is observable and the other is an underlying process. The pair of random processes is jointly Markov, but neither of them is required to be a Markov process.

There are several important examples of bivariate Markov chains. The most familiar one might be the hidden Markov model (HMM), which is detailed in [2], [3], [4]. A hidden Markov model is a discrete-time random process where its underlying process is a discrete-time finite Markov chain, and its observable process is a sequence of random variables such that each only depends on the state of the underlying Markov process at that time. For example, suppose that there are two kinds of weather conditions, sunny or rainy, and I have two choices, to go to the lab or stay home. The two kinds of weather conditions may be represented by two states of a Markov chain. As an example, suppose that the transition probabilities of the Markov chain are specified as follows: Given that today is sunny, the probability of tomorrow being sunny is 0.7 and the probability of tomorrow being rainy is 0.3. Given that today is rainy, the probability of tomorrow being sunny is 0.5 and the probability of tomorrow being rainy is 0.5. To complete the model, we specify the probabilities of my choices given the current state of the Markov chain. Given that today is rainy, the probability that I go to the lab is 0.1 and the probability that I stay home is 0.9. Given that today is sunny, the probability that I go to the lab is 0.8 and the probability that I stay home is 0.2. The hidden Markov model is illustrated in Figure 1.1. Suppose that the observer is my friend who lives in another city, and have no idea of the weather condition of my city nor the probabilities mentioned above, i.e. the parameters of the hidden Markov model, but each day I will tell the observer either I go to the lab or stay

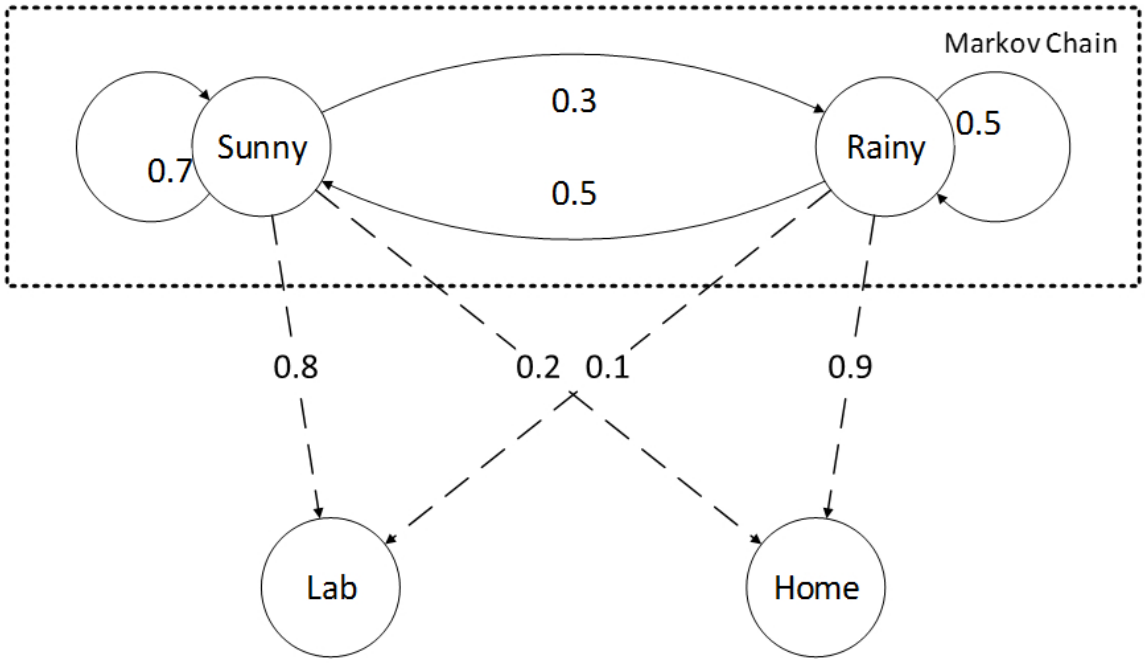

Figure 1.1: Example of hidden Markov model

home. In the view of the observer, this is a hidden Markov model. After a long time, e.g. one year, the observer could estimate the parameters only based on the data of what I did everyday.

As demonstrated before, there is no special limit on the two processes except that the two together should be jointly Markov. Each of them could be a continuous-time process or discrete-time process, have finite states, countably infinite states or uncountably infinite states. The kind of bivariate Markov chains we are going to talk about in this thesis are composed of a pair of continuous-time processes with finite states. We just refer to them as continuous-time bivariate Markov chains or just as bivariate Markov chains. A familiar bivariate Markov chain is the Markov modulated Poisson process (MMPP) which is illustrated in [5], [6] and [7]. For the MMPP, the underlying process is a continuous-time finite state Markov chain, and the observable process is a conditionally Poisson process with the rate of each event depends only on the current state of the underlying Markov chain. Figure 1.2 shows an example of MMPP. There are many other kinds of continuous-time

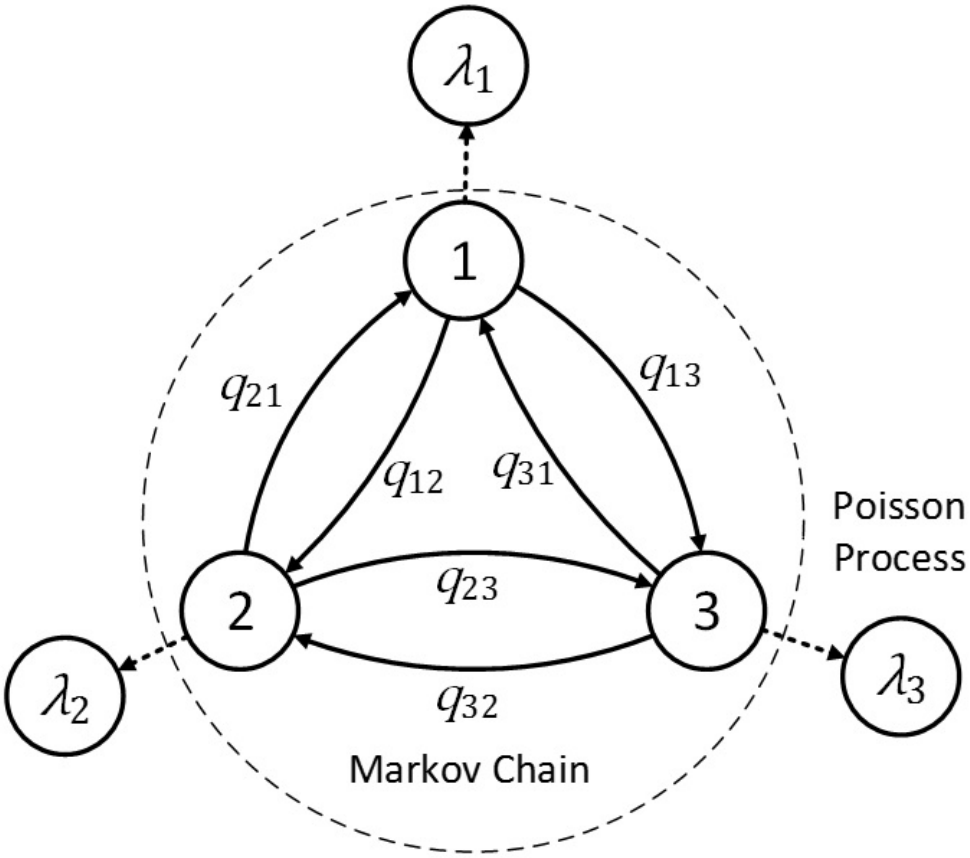

Figure 1.2: An example model of Markov modulated Poisson process

bivariate Markov chains like aggregated Markov chains, Markov modulated Markov process (MMMP), etc. We are going to further expound the MMPP and general continuous-time bivariate Markov chains in the following chapters.

1.2 Overview of the algorithms

The two algorithms we use in this thesis is expectation-maximization (EM) algorithm and a recursive parameter algorithm which is based on a stochastic approximation. The EM algorithm was first applied to the MMPP by Rydén [8], and first applied to general continuous-time bivariate Markov chains in [1]. The EM algorithm is an off-line iterative estimation algorithm. After each iteration, the EM algorithm obtains a new estimate with the same

or a higher likelihood than the previous estimate.

A recursive parameter algorithm, which we refer to it as recursive algorithm in the following, was first developed for the MMPP by Lindgren and Holst in [9] and [10, Eq. 16], and first developed for general continuous-time bivariate Markov chains in [11]. The recursive algorithm is an on-line estimation algorithm. We do not estimate based on all the data like what we did in the EM algorithm, we rather receive part of the data at each time, let us call it a block of data. The estimation is only based on the current block of data and previous estimates.

If we use the former hidden Markov model example to explain the basic idea of the two algorithms, then the EM algorithm would be based on the entire data record of what I did each day and send the records to the observer after a year, while the recursive algorithm would be based on what I tell the observer I did every day or every few days, and the observer revises his or her estimates based on the new record I told him or her each time. Comparing the EM algorithm with the recursive algorithm, the EM algorithm requires much larger memory but could get a more accurate estimate, while the recursive algorithm requires smaller amount of memory and could perform real-time estimation.

1.3 Organization

The reminder of this thesis is organized as follows. In chapter 2, we present a literature review of the two random processes, i.e. the MMPP and the bivariate Markov chain. In the first section of chapter 2, we discuss the MMPP, define the associated notations and specify its parameter. We present an example of an MMPP to help demonstrate these notations. In the second section, we discuss the bivariate Markov chain, and its parameter.

In chapter 3, we show how to apply the EM algorithm and recursive algorithm to the MMPP. In the first section of chapter 3, we introduce the likelihood function of the MMPP which plays an important role in the two algorithms. And we introduce a detailed MMPP example. In the second section, we elaborate on the estimation procedure of the EM algorithm for the MMPP. We introduce the forward-backward recursions and show how to

implement the EM algorithm using the forward-backward recursions. In the third section, we present the recursive algorithm for the MMPP. In the fourth section, we provide some numerical results for the two algorithms.

In chapter 4, we present the EM algorithm and the recursive algorithm for the bivariate Markov chain, as well as numerical results. In the first section of chapter 4, we introduce the likelihood function of the bivariate Markov chain. In the second section, we present the EM algorithm with forward-backward recursions. In the third section, we talk about the estimation procedure of the recursive algorithm for the bivariate Markov chain with forward recursion. Numerical results of the EM algorithm and the recursive algorithm are provided in the fourth section.

In chapter 5, we summarize the contribution of this thesis and point to possible future research directions.

Chapter 2: Literature Review

In this chapter, we review the definition and basic theories of Markov modulated Poisson process and bivariate Markov chains.

2.1 Continuous-Time Markov Modulated Poisson Process

Markov modulated Poisson process (MMPP) has a hidden Markov model representation. It is composed of two random processes, one is a Markov chain which is the underlying process, and the other is a conditionally Poisson process which is the observable process. The parameter of this conditionally Poisson process is determined by the Markov process. Suppose $Z = \{Z(t), t \geqslant 0\}$ is a continuous-time Markov modulated Poisson process, and it is composed of a underlying process S and a observable process X, i.e. $Z = \{X, S\}$. Suppose the continuous time Markov chain $\{S(t), t \geqslant 0\}$ has r states, $\{1, ..., r\}$, and $Q = \{q_{ij}, i, j = 1, ..., r\}$ is the generator of S. Let's denote q_{ii} by $-q_i$, which represents the total transition rate from state i [12, p. 306], [13, p. 362]. Suppose the rates of the conditionally Poisson process corresponding to the r states $\{X(t), t \geqslant 0\}$ are $\Lambda = \text{diag}\ \{\lambda_1, ..., \lambda_r\}$. The parameter space is composed of Q and Λ, $\phi = \{(Q, \Lambda)\}$.

Suppose $T_k, k = 1, 2, ...$ is the time of the kth Poisson event, and assume $T_0 = 0$. Let Y_k denote the dwell time between two Poisson events, i.e. $Y_k = T_k - T_{k-1}$, for $k = 1, 2, ...$ Figure 2.1 gives an example of a MMPP whose underlying Markov chain has three states, i.e. $r = 3$. Here, y_k denotes a realization of Y_k, and t_k denotes a realization of T_k.

To simplify the problem, let the zeroth event happens at time $t = 0$, and the first event comes after that time. At this time, if we consider the underlying process and observable process together, we can get a Markov renewal process [14, Proposition 1.11] $\{S(T_k), Y_k\}_{k=1}^{\infty}$, which applies to the Markov modulated Poisson process. Its transition probability matrix,

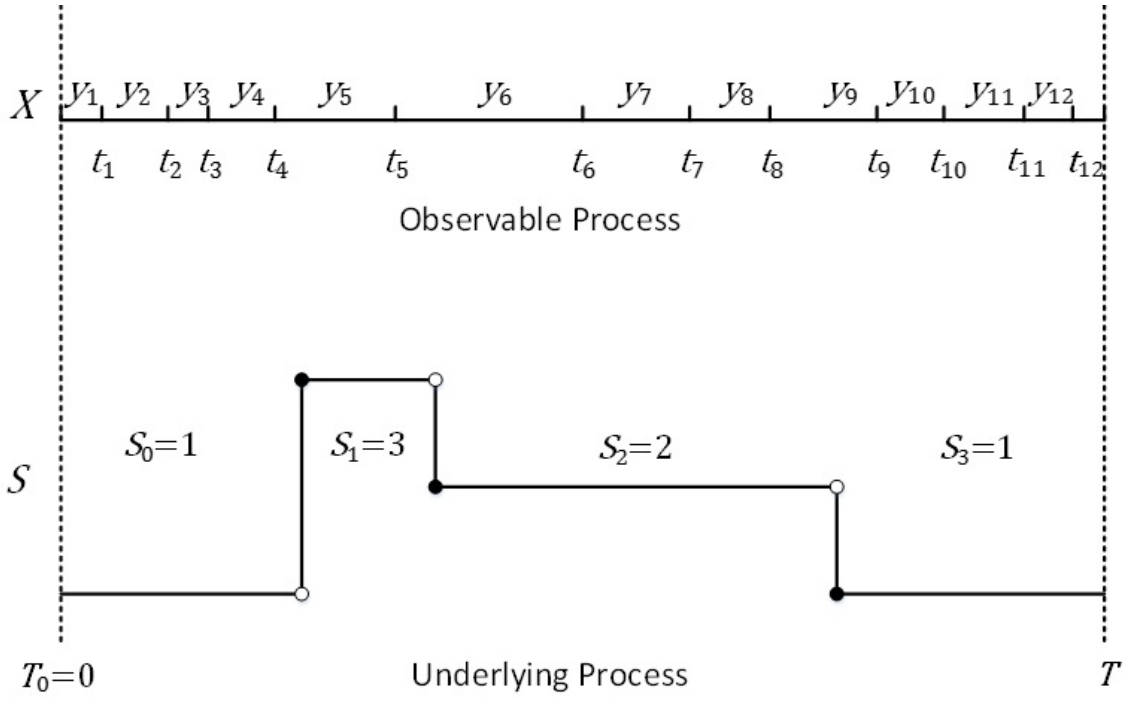

Figure 2.1: Example of Markov modulated Poisson process

as shown in [15], is $F(y) = \{F_{ij}(y)\}$, where

$$F_{ij}(y) = P(Y_k \leqslant y, S(t_{k-1} + y) = j \mid S(t_{k-1}) = i).$$

With this transition probability matrix, we can find the transition density matrix $f(y)$ which is the derivative of $F_{ij}(y)$ with respect to y. This is given by [16].

$$f(y) = e^{(Q-\Lambda)y}\Lambda. \tag{2.1}$$

2.2 Continuous-Time Bivariate Markov Chains

A bivariate Markov chain is composed of two processes which are jointly Markov, one is the underlying process, and the other is the observable process. Suppose $Z = \{Z(t), t \geqslant 0\}$ is a continuous-time bivariate Markov chain, and it is composed of X and S, X is the observable

process, and S is the underlying process, i.e. $Z = \{X, S\} = \{(X(t), S(t)), t \geqslant 0\}$. Each process alone, i.e., X or S, is not necessarily Markov, but together (X, S) is Markov. Let the process $X = \{X(t), t \geqslant 0\}$ have d states such that the state space is $\mathbb{X} = \{1, ..., d\}$. Suppose $S = \{S(t), t \geqslant 0\}$ has r states, the state space is $\mathbb{S} = \{1, ..., r\}$. We could consider a bivariate Markov process as a rd-state Markov chain with part of its parameters hidden. Then the state space of the bivariate Markov chain Z is $\mathbb{Z} = \mathbb{X} \times \mathbb{S}$. The process Z is a continuous-time process that could jump from one state to another with finite number of jumps in any finite interval [1]. We can assume that the transition matrix P_t at time 0 is $P_0 = I$. I is the identity matrix, and P_t is continuous at $t = 0$.

$$\lim_{t \to 0} P_t = I. \tag{2.2}$$

P_t is differentiable at $t = 0$ [17, Propositions 8.3.2-8.3.3]. The generator matrix G is the derivative of the transition matrix P_t at $t = 0$.

$$G = \lim_{t \to 0} \frac{1}{t}(P_t - I). \tag{2.3}$$

So we could use the generator matrix G to represents the rates of those jumps to interpret this bivariate Markov chain Z. Let

$$G = \{g_{ab}(ij), a, b \in \mathbb{X}; i, j \in \mathbb{S}\}. \tag{2.4}$$

Suppose that at time $t = 0$, Z could be at a joint state $\{(a, i)\}$, i.e. $Z(0) = (a, i)$. The probability of $Z(t) = (b, j)$ for a sufficiently small t is [18, p. 333]

$$\begin{aligned} &P(Z(t) = (b, j) \mid Z(0) = (a, i)) \\ &= \begin{cases} g_{ab}(ij)t + o(t), & (a, i) \neq (b, j) \\ 1 + g_{aa}(ii)t + o(t), & (a, i) = (b, j), \end{cases} \end{aligned} \tag{2.5}$$

where $g_{aa}(ii) = -\sum_{(b,j)\neq(a,i)} g_{ab}(ij)$. The properties of the generator matrix are as follows: [11]

(i) $-\infty \leqslant g_{aa}(ii) \leqslant 0$,

(ii) $0 \leqslant g_{ab}(ij) < \infty$ whenever $(a, i) \neq (b, j)$,

(iii) $\sum_{b,j} g_{ab}(ij) \leqslant 0 for all (a, i) \in \mathbb{Z}$ with equality if

$$\sup_{(a,i)}\{-g_{aa}(ii)\} \leqslant \infty. \tag{2.6}$$

From [11], [19, p. 267]

$$P_t = e^{Gt}. \tag{2.7}$$

For general bivariate Markov chain, define for any $(a, i), (b, j) \in \mathbb{X}$, the transition probability function is

$$F_{ij}^{ab}(t) = P(Z_1 = (b, j), \Delta T_1 \leqslant t | Z_0 = (a, i)). \tag{2.8}$$

Because bivariate Markov chain is homogeneous, then for all k,

$$F_{ij}^{ab}(t) = P(Z_{k+1} = (b, j), \Delta T_{k+1} \leqslant t | Z_k = (a, i)). \tag{2.9}$$

The transition density function $f_{ij}^{ab}(t)$ is the derivative of $F_{ij}^{ab}(t)$ with respect to t. Define

the transition density matrix of the transition of X from a to b that composed by $f_{ij}^{ab}(t)$ is

$$f^{ab}(t) = \{f_{ij}^{ab}(t), i \in \mathbb{S}_a, j \in \mathbb{S}_b\}. \tag{2.10}$$

Then define the transition probability as

$$\bar{F}_{ij}^{a}(t) = P(S(t) = j, \Delta T_1 > t | X(0) = a, S(0) = i). \tag{2.11}$$

Its transition matrix is

$$\bar{F}^{a}(t) = \{\bar{F}_{ij}^{a}(t), i, j \in \mathbb{S}_a\}. \tag{2.12}$$

Their relationships with generator matrix G are as follows. For $t \geqslant 0$, from [20], [21], [1]

$$f^{ab}(t) = e^{G_{aa}t} G_{ab}, \quad a \neq b, \tag{2.13}$$

$$\bar{F}^{a}(t) = e^{G_{aa}t}. \tag{2.14}$$

The MMPP introduced earlier is a special kind of bivariate Markov chain. Its generator is given by

$$G = \begin{pmatrix} Q-\Lambda & \Lambda & & & & \\ & Q-\Lambda & \Lambda & & & \\ & & Q-\Lambda & \Lambda & & \\ & & & Q-\Lambda & \Lambda & \\ & & & & \ddots & \end{pmatrix} \tag{2.15}$$

The generator matrix above is a infinite block Toeplitz matrix [11]. For example, the

generator of a MMPP with modulo-2 counting of events is [22]

$$G = \begin{pmatrix} Q - \Lambda & \Lambda \\ \Lambda & Q - \Lambda \end{pmatrix} \tag{2.16}$$

$G_{12} = G_{21}$ is because the transition rate of X from 1 to 2 and from 2 to 1 are symmetric. And from the above we know that $G_{11} = G_{22}$.

Chapter 3: Estimation Algorithms for Markov Modulated Poisson Process

In this chapter, we illustrate the EM algorithm and recursive algorithm for the MMPP. Numerical results are also provided.

3.1 Likelihood Function

The likelihood function is the most important part in any of the two following algorithms. We are going to maximize the likelihood function to find the parameters. Using the terms introduced in chapter 2, the inter-arrival time of the observable sequence is $y_1, ..., y_n$, a realization of $Y_1, ..., Y_n$. Suppose the observable process has n jumps during $[0, T]$. The likelihood function with parameter ϕ is

$$p_{Y_1,...,Y_n}(y_1, ..., y_n; \phi) = \pi(Q, \Lambda) \left\{ \prod_{i=1}^{n} f(y_i; \phi) \right\} \mathbf{1}, \tag{3.1}$$

where $\mathbf{1}$ is an $r \times 1$ column vector of ones, π is the distribution of S_0 that makes $\{S_{k-1}, Y_k\}$ stationary with parameter (Q, Λ) [8]. To simplify the notation, let $p(y^n; \phi)$ denote the likelihood function above. The likelihood function (3.1) is the one of the observable process $y_1, ..., y_n$. We also need the likelihood function of the complete statistics $s_1, ..., s_n, y_1, ..., y_n$, which is denoted by $p(y, s; \Phi)$. To further extend the likelihood function introduced above, we need to define more terms. Like we introduced before, t_k is a realization of the time of kth jump of the observable process. Assume $t_0 = 0$, and that the time of last jump is $T = t_n$. Let the jump times of the underlying process S be denoted by u_k, i.e. $0 < u_1 < u_2 < \cdots < u_m < T$. And to simplify the problem, suppose $u_0 = 0$, $u_{m+1} = T$, and let

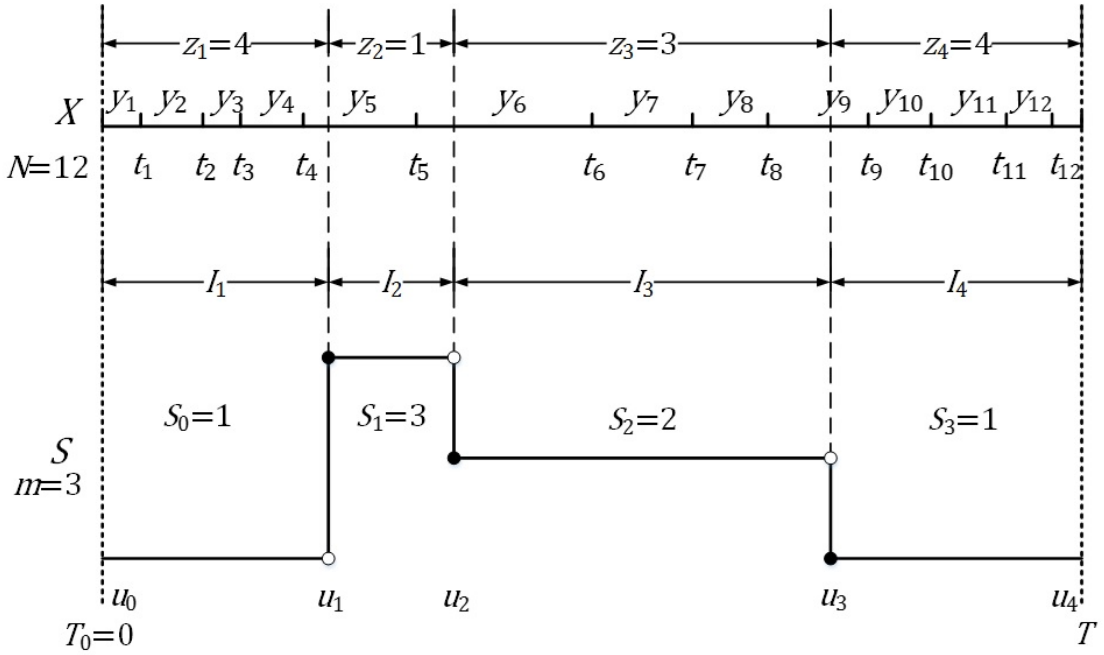

Figure 3.1: Example of Markov modulated Poisson process with notations

$I_k = [u_{k-1}, u_k)$ for $1 \leqslant k \leqslant m+1$. The length of I_k is denoted by $\Delta_{u_k} = u_k - u_{k-1}$. The state of S during time I_k, $S(u_{k-1})$, is denoted by s_k. Let z_k denote the number of jumps of the observable process X during I_k. $N(t)$ is the number of jumps of the observable process until time t, not counting the one at time $t_0 = 0$, i.e. $N(t) = \#\{k : 0 < k \leqslant n, t_k \leqslant t\}$. If we put these notations on Figure 2.1, it will become Figure 3.1.

The complete likelihood function $p(y, s; \phi)$ is given by [8]

$$\begin{aligned} p(y, s; \Phi) = \pi_{s_1} &\left\{ \prod_{k=1}^{m} q_{s_k} e^{-q_{s_k}\Delta u_k} \times \frac{q_{s_k, s_{k+1}}}{q_{s_k}} \right\} e^{-q_{s_{m+1}}\Delta u_{m+1}} \\ &\times \left\{ \prod_{k=1}^{m+1} \frac{(\lambda_{s_k}\Delta u_k)^{z_k}}{z_k!} e^{-\lambda_{s_k}\Delta u_k} \times \frac{z_k!}{(\Delta u_k)^{z_k}} \right\}. \end{aligned} \tag{3.2}$$

3.2 EM Algorithm

The EM (expectation-maximization) algorithm we are using here was introduced in [15], [11] and [8]. This algorithm is also used in parameter estimation for hidden Markov models. We are going to use this algorithm to maximize the likelihood function to estimate the parameter of the MMPP.

The above likelihood function is very complex and hard to deal with, so we need to simplify it. Before we present the simplified form, we need to define some notations. Let m_{ij} denotes the number of jumps of the process S from state i to state j during $[0, T]$ when $i \neq j$, i.e.

$$\begin{aligned} m_{ij} &= \#\{k : 1 \leqslant k \leqslant m, s_k = i, s_{k+1} = j\} \\ &= \#\{t : 0 < t \leqslant T, S(t-) = i, S(t) = j\}. \end{aligned} \tag{3.3}$$

Let T_i be the total dwell time of $\{S_t = i\}$ during time $[0, T]$, i.e.

$$T_i = \sum_{\{k:1 \leqslant k \leqslant m+1, s_k = i\}} \Delta u_k = \int_0^T I\{S(t) = i\} dt, \tag{3.4}$$

where I is indicator function. Let n_i be the number of jumps of the process X when $\{S_t\}$ remains on state i, i.e.

$$n_i = \sum_{\{k:1 \leqslant k \leqslant m+1, s_k = i\}} z_k = \sum_{k=1}^{n} I\{S(t_k) = i\}. \tag{3.5}$$

If we take logarithm of the likelihood function and simplify it using the above terms, we

obtain [8]

$$\log p(y,s;\phi) = \sum_{i=1}^{r} I\{S(0)=i\}\log\pi_i - \sum_{i=1}^{r} T_i q_i + \sum_{i=1}^{r}\sum_{\substack{j=1\\ j\neq i}}^{r} m_{ij}\log q_{ij} + \sum_{i=1}^{r}(n_i\log\lambda_i - \lambda_i T_i). \tag{3.6}$$

Assuming we have an estimate $\hat{\phi}_l = (Q^l, \Lambda^l)$ at the lth iteration, we will get new estimates $\hat{\pi}_i, \hat{m}_{ij}, \hat{T}_i$ and $\hat{n}_i$ corresponding to π_i, m_{ij}, T_i and n_i under the given $\hat{\phi}_l$ as follows:

$$\hat{\pi}_i = P_{\hat{\phi}_l}\{S(0)=i|N(u), 0\leqslant u\leqslant T\}, \tag{3.7}$$

$$\begin{aligned}\hat{m}_{ij} &= E_{\hat{\phi}_l}[m_{ij}|N(u), 0\leqslant u\leqslant T] \\ &= \int_0^T P_{\hat{\phi}_l}\{S(t-)=i, S(t)=j|N(u), 0\leqslant u\leqslant T\}dt,\end{aligned} \tag{3.8}$$

$$\hat{T}_i = E_{\hat{\phi}_l}[T_i|N(u), 0\leqslant u\leqslant T] = \int_0^T P_{\hat{\phi}_l}\{S(t)=i|N(u), 0\leqslant u\leqslant T\}dt, \tag{3.9}$$

$$\hat{n}_i = E_{\hat{\phi}_l}[n_i|N(u), 0\leqslant u\leqslant T] = \sum_0^T P_{\hat{\phi}_l}\{S(t_k)=i|N(u), 0\leqslant u\leqslant T\}. \tag{3.10}$$

As explained in [8], $\hat{m}_{ij}$, $\hat{T}_i$ and $\hat{n}_i$ are calculated as follows:

$$\begin{aligned}\hat{m}_{ij} &= E_{\hat{\phi}_l}[m_{ij}|N(u), 0\leqslant u\leqslant T] = \frac{q_{ij}^0}{P_{\hat{\phi}_l}\{N(u), 0\leqslant u\leqslant T\}} \\ &\times \int_0^T P_{\hat{\phi}_l}\{N(u), 0\leqslant u<t, S(t-)=i\}P_{\hat{\phi}_l}\{N(u), t\leqslant u\leqslant T|S(t)=j\}dt,\end{aligned} \tag{3.11}$$

$$\hat{T}_i = E_{\hat{\phi}_l}[T_i | N(u), 0 \leqslant u \leqslant T] = \frac{1}{P_{\hat{\phi}_l}\{N(u), 0 \leqslant u \leqslant T\}} \times \int_0^T P_{\hat{\phi}_l}\{N(u), 0 \leqslant u < t, S(t) = i\} P_{\hat{\phi}_l}\{N(u), t \leqslant u \leqslant T | S(t) = i\} dt, \tag{3.12}$$

$$\hat{n}_i = E_{\hat{\phi}_l}[n_i | N(u), 0 \leqslant u \leqslant T] = \frac{1}{P_{\hat{\phi}_l}\{N(u), 0 \leqslant u \leqslant T\}} \times \sum_0^T P_{\hat{\phi}_l}\{N(u), 0 \leqslant u < t_k, S(t_k) = i\} P_{\hat{\phi}_l}\{N(u), t_k \leqslant u \leqslant T | S(t_k) = i\}. \tag{3.13}$$

From the above three new estimates, we see that they all contain forward-backward densities. The forward density under $\hat{\phi}_l$ is

$$P_{\hat{\phi}_l}\{N(u), 0 \leqslant u < t, S(t-) = i\} = \pi \left\{ \prod_{k=1}^{N(t)} f(y_k) \right\} \bar{F}(t - t_{N(t)}) \mathbf{1}_i \tag{3.14}$$

and the backward density under $\hat{\phi}_l$ is

$$P_{\hat{\phi}_l}\{N(u), t \leqslant u \leqslant T | S(t-) = j\} = \mathbf{1}_i' f(t_{N(t)+1} - t) \left\{ \prod_{k=N(t)+2}^{n} f(y_k) \mathbf{1}_i \right\}, \tag{3.15}$$

where $\mathbf{1}_i$ denotes an $r \times 1$ column vector of zeros except the ith element is one, $f(y)$ is as defined in chapter 2, $\bar{F}(t)$ is a $r \times r$ matrix whose (i, j) element denotes the transition probability of S from state i at time zero, to state j at time t, where during this period $[0, t]$ no jumps of process S occurred. The matrix $\bar{F}(t)$ is given as

$$\bar{F}(t) = e^{(Q - \Lambda)t}. \tag{3.16}$$

Note that both $f(t)$ and $\bar{F}(t)$ should be calculated under the newest parameters $\hat{\phi}_l$.

With the new estimates $\hat{m}_{ij}$, $\hat{T}_i$ and $\hat{n}_i$, the new parameter $\{(\hat{q}_{ij}, \hat{\lambda}_i), i, j = 1, \ldots, r\}$ would be given by

$$\begin{aligned} \hat{q}_{ij} &= \frac{\hat{m}_{ij}}{\hat{T}_i}, i \neq j \\ \hat{\lambda}_i &= \frac{\hat{n}_i}{\hat{T}_i}. \end{aligned} \tag{3.17}$$

Then we could calculate new $\hat{m}_{ij}$, $\hat{T}_i$ and $\hat{n}_i$ under new parameters, etc. As we calculate these estimates iteratively, the likelihood values corresponding to these estimates will form an increasing sequence.

3.2.1 Forward-Backward recursions

The forward-backward densities, can be calculated recursively which simplifies their calculation greatly. Make $L(0) = \pi$ and $R(n+1) = \mathbf{1}$. The products part in forward density $\pi \prod_{k=1}^{N(t)} f(y_k)$ could be written as $L(k) = L(k-1)f(y_k)$, for $k = 1, \ldots, n$. The products part in backward density $\prod_{k=N(t)+2}^{n} f(y_k)\mathbf{1}_i$ could be similarly written as $R(k) = f(y_k)R(k+1)$, for $k = n, \ldots, 1$.

But we should not use $L(k)$ and $R(k)$ to calculate directly, because the forward-backward densities will approach to zero or infinity exponentially as $n \to \infty$. So we need to scale the forward-backward densities [15], [23], [24]. Let $c_k = p(y_k|y^{k-1})$, for $k = 1, \ldots, n$. Let $P_{\hat{\phi}_l}\{N(u), t \leqslant u \leqslant T\}$ be denoted by $p(y^n) = p(y_1, \ldots, y_n) = \prod_{i=1}^{n} p(y_i|y^{i-1})$, let $p(y_1|y_0) = p(y_1)$. Therefore, we could rewrite $\hat{m}_{ij}$ in terms of the c_k as follows:

$$\hat{m}_{ij} = \sum_{k=1}^{n} \frac{q_{ij}}{c_k} \left(\pi \prod_{l=1}^{k-1} \frac{f(y_l)}{c_l} \right) \times \int_{t_{k-1}}^{t_k -} \bar{F}(t - t_{k-1}) \mathbf{1}_i \mathbf{1}_j' f(t_k - t) dt \left(\prod_{l=k+1}^{n} \frac{f(y_l)}{c_l} \mathbf{1} \right). \tag{3.18}$$

The scaled forward recursion is redefined as

$$\begin{aligned}\tilde{L}(0) &= \pi \\ \tilde{L}(k) &= \frac{\tilde{L}(k-1)f(y_k)}{c_k}, k = 1, \ldots, n,\end{aligned} \tag{3.19}$$

where c_k is defined by

$$c_k = \pi \prod_{l=1}^{k-1} \frac{f(y_l)}{c_l} f(y_k)\mathbf{1} = \tilde{L}(k-1)f(y_k)\mathbf{1}, k = 1, \ldots, n. \tag{3.20}$$

And the scaled backward recursion is defined as

$$\begin{aligned}\tilde{R}(n+1) &= \mathbf{1} \\ \tilde{R}(k) &= \frac{f(y_k)\tilde{R}(k+1)}{c_k}, k = n, \ldots, 1,\end{aligned} \tag{3.21}$$

We could also get the scaled forward-backward densities directly,

$$\begin{aligned}\tilde{L}(k) &= \pi \prod_{l=1}^{k} \frac{f(y_l)}{c_l}, k = 1, \ldots, n, \\ \tilde{R}(k) &= \prod_{l=k}^{n} \frac{f(y_l)}{c_l}\mathbf{1}, k = n, \ldots, 1,\end{aligned} \tag{3.22}$$

In terms of c_k, the log-likelihood function could be rewritten as follow:

$$\log p(y^n) = \sum_{k=1}^{n} \log(y_k|y^{k-1}) = \sum_{k=1}^{n} \log c_k. \tag{3.23}$$

3.2.2 Implementation of EM Algorithm

To make the implementation easier, we need to rewrite the $\hat{m}_{ij}$, $\hat{T}_i$ and $\hat{n}_i$. Let $\hat{m}$ denotes an $r \times r$ matrix where its (i,j) element is $\hat{m}_{ij}$, let $\hat{n}$ denote a $r \times 1$ matrix whose ith element is $\hat{n}_i$. As shown in [15], [25]

$$\hat{m} = Q \odot \sum_{k=1}^{n} \frac{\mathfrak{I}_k'}{c_k}, \tag{3.24}$$

where $\odot$ denotes the Schur-Hadamard product of two matrices. and $\mathfrak{I}_k$ is the upper right $r \times r$ matrix of $e^{C_k y_k}$. We could use the function *expm* in MATLAB to calculate the matrix exponential [26]. C_k is a $2r \times 2r$ matrix

$$C_k = \begin{bmatrix} Q - \Lambda & \Lambda R(k+1) L(k-1) \\ \mathbf{0} & Q - \Lambda \end{bmatrix}, \tag{3.25}$$

where $\mathbf{0}$ is an $r \times r$ matrix that all of its elements are zero.

$$\hat{n} = \sum_{k=1}^{n} L(k)' \odot R(k+1), \tag{3.26}$$

and $\hat{T}_i = \hat{m}_{ii}/q_{ii}$. The likelihood function of the observable process under parameter ϕ could be rewritten in terms of the c_k as follow:

$$p(y_1, ..., y_n; \Phi) = \pi \left\{ \prod_{k=1}^{n} f(y_i; \Phi) \right\} \mathbf{1} = \prod_{k=1}^{n} c_k. \tag{3.27}$$

3.3 Recursive Algorithm

The recursive algorithm we use here was introduced in [27]. The advantage of the recursive algorithm is on-line and does not require a large memory to save all the data. Instead of

estimating the parameters after we have got all the data as in the EM algorithm, we receive a small block of data and then obtain a new parameter estimate based on the latest block and previous estimate. Suppose we have made l estimations and got the parameter $\hat{\phi}_l$ and the $l+1$th block of data $\boldsymbol{y}_{l+1}$, $\boldsymbol{y}_{l+1} = (y_{ln}, \ldots, y_{(l+1)n})$, where n is the block size. But since we do not need the former l blocks of data, we could simply number the data of $l+1$th block $\boldsymbol{y}_{l+1} = (y_0, \ldots, y_n) = y^n$. The $l+1$th estimate $\hat{\phi}_{l+1}$ would be

$$\hat{\phi}_{l+1} = \mathscr{P}_{\Phi}(\hat{\phi}_l + \gamma_l \Psi(\boldsymbol{y}_{l+1}; \hat{\phi}_l)), \tag{3.28}$$

where γ_l is convergence factor $\gamma_l = \gamma_0 l^{-\alpha}$, for some $\gamma_0 > 0$, $\alpha \in (0.5, 1]$, $\Psi(\boldsymbol{y}_{l+1}; \hat{\phi}_l)$ is the derivative of the log-likelihood function, i.e. score function of $\boldsymbol{y}_{l+1}$ with respect to the parameters $\hat{\phi}_l$. $\mathscr{P}_{\Phi}$ is a projection that confines the parameter estimate within a feasible set.

After we obtain the $l+1$th estimate, we could average it in case to improve its statistical properties

$$\bar{\phi}_{l+1} = \frac{1}{l+1} \sum_{k=1}^{l+1} \hat{\phi}_k. \tag{3.29}$$

To simplify the derivative, define a new parameter matrix θ as follows:

$$\theta_{ij} = \begin{cases} q_{ij} & i \neq j \\ \lambda_{ii} & i = j. \end{cases} \tag{3.30}$$

Reshape matrix θ into an $r^2 \times 1$ vector whose elements are taken column-wise from θ. The vector is denoted by vec θ, vec is known as standard vectorization function. Define a new vector ξ_k which is a $1 \times r$ row vector such that $\xi_k(j) = p(S_k = j | y^k)$. It could be written in recursive form as

$$\xi_k = \xi_{k-1} f(y_k; \theta) c_k^{-1}. \tag{3.31}$$

Then c_k could be written as

$$c_k = \xi_{k-1} f(y_k;\theta)\mathbf{1}. \tag{3.32}$$

As shown in [27], the score vector of an MMPP given the observed process S is

$$\begin{aligned}\Psi(y^k;\theta) &= \frac{\partial}{\partial(\text{vec }\theta)'}\log p(y^k) \\ &= \frac{1}{p(y^k)}\frac{\partial}{\partial(\text{vec }\theta)'}p(y^k) \\ &= \sum_{i=1}^{r}\frac{1}{p(y^k)}\frac{\partial}{\partial(\text{vec }\theta)'}p(y^k, S_k = i).\end{aligned} \tag{3.33}$$

which is a $1 \times r^2$ vector. To get rid of the sum, define a new $r \times r^2$ matrix $\psi(y^k;\theta)$ such that its (i,j) element is

$$\psi(y^k;\theta)_{ij} = \frac{1}{p(y^k)}\frac{\partial}{\partial(\text{vec }\theta)_j}p(y^k, S_k = i). \tag{3.34}$$

The relationship between $\psi(y^k;\theta)$ and the score vector $\Psi(y^k;\theta)$ is given by

$$\Psi(y^k;\theta) = \mathbf{1}'\psi(y^k;\theta). \tag{3.35}$$

Expanding the $\psi(y^k;\theta)_{ij}$, as was done in [27], we obtain

$$\psi(y^k;\theta)_{ij} = c_k^{-1}\sum_{\alpha=1}^{r}\left\{\psi(y^{k-1};\theta)_{\alpha j} f_{\alpha i}(y_k;\theta) + \xi_{k-1}(\alpha)\frac{\partial}{\partial(\text{vec }\theta)_j}f_{\alpha i}(y_k;\theta)\right\}. \tag{3.36}$$

The matrix $\psi(y^k;\theta)$ would be

$$\psi(y^k;\theta) = c_k^{-1}\left\{f(y_k;\theta)'\psi(y^{k-1};\theta) + (I_r \otimes \xi_{k-1})\frac{\partial \text{vec } f(y_k;\theta)}{\partial(\text{vec }\theta)'}\right\}, \tag{3.37}$$

where I_r is an $r \times r$ identity matrix and $\otimes$ is the Kronecker product. From this equation, we can see that we have already got all the terms except the last term which is also given by [27]

$$\frac{\partial \text{vec } f(y;\theta)}{\partial(\text{vec } \theta)'} = (\Lambda \otimes I_r) H_{r^2} e^{N_y} M'_{r^2} \mathscr{I} + (I_r \otimes e^{Q-\Lambda} y)\, \mathscr{J}\,, \tag{3.38}$$

where H_{r^2} is an $r^2 \times 2r^2$ matrix whose left r^2 columns are I_{r^2} and the right r^2 columns are zeros, i.e. zero matrix $\mathbf{0}$. M_{r^2} is also an $r^2 \times 2r^2$ matrix, but its left r^2 columns are zero matrix $\mathbf{0}$ and the right r^2 columns are I_{r^2}. N is a $2r^2 \times 2r^2$ matrix

$$N = \begin{bmatrix} I_r \otimes (Q-\Lambda) & I_r \otimes I_r \\ \mathbf{0} & (Q-\Lambda)' \otimes I_r \end{bmatrix}. \tag{3.39}$$

$\mathscr{I}$ is an $r^2 \times r^2$ matrix $\mathscr{I} = \partial \text{vec}(Q-\Lambda)/\partial(\text{vec } \theta)'$. Its kth column $\mathscr{I}_k$ is given by $\mathscr{I}_{i+r(j-1)} = \text{vec } V(i,j)$. V is a $r \times r$ matrix whose (i,j) element is

$$V(i,j) = \frac{\partial(Q-\Lambda)}{\partial(\theta_{ij})'} = \begin{cases} -\mathbf{1}_{ii} & i = j \\ \mathbf{1}_{ij} - \mathbf{1}_{ii} & i \neq j \end{cases}, \tag{3.40}$$

where $\mathbf{1}_{ij}$ denotes an $r \times r$ matrix of zeros except for its (i,j) element is equal to one. Similarly, $\mathscr{J}$ is an $r^2 \times r^2$ matrix $\mathscr{J} = \partial \text{vec}(\Lambda)/\partial(\text{vec } \theta)'$. Its kth column $\mathscr{J}_k$ is given by $\mathscr{J}_{i+r(j-1)} = \text{vec } U(i,j)$. U is a $r \times r$ matrix whose (i,j) element is

$$U(i,j) = \frac{\partial(\Lambda)}{\partial(\theta_{ij})'} = \begin{cases} -\mathbf{1}_{ii} & i = j \\ 0 & i \neq j \end{cases}. \tag{3.41}$$

3.4 Numerical Results

3.4.1 Numerical Results of EM Algorithm

In this estimation, we choose $r = 2$, hence we have $r^2 = 4$ parameters. Denote the true parameters by ϕ_0 and denote the initial value by $\hat{\phi}_0$. The true value of case A and B are the same, it is given by

$$\phi_0 = (q_{12}, q_{21}, \lambda_1, \lambda_2) = (10, 1, 100, 10)\,. \tag{3.42}$$

The true value of case C and D are the same, it is given by

$$\phi_0 = (q_{12}, q_{21}, \lambda_1, \lambda_2) = (5, 2, 100, 50)\,. \tag{3.43}$$

The initial values of the four cases are as follows:

1. Case A

$$\hat{\phi}_0 = (q_{12}, q_{21}, \lambda_1, \lambda_2) = (7.36, 0.88, 81.7, 9.48) \tag{3.44}$$

2. Case B

$$\hat{\phi}_0 = (q_{12}, q_{21}, \lambda_1, \lambda_2) = (1, 10, 500, 1) \tag{3.45}$$

3. Case C

$$\hat{\phi}_0 = (q_{12}, q_{21}, \lambda_1, \lambda_2) = (10.12, 1.9, 128, 53.9) \tag{3.46}$$

4. Case D

$$\hat{\phi}_0 = (q_{12}, q_{21}, \lambda_1, \lambda_2) = (10, 1, 150, 1) \tag{3.47}$$

Tables 3.1-3.4 show the numerical results of four cases EM algorithm estimation. The bias in the tables represents the difference between the mean of the estimates and the true parameters, i.e., $\sum_{k=1}^{n}(\hat{\phi}_k - \phi_0)/n$. σ^2 is the mean squared error in this estimation, $\sum_{k=1}^{n}(\hat{\phi}_k - \phi_0)^2/n$, where n is the number of observations. From the given initial value

Table 3.1: Bias and MSE in EM estimating an MMPP of case A

Case A	q_{12}	q_{21}	λ_1	λ_2
True Values	10	1	100	10
Initial Values	7.36	0.88	81.7	9.48
Mean Estimation Results	9.20	0.92	101.48	10.13
Bias	-0.80	-0.08	1.48	0.13
σ^2	0.92	0.01	29.98	0.40

Table 3.2: Bias and MSE in EM estimating an MMPP of case B

Case B	q_{12}	q_{21}	λ_1	λ_2
True Values	10	1	100	10
Initial Values	1	10	500	1
Mean Estimation Results	10.50	1.16	95.2	9.93
Bias	0.50	0.16	-4.80	-0.07
σ^2	1.74	0.10	35.58	0.21

and true values above, we could see that in cases A and C, the initial values are closer to the true values than in cases B and D, and hence the parameter is easier to estimate. In cases C and D, the differences between q_{12} and q_{21}, and between λ_1 and λ_2, are smaller than in cases A and B, and are harder to estimate. From the results we can see that these differences do not affect the performance of EM algorithm much.

Table 3.3: Bias and MSE in EM estimating an MMPP of case C

Case C	q_{12}	q_{21}	λ_1	λ_2
True Values	5	2	100	50
Initial Values	10.12	1.9	128	53.9
Mean Estimation Results	4.90	2.01	98.00	51.35
Bias	-0.10	0.01	-2.00	1.35
σ^2	0.77	0.93	19.23	7.64

Table 3.4: Bias and MSE in EM estimating an MMPP of case D

Case D	q_{12}	q_{21}	λ_1	λ_2
True Values	5	2	100	50
Initial Values	10	1	150	1
Mean Estimation Results	5.92	2.91	96.42	50.01
Bias	0.92	0.91	-3.58	0.01
σ^2	1.38	2.03	52.22	5.23

3.4.2 Numerical Results of Recursive Algorithm

Similar to the EM algorithm, we also choose $r = 2$, thus we need to estimate $r^2 = 4$ parameters. The true value and initial value for the recursive algorithm are shown as follows. Recall the parameter matrix in 3.30, when $r = 2$, the θ is

$$\theta = \begin{pmatrix} \lambda_1 & q_{12} \\ q_{21} & \lambda_2 \end{pmatrix}. \tag{3.48}$$

The true value of case A and B are the same, it is given by

$$\theta_0 = \begin{pmatrix} 100 & 10 \\ 1 & 10 \end{pmatrix}. \tag{3.49}$$

The true value of case C and D are the same, it is given by

$$\theta_0 = \begin{pmatrix} 100 & 5 \\ 2 & 50 \end{pmatrix}. \tag{3.50}$$

The initial values of the four cases are as follows:

1. Case A

$$\hat{\theta}_0 = \begin{pmatrix} 97.7 & 9.36 \\ 0.98 & 9.88 \end{pmatrix} \tag{3.51}$$

2. Case B

$$\hat{\theta}_0 = \begin{pmatrix} 81.7 & 7.36 \\ 0.88 & 9.48 \end{pmatrix} \tag{3.52}$$

3. Case C

$$\hat{\theta}_0 = \begin{pmatrix} 98 & 4.5 \\ 1.9 & 49 \end{pmatrix} \tag{3.53}$$

4. Case D

$$\hat{\theta}_0 = \begin{pmatrix} 80 & 3 \\ 1 & 45 \end{pmatrix} \tag{3.54}$$

Similarly to the result of the EM algorithm, in cases A and C, the initial values are closer to the true values than in cases B and D, and hence the parameter is easier to estimate. In cases C and D, the differences between q_{12} and q_{21}, and between λ_1 and λ_2, are smaller than in cases A and B, and hence are harder to estimate.

Except for the initial value of the parameter, there are some other factors that may affect the estimation results. These are the convergence factors γ_0 and α. In this estimation, we choose $\gamma_0 = 0.5,\ 0.1,\ 0.07,\ 0.05$, and $\alpha = 1,\ 0.7$. For different initial parameter values, I use different parameters. If the initial value is close to the true value, I use smaller γ_0, and if the initial value is far from the true value, I use larger γ_0. Tables 3.5-3.8 show the numerical results of the four cases. The bias and σ^2 in the tables are defined the same way as before. Let $\hat{\theta}_l$ denotes the kth estimate, the bias equals $\sum_{l=1}^{n}(\hat{\theta}_l - \theta_0)/n$, $\sigma^2 = \sum_{l=1}^{n}(\hat{\theta}_l - \theta_0)^2/n$.

Table 3.5: Bias and MSE in recursive estimating an MMPP of case A

Case A			q_{12}	q_{21}	λ_1	λ_2
True value			10	1	100	10
Initial value			9.36	0.98	97.7	9.88
γ_0	α	Estimation results				
0.1	1	mean	10.112	0.998	98.428	9.918
		Bias	0.112	-0.002	-1.572	-0.082
		σ_s^2	0.083	0.067	2.477	0.114
0.07	1	mean	10.707	1.085	98.343	9.567
		Bias	0.707	0.085	-1.657	-0.433
		σ_s^2	0.610	0.037	2.902	4.271
0.05	1	mean	9.740	1.098	98.084	9.930
		Bias	-0.260	0.098	-1.916	-0.070
		σ_s^2	0.074	0.025	3.675	0.096

From the results we can see that the initial values affect the performance a lot, but with closer initial values, we could get better results. And if given initial values that are far away from the true values like in cases B and D in the EM algorithm, sometimes the produce does not converge. Table 3.9-3.12 show the averaged estimation result, after averaged, the mean squared error σ^2 is decreased. The boundary here is $\theta_l \in [0, \infty)$, for $l = 1, \ldots, r^2$.

3.4.3 Summary of Numerical Results

From the results we could see that the estimations of Λ are always better than Q. This is because the Λ is the rate of the observable process X, hence the components of Λ are easier to estimate, while Q is the parameter of the underlying process S, hence its components are harder to estimate.

We could also see that given the same amount of data, the EM algorithm performs better than the recursive algorithm. The performance of the EM algorithm does not depend on the initial values much for the examples given here, while the performance of the recursive algorithm is affected by the initial values much more. If the initial value is closer to the true value, the recursive algorithm performs better. The estimation results of the recursive algorithm are also affected by the other factors γ_0 and α, in $\gamma_l = \gamma_0 l^{-\alpha}$ in equation 3.28.

Table 3.6: Bias and MSE in recursive estimating an MMPP of case B

Case B			q_{12}	q_{21}	λ_1	λ_2
True value			10	1	100	10
Initial value			7.36	0.88	81.7	9.48
γ_0	α	Estimation results				
0.5	1	mean	11.702	1.103	88.103	9.778
		Bias	1.702	0.103	-11.897	-0.222
		σ_s^2	3.944	0.205	141.775	0.154
0.1	1	mean	8.628	1.018	83.170	9.595
		Bias	-1.372	0.018	-16.830	-0.405
		σ_s^2	1.949	0.067	283.260	0.326
0.1	0.7	mean	11.822	1.234	88.437	9.839
		Bias	1.822	0.234	-11.563	-0.161
		σ_s^2	3.928	0.114	153.314	0.430

Table 3.7: Bias and MSE in recursive estimating an MMPP of case C

Case C			q_{12}	q_{21}	λ_1	λ_2
True value			5	2	100	50
Initial value			4.5	1.9	98	49
γ_0	α	Estimation results				
0.1	1	mean	5.395	2.061	98.907	49.745
		Bias	0.395	0.061	-1.093	-0.255
		σ_s^2	0.292	0.064	1.196	0.090
0.07	1	mean	5.369	1.990	98.515	49.647
		Bias	0.369	-0.010	-1.485	-0.353
		σ_s^2	0.184	0.082	2.211	0.166
0.05	1	mean	4.964	2.030	98.424	49.420
		Bias	-0.036	0.030	-1.576	-0.580
		σ_s^2	0.049	0.022	2.485	0.361

Table 3.8: Bias and MSE in recursive estimating an MMPP of case D

Case D			q_{12}	q_{21}	λ_1	λ_2
True value			5	2	100	50
Initial value			3	1	80	45
γ_0	α	Estimation results				
0.5	1	mean	4.777	2.749	89.547	46.431
		Bias	-0.223	0.749	-10.453	-3.569
		σ_s^2	0.934	0.971	109.424	13.347
0.1	1	mean	3.792	3.022	82.192	46.109
		Bias	-1.208	1.022	-17.808	-3.891
		σ_s^2	1.589	1.156	317.149	15.178
0.1	0.7	mean	4.869	1.889	93.292	47.395
		Bias	-0.131	-0.111	-6.708	-2.605
		σ_s^2	1.086	0.485	46.955	7.378

Table 3.9: Averaged result of bias and MSE in recursive estimating an MMPP of case A

Case A			q_{12}	q_{21}	λ_1	λ_2
True value			10	1	100	10
Initial value			9.36	0.98	97.7	9.88
γ_0	α	Estimation results				
0.1	1	mean	9.990	1.063	98.226	10.077
		Bias	-0.010	0.063	-1.774	0.077
		σ_s^2	0.032	0.008	3.149	0.035
0.07	1	mean	10.672	1.290	98.062	9.922
		Bias	0.672	0.290	-1.938	-0.078
		σ_s^2	0.471	0.086	3.765	0.010
0.05	1	mean	9.655	1.079	97.934	10.001
		Bias	-0.345	0.079	-2.066	0.001
		σ_s^2	0.121	0.007	4.268	0.030

Table 3.10: Averaged result of bias and MSE in recursive estimating an MMPP of case B

Case B			q_{12}	q_{21}	λ_1	λ_2
True value			10	1	100	10
Initial value			7.36	0.88	81.7	9.48
γ_0	α	Estimation results				
0.5	1	mean	11.060	1.361	86.116	9.968
		Bias	1.060	0.361	-13.884	-0.032
		σ_s^2	1.378	0.151	182.819	0.067
0.1	1	mean	8.231	1.148	82.608	9.697
		Bias	-1.769	0.148	-17.392	-0.303
		σ_s^2	3.165	0.043	602.480	0.170
0.1	0.7	mean	10.348	1.281	85.426	9.825
		Bias	0.348	0.281	-14.574	-0.175
		σ_s^2	0.351	0.082	212.575	0.056

Table 3.11: Averaged result of bias and MSE in recursive estimating an MMPP of case C

Case C			q_{12}	q_{21}	λ_1	λ_2
True value			5	2	100	50
Initial value			4.5	1.9	98	49
γ_0	α	Estimation results				
0.1	1	mean	5.348	2.188	98.726	49.693
		Bias	0.348	0.188	-1.274	-0.307
		σ_s^2	0.176	0.037	1.623	0.110
0.07	1	mean	5.155	2.087	98.441	49.599
		Bias	0.155	0.087	-1.559	-0.401
		σ_s^2	0.078	0.015	2.436	0.173
0.05	1	mean	4.901	2.069	98.328	49.407
		Bias	-0.099	0.069	-1.672	-0.593
		σ_s^2	0.041	0.021	2.798	0.361

Table 3.12: Averaged result of bias and MSE in recursive estimating an MMPP of case D

Case D			q_{12}	q_{21}	λ_1	λ_2
True value			5	2	100	50
Initial value			3	1	80	45
γ_0	α	Estimation results				
0.5	1	mean	5.294	3.330	86.313	47.289
		Bias	0.294	1.330	-13.687	-2.711
		σ_s^2	0.747	1.902	187.433	7.557
0.1	1	mean	3.317	2.957	81.445	45.946
		Bias	-1.683	0.957	-18.555	-4.054
		σ_s^2	2.839	0.949	344.296	16.443
0.1	0.7	mean	4.565	3.131	87.255	47.093
		Bias	-0.435	1.131	-12.745	-2.907
		σ_s^2	0.415	1.552	163.515	8.562

These factors should be chosen based on the distance from initial value to the true value. If the distance is small and we choose a large factor γ_l, the estimation procedure may not converge, and if the distance is large and we choose a smaller factor, the estimation results would be close to the initial value and still be far away from the true value. As illustrated before, the distance between q_{12} and q_{21}, and between λ_1 and λ_2, also have large effects on the performance of the recursive algorithm. In addition, the performance of the recursive algorithm is not as stable as the EM algorithm, thus the mean squared error is always larger than the EM algorithm.

For the recursive algorithm, comparing the original estimation results with the averaged results, the bias of the original estimation is always smaller than that of the averaged ones, but sometimes, the mean squared error is greater than that of the averaged ones. This is because by averaging the estimation results, we force the $l+1$th estimate to converge by the factor $1/l+1$ in equation 3.29. So the results obtained through averaging may not always be better than the original ones, but are more stable.

Chapter 4: Estimation Algorithms for Bivariate Markov Chains

In this chapter, we discuss the estimation procedure of the EM algorithm and recursive algorithm for the bivariate Markov chain. Numerical results are provided.

4.1 Likelihood Function [1]

The likelihood function of the bivariate Markov chain is similar to that of the MMPP, and it is also very important in the following discussion. Suppose that at $t = 0$, both processes X and S jump. Firstly, let's define some terms. Suppose the $k+1$th jump of process X happens at $t = T_k$, and denote the state of X at time T_k by $X_k = X_{T_k}$. Denote the state of S at time T_k, by $S_k = S_{T_k}$, for $k = 1, 2, \ldots$. The realization of T_k would be t_k. So the state of $Z = (X, S)$ at time T_k would be $Z_k = (X_k, S_k)$, its realization would be $z_k = (x_k, s_k)$. And denote the dwell time of X at state X_{k-1} by Y_k, i.e. $Y_k = T_k - T_{k-1}$, and its realization would be y_k. The complete form likelihood function is $p_{X_0,S_0,Y_1,X_1,S_1,\ldots,Y_n,X_n,S_n}(x_0, s_0, y_1, x_1, s_1, \ldots, y_n, x_n, s_n; \phi)$, and it could be written as

$$\begin{aligned} p_\phi(x_0, s_0, y_1, x_1, s_1, \ldots, y_n, x_n, s_n) &= p_\phi(z_0, y_1, z_1, \ldots, y_n, z_n) \\ &= p_\Phi(z_0) \prod_{k=1}^{n} p_\phi(y_k, z_k | z_{k-1}), \end{aligned} \tag{4.1}$$

because of the chain rule. And it is easy to see that

$$p_\phi(x_0, y_1, x_1, \ldots, y_n, x_n) = \sum_{s_0,\ldots,s_n} p_\phi(x_0, s_0, y_1, x_1, s_1, \ldots, y_n, x_n, s_n). \tag{4.2}$$

The likelihood function of the observable process is

$$\begin{aligned} p(X(t), 0 \leqslant t \leqslant T; \phi) &= p_{X_0, Y_1, X_1, \ldots, Y_n, X_n}(x_0, y_1, x_1, \ldots, y_n, x_n; \phi) \\ &= p_\phi(x_0, y_1, x_1, \ldots, y_n, x_n) = \sum_{s_0, \ldots, s_n} p_\phi(z_0) \prod_{k=1}^{n} p_\phi(y_k, z_k | z_{k-1}). \end{aligned} \tag{4.3}$$

In chapter 2 we defined the transition density function $f_{ij}^{ab}(t)$. It denotes the transition density of X when it starts from state a at time 0 while S is in state i, then jumps at time t to state b while S is in state j, which is $p_\phi(y_k = t, z_k = (b, j) | z_{k-1} = (a, i))$, for $k = 1, \ldots, n$. Note that S does not need to jump to state j from state i directly, it could jump to any states during time t, but at time t, it should be at state j. And S does not need to jump at time t though it could. Conversely, X does not jump until time t. With given parameter ϕ, the transition density function will change, so let's denote the one under parameter ϕ by $f_{ij}^{ab}(t; \phi)$, and so does $F_{ij}^{ab}(t; \phi)$, $f^{ab}(t; \phi)$, $\bar{F}_{ij}^{a}(t; \phi)$ and $\bar{F}^{a}(t; \phi)$. Now we could rewrite the likelihood function of the observable process as product of transition density matrices, [20], [28]

$$p(X(t), 0 \leqslant t \leqslant T; \phi) = \nu_{x_0}(\phi) \left\{ \prod_{l=1}^{n} f^{x_{l-1} x_l}(y_l) \right\} \mathbf{1} \tag{4.4}$$

where $\nu_{x_0}(\phi)$ is the x_0th element of row vector $\nu(\phi)$. This is the initial distribution of the bivariate Markov chain with parameter ϕ when X starts at state x_0, which corresponds with $p_\phi(z_0)$. And $\mathbf{1}$ is a column vector of ones as defined before, which correspond with the sum of $s_0, \ldots, s_n$. Similarly to the MMPP, we are looking for the likelihood function from $t = 0$ to $t = T$, not t_n. Suppose the $n + 1$th jumps happens after time T, which means there is no jump of X during (t_n, T). The density of $X(0) = a$ and there is no jump during $(0, t)$ is also defined before as matrix $\bar{F}^{a}(t)$. With the parameter ϕ, the likelihood function

of observable process X during time $t \in [0, T]$ would be

$$p(X(t), 0 \leqslant t \leqslant T; \phi) = \nu_{x_0}(\phi) \left\{ \prod_{l=1}^{n} f^{x_{l-1}x_l}(y_l) \right\} \bar{F}^{x_n}(T - t_n; \phi)\mathbf{1}. \tag{4.5}$$

The complete form likelihood function under parameter ϕ is $p(Z(t), 0 \leqslant t \leqslant T; \phi)$.

4.2 EM Algorithm

The EM algorithm we are using here is introduced in [1] and [8]. In the EM algorithm, we need to find the parameter $\hat{\phi}$ that maximizes the log-likelihood function given the observable process. Denote the lth iteration parameter estimate by $\hat{\phi}_l$ [1],

$$\hat{\phi}_{l+1} = \arg\max_{\hat{\phi} \in \Phi} E\{\log p(\{Z(t), 0 \leqslant t \leqslant T\}; \hat{\phi}) | X(t), 0 \leqslant t \leqslant T; \hat{\phi}_l\}. \tag{4.6}$$

Similarly to the MMPP, let m_{ij}^{ab} denotes the number of jumps of Z from (a, i) to (b, j) during $[0, T]$. Note that there is no special requirement that $i \neq j$ and $a \neq b$ simultaneously in MMPP case, and we are going to discuss those cases later.

$$m_{ij}^{ab} = \#\{t : 0 < t \leqslant T, Z(t-) = (a, i), Z(t) = (b, j)\}. \tag{4.7}$$

Let T_i^a denote the total dwell time of Z in state (a, i) during $[0, T]$, i.e.

$$T_i^a = \int_0^T I\{Z(t) = (a, i)\} dt. \tag{4.8}$$

For each estimate $\hat{\phi}$, for example, in lth iteration, we have got a new estimation $\hat{\phi}_l$, we can get new $\hat{m}_{ij}^{ab}$ and $\hat{T}_i^a$ corresponding with m_{ij}^{ab} and T_i^a by

$$\begin{aligned}\hat{m}_{ij}^{ab} &= E\{m_{ij}^{ab}|X(t), 0 \leqslant t \leqslant T; \hat{\phi}_l\} \\ &= \sum_{t\in[0,T]} p(Z(t-) = (a,i), Z(t) = (b,j)|X(\tau), 0 \leqslant \tau \leqslant T; \hat{\phi}_l),\end{aligned} \tag{4.9}$$

$$\hat{T}_i^a = E\{T_i^a|X(t), 0 \leqslant t \leqslant T; \hat{\phi}_l\}. \tag{4.10}$$

There are two conditions of $\hat{m}_{ij}^{ab}$

1. $a = b, i \neq j$ $\hat{m}_{ij}^{ab}$ becomes $\hat{m}_{ij}^{aa}$ under this condition.

$$\hat{m}_{ij}^{aa} = \int_0^T p(Z(t-) = (a,i), Z(t) = (a,j)|X(\tau), 0 \leqslant \tau \leqslant T)dt. \tag{4.11}$$

2. $a \neq b$

$$\hat{m}_{ij}^{ab} = \sum_{\substack{k:x_k=a,\\ x_{k+1}=b}} p(Z(t_k-) = (a,i), Z(t_k) = (a,j)|X(\tau), 0 \leqslant \tau \leqslant T)dt. \tag{4.12}$$

By these parameters, we could get the $l+1$th estimation of $\hat{\phi}$, i.e. $\hat{\phi}_{l+1}$ by

$$\hat{\phi}_{l+1} = \left\{ \frac{\hat{m}_{ij}^{ab}}{\hat{T}_i^a}, (a,i) \neq (b,j) \right\}. \tag{4.13}$$

4.2.1 Forward-Backward Recursions

Similarly to the MMPP, we need to find the forward-backward recursions of the bivariate Markov chains. Let $L(0) = \nu_{x_0}$ and $R(n+1) = \bar{F}^{x_n}(T - t_n)\mathbf{1}$. Then we could

write the forward density $L(k) = \{p(X(t), 0 \leqslant t \leqslant t_k, S_k = i), i \in \mathbb{S}_{x_k}\}$ recursively as $L(k) = L(k-1)f^{x_{k-1}x_k}(y_k)$, for $k = 1, \ldots, n$. And similarly, the backward density $R(k) = \{p(X(t), t_{k-1} < t \leqslant T | X_{k-1} = x_{k-1}, S_{k-1} = i), i \in \mathbb{S}_{x_{k-1}}\}'$ could be written recursively as $R(k) = f^{x_{k-1}x_k}(y_k)R(k+1)$, for $k = n, \ldots, 1$. We could rewrite the likelihood function of the observable process by forward-backward recursions as follow:

$$p(X(t), 0 \leqslant t \leqslant T) = L(k)R(k+1), \ k = 0, \ldots, n. \tag{4.14}$$

We also need to scale the forward-backward recursions. The scaled forward recursion is defined as

$$\begin{aligned} \tilde{L}(0) &= L(0) = \nu_{x_0} \\ \tilde{L}(k) &= \frac{\tilde{L}(k-1)f^{x_{k-1}x_k}(y_k)}{c_k}, k = 1, \ldots, n, \end{aligned} \tag{4.15}$$

where c_k is defined by

$$c_k = \tilde{L}(k-1)f^{x_{k-1}x_k}(y_k)\mathbf{1}, k = 1, \ldots, n. \tag{4.16}$$

the scaled backward recursion is defined as

$$\begin{aligned} \tilde{R}(n+1) &= R(n+1)\bar{F}^{x_n}(T - t_n)\mathbf{1} \\ \tilde{R}(k) &= \frac{f^{x_{k-1}x_k}(y_k)\tilde{R}(k+1)}{c_k}, k = n, \ldots, 1, \end{aligned} \tag{4.17}$$

The relationship between the scaled and unscaled forward-backward recursions are

$$\begin{aligned} \tilde{L}(k) &= \frac{L(k)}{\prod_{l=1}^{k} c_l}, k = 1, \ldots, n \\ \tilde{R}(k) &= \frac{R(k)}{\prod_{l=k}^{n} c_l}, k = n, \ldots, 1, \end{aligned} \tag{4.18}$$

Now we could rewrite the likelihood function with c_k

$$p(X(t), 0 \leqslant t \leqslant T) = \prod_{k=1}^{n} c_k. \tag{4.19}$$

Then the log-likelihood function would be

$$\log p(X(t), 0 \leqslant t \leqslant T) = \sum_{k=1}^{n} \log c_k. \tag{4.20}$$

4.2.2 Implementation of EM algorithm

With these forward-backward recursions, we could rewrite $\hat{m}_{ij}^{ab}$ and $\hat{T}_i^a$ which are given by [1]. The $\hat{m}_{ij}^{ab}$ under the first condition $(a = b, i \neq j)$, i.e. $\hat{m}_{ij}^{aa}$ is

$$\hat{m}_{ij}^{aa} = \left[G_{aa} \odot \sum_{k:x_k=a} \frac{\mathfrak{I}_k'}{c_{k+1}} \right]_{ij}, \tag{4.21}$$

where $\mathfrak{I}_k$ is the upper right $r_{x_k} \times r_{x_k}$ matrix of the matrix $e^{C_k y_{k+1}}$. C_k is a $2r_{x_k} \times 2r_{x_k}$ matrix Define a $2r_{x_k} \times 2r_{x_k}$ matrix

$$C_k = \begin{bmatrix} G_{x_k x_k} & G_{x_k x_{k+1}} \tilde{R}(k+2) \tilde{L}(k) \\ \mathbf{0} & G_{x_k x_k} \end{bmatrix}, for k = 0, \ldots, n-1 \tag{4.22}$$

Under the second condition $(a \neq b)$

$$\hat{m}_{ij}^{ab} = \left[G_{ab} \odot \sum_{\substack{k:x_k=a, \\ x_{k+1}=b}} \frac{\mathfrak{I}_k'}{c_{k+1}} \right]_{ij}, \tag{4.23}$$

where

$$\mathfrak{J}_k = \tilde{R}(k+2)\tilde{L}(k)e^{G_{x_k x_k} y_{k+1}}, k = 0, \ldots, n-1. \tag{4.24}$$

And

$$\hat{T}_i^a = \left[\sum_{k:x_k=a} \frac{\mathfrak{J}'_k}{c_{k+1}} \right]_{ii}. \tag{4.25}$$

4.3 Recursive Algorithm

The recursive algorithm for estimating the parameter of a bivariate Markov chain was introduced in [11, Chap. 7]. We use a similar algorithm as in the MMPP which is a special case of a bivariate Markov chain. Unlike the batch EM algorithm which uses the entire data iteratively, here we estimate the parameter recursively from the current estimate of the parameter and the newly acquired data $\{X_k, Y_k\}$. The algorithm works in a block mode. Recall that X_k represent the state and Y_k represent inter-event time of the observable process. Suppose that we have received the $l+1$th block of data $\boldsymbol{y}_l = (x_{(l-1)n}, y_{(l-1)n}, \ldots, x_{ln}, y_{ln})$, where $n+1$ is the block size, and after we have done l estimations, the lth estimate is $\hat{\phi}_l$. Then the $l+1$th estimate is given by

$$\hat{\phi}_{l+1} = \mathscr{P}_\Phi(\hat{\phi}_l + \gamma_l \Psi(\boldsymbol{y}_{l+1}; \hat{\phi}_l)). \tag{4.26}$$

All the terms are defined similarly to the MMPP; $\Psi(\boldsymbol{y}_{l+1}; \hat{\phi}_l)$ is the derivative of the log-likelihood function, i.e. the score function evaluated for the data $\boldsymbol{y}_{l+1}$ and parameters $\hat{\phi}_l$, γ_l is the convergence factor $\gamma_l = \gamma_0 l^{-\alpha}$, for some $\gamma_0 > 0$, $\alpha \in (0.5, 1]$, and $\mathscr{P}_\Phi$ is a projection onto the parameter space.

To simplify the calculation, we need to vectorize the parameter matrix ϕ, and we denote the lth parameter estimate by $\hat{\phi}_l$. The relationship between generator matrix G and parameter ϕ is illustrated as follows. G is composed of the elements of the parameter ϕ. For example for a $r = 2$ and $d = 2$ bivariate Markov chain, the generator matrix G would

be a 4×4 matrix, and the dimension of parameters ϕ would be $(r+d)^2 - (d+r) = 12$.

$$G = \begin{pmatrix} -\phi_4 - \phi_7 - \phi_{10} & \phi_4 & \phi_7 & \phi_{10} \\ \phi_1 & -\phi_1 - \phi_8 - \phi_{11} & \phi_8 & \phi_{11} \\ \phi_2 & \phi_5 & -\phi_2 - \phi_5 - \phi_{12} & \phi_{12} \\ \phi_3 & \phi_6 & \phi_9 & -\phi_3 - \phi_6 - \phi_9 \end{pmatrix} \tag{4.27}$$

Then we could get the score vector given the observed process X as follow:

$$\Psi(\boldsymbol{y}_k; \phi) = \frac{\partial}{\partial(\phi')} \log p(\boldsymbol{y}_k). \tag{4.28}$$

After we have got the $l+1$th estimate, we need to average it in case to improve its statistical properties

$$\bar{\phi}_{l+1} = \frac{1}{l+1} \sum_{k=1}^{l+1} \hat{\phi}_k. \tag{4.29}$$

4.3.1 Forward recursion

The likelihood function of the observable process under parameter ϕ is given before

$$p(X(t), 0 \leqslant t \leqslant T; \phi) = \nu_{x_0}(\phi) \left\{ \prod_{l=1}^{n} f^{x_{l-1} x_l}(y_l) \right\} \mathbf{1} \tag{4.30}$$

Denote the forward density of the 1st block of observable and underlying data $\boldsymbol{Y}_1, S_n$ under parameter ϕ by $p(\boldsymbol{y}_1, s_n; \phi)$. Its forward recursion is give by [11]

$$p(\boldsymbol{y}_1, s_n; \phi) = p_\phi(y_0^n, x_0^n, s_n) = \sum_{s_{n-1}} p_\phi(y_0^{n-1}, x_0^{n-1}, s_{n-1}) p_\phi(y_n, z_n | z_{n-1}), \tag{4.31}$$

where $y_0^n, x_0^n = (x_0, y_0, x_1, y_1, \ldots, x_n, y_n)$, and $p_\phi(y_0, x_0, s_0) = p_\phi(z_0)$ since $y_0 = 0$. Also, we need to define the forward recursion L. $L(k)$ is an $r \times 1$ vector for $k = 1, \ldots, n$, where r is the number of states of the underlying process S as defined in chapter 2. Under parameter ϕ, $L(k)$ is given by [9]

$$L(k;\phi) = (p_\phi(y_0^k, x_0^k, 1), \ldots, p_\phi(y_0^k, x_0^k, r)). \tag{4.32}$$

From equation (4.30), its recursive form is given by

$$L(k;\phi) = L(k-1;\phi) f^{x_{k-1}x_k}(y_k;\phi), \tag{4.33}$$

where $f^{ab}(t;\phi)$ is the transition density under parameter ϕ as defined before in equation (2.10) and (2.13). And $L(0;\phi)$ corresponds with the $\nu_{x_0}(\phi)$ in equation (4.29). Also we need to find the scaled forward recursion $\tilde{L}$ with scale factor c, which is defined as follows:

$$\begin{aligned} c_0 &= \nu_{x_0}(\phi)\mathbf{1}, \\ c_k &= \tilde{L}(k-1;\phi) f^{x_{k-1}x_k}(y_k;\phi)\mathbf{1}, \quad k = 1, \ldots, n. \end{aligned} \tag{4.34}$$

Then we could scale the forward recursion

$$\begin{aligned} \tilde{L}(0;\phi) &= \frac{\nu_{x_0}(\phi)}{c_0}, \\ \tilde{L}(k;\phi) &= \frac{\tilde{L}(k-1;\phi) f^{x_{k-1}x_k}(y_k;\phi)}{c_k}, \quad k = 1, \ldots, n. \end{aligned} \tag{4.35}$$

The relationship between the unscaled forward recursion $L(k;\phi)$ and the scaled forward recursion $\tilde{L}(k;\phi)$ is given by

$$\tilde{L}(k;\phi) = \frac{L(k;\phi)}{\prod_{l=0}^{k} c_l}, \quad k = 0, \ldots, n. \tag{4.36}$$

Note that c_k could also be written as

$$c_k = p_\phi(x_k, y_k | x_{k-1}, y_{k-1}, \ldots, x_1, y_1, x_0). \tag{4.37}$$

In terms of c_k, the likelihood function of the observable process $X(t)$ could be rewritten as

$$p(X(t), 0 \leqslant t \leqslant T; \phi) = \prod_{k=0}^{n} c_k. \tag{4.38}$$

The log-likelihood function would be

$$\log p(X(t), 0 \leqslant t \leqslant T; \phi) = \sum_{k=0}^{n} \log c_k. \tag{4.39}$$

4.3.2 Implementation of Recursive Algorithm

To calculate the score function $\Psi(\boldsymbol{y}_k; \phi)$, we need to define a new $\tilde{r} \times r$ matrix $\delta_n(\phi)$, where $\tilde{r}$ is the number of unknown parameters, i.e. the dimension of ϕ.

$$\begin{aligned} \delta_n(\phi) &= \frac{1}{\prod_{k=0}^{n} c_k} \left\{ \frac{\partial p_\phi(y_0^n, x_0^n, s_n)}{\partial \phi_l}, l = 1, \ldots, \tilde{r}; s_n = 1, \ldots, r \right\} \\ &= \frac{1}{\prod_{k=0}^{n} c_k} \left[D_\phi p_\phi(y_0^n, x_0^n, 1), D_\phi p_\phi(y_0^n, x_0^n, 2), \ldots, D_\phi p_\phi(y_0^n, x_0^n, r) \right]. \end{aligned} \tag{4.40}$$

Denote the derivative of a function $f(x;\phi)$, with respect to the lth element of ϕ, i.e. ϕ_l by $\partial_l f(x;\phi)$. Define an $\tilde{r} \times r$ matrix $\mathscr{X}(n-1;\phi)$ whose lth row is given by $\tilde{L}(n-1;\phi)\partial_l f^{x_{n-1}x_n}(y_n;\phi)$. Now $\delta_n(\phi)$ could be rewritten as [11]

$$\delta_n(\phi) = \frac{1}{c_k}\left[\delta_{n-1}(\phi) f^{x_{n-1}x_n}(y_n;\phi) + \mathscr{X}(n-1;\phi)\right]. \tag{4.41}$$

Recall that $f^{ab}(y_n;\phi)$ is given in chapter 2 that $f^{ab}(y;\phi) = e^{G_{aa}y}G_{ab}$. The score function could be rewritten as [11]

$$\begin{aligned} \Psi(\boldsymbol{y}_1;\phi) &= \Psi(y_0^n, x_0^n;\phi) \\ &= \frac{\partial}{\partial(\phi')}\log p(y_0^n, x_0^n) = \delta_n(\phi)\mathbf{1}. \end{aligned} \tag{4.42}$$

In order to calculate the score function $\Psi(\boldsymbol{y}_1;\phi)$, we need to find the derivative of matrix $f^{ab}(y;\phi)$ with respect to parameter ϕ_l, $\partial_l f^{ab}(y;\phi)$, in the matrix $\mathscr{X}(n-1;\phi)$, which is given in [29].

$$\partial_l f^{ab}(y;\phi) = \partial_l e^{G_{aa}y}G_{ab} = (\partial_l e^{G_{aa}y})G_{ab} + e^{G_{aa}y}(\partial_l G_{ab}). \tag{4.43}$$

The calculation of $\partial_l G_{ab}$ involves taking the derivative of each element of matrix G_{ab} with respect to ϕ_l, i.e. $\partial_l G_{ab} = \{\partial_l[(G_{ab})_{ij}]/\partial_l\phi_l\}$. Then we need to calculate the matrix derivative $\partial_l e^{G_{aa}y}$ which is given by [30]. Suppose we are going to find the derivative of the exponential of an $r \times r$ matrix A. In our case, $A = G_{aa}y$. Its derivative with respect to ϕ_l is $\partial_l A = \{\partial_l a_{ij}/\partial_l\phi_l\}$, where a_{ij} is the (i,j) element of matrix A. Note that $\partial_l A$ is also an $r \times r$ matrix. Define an $\tilde{r} \times r$ Jacobian matrix

$$\bigtriangledown A = [\mathrm{vec}(\partial_1 A, \ldots, \partial_{\tilde{r}} A)]. \tag{4.44}$$

Recall that vec is the standard vectorization function and $\tilde{r}$ is the dimension of ϕ. The derivative of the exponential of A is given by [11]

$$\bigtriangledown e^A = [e^C]_{12} \bigtriangledown A, \tag{4.45}$$

where C is given by

$$C = \begin{pmatrix} A' \bigotimes I_r & I_{r^2} \\ \mathbf{1} & I_r \bigotimes A \end{pmatrix}. \tag{4.46}$$

Recall that $\bigotimes$ denotes the Kronecker product, I_r denotes an $r \times r$ identity matrix, and $[e^C]_{12}$ is the upper right $r^2 \times r^2$ block of the matrix e^C.

4.4 Numerical Results

4.4.1 Numerical Results for EM Algorithm

If an MMPP and a bivariate Markov chain have the same number of underlying states r, then the estimation of the parameter of a bivariate Markov chain is usually much harder than the estimation of the parameter of an MMPP. Since even if $d = r$, the number of unknown parameters in the bivariate Markov chain is $(r+d)^2 - (d+r) = (2r)^2 - 2r$ while for the MMPP that number is only r^2.

Tables 4.1-4.4 show the numerical results of four bivariate Markov chains, all of them have the same true parameter, but different initials. The example we present is for a bivariate Markov chain with $r = 2, d = 2$. Its generator matrix is

$$G = \begin{pmatrix} -70 & 10 & 50 & 10 \\ 20 & -55 & 25 & 10 \\ 50 & 0 & -60 & 10 \\ 0 & 10 & 20 & -30 \end{pmatrix} \tag{4.47}$$

Let θ_i denotes the diagonal parameter on the ith row. Because $-\theta_i$ is the sum of the other elements at the same row, its value could represent the estimation accuracy of the other elements in that row. Instead of showing the estimation results of all sixteen parameters, we present the estimation results of the four diagonal parameters. The initial values of the four cases are

1. Case A

$$G = \begin{pmatrix} -55 & 5 & 45 & 5 \\ 15 & -40 & 20 & 5 \\ 45 & 0 & -50 & 5 \\ 0 & 5 & 25 & -30 \end{pmatrix} \tag{4.48}$$

2. Case B

$$G = \begin{pmatrix} -80 & 20 & 40 & 20 \\ 10 & -65 & 35 & 20 \\ 60 & 0 & -80 & 20 \\ 0 & 20 & 30 & -50 \end{pmatrix} \tag{4.49}$$

3. Case C

$$G = \begin{pmatrix} -120 & 30 & 70 & 20 \\ 2 & -8 & 5 & 1 \\ 70 & 0 & -100 & 3 \\ 0 & 1 & 2 & -3 \end{pmatrix} \tag{4.50}$$

4. Case D

$$G = \begin{pmatrix} -36 & 3 & 30 & 3 \\ 2 & -48 & 45 & 1 \\ 70 & 0 & -100 & 30 \\ 0 & 30 & 40 & -70 \end{pmatrix} \tag{4.51}$$

Table 4.1: Bias and MSE in EM estimating a bivariate Markov chain of case A

Case A	θ_1	θ_2	θ_3	θ_4
True Values	-70	-55	-60	-30
Initial Values	-55	-40	-50	-30
Bias	7.95	7.52	6.40	1.46
σ^2	64.19	68.16	42.92	3.46

Table 4.2: Bias and MSE in EM estimating a bivariate Markov chain of case B

Case B	θ_1	θ_2	θ_3	θ_4
True Values	-70	-55	-60	-30
Initial Values	-80	-65	-80	-50
Bias	-11.26	1.97	-16.25	-4.79
σ^2	129.13	6.75	266.62	23.91

Similarly to the MMPP, denote the true parameters by ϕ_0, the bias shown in the tables represents the difference between the mean of the estimates and the true parameters, i.e., $\sum_{l=1}^{n}(\hat{\phi}_l - \phi_0)/n$. σ^2 is the mean squared error in this estimation, $\sum_{l=1}^{n}(\hat{\phi}_l - \phi_0)^2/n$, where n is the number of observations.

Figures 4.1-4.8 show the estimation result versus the number of iterations. Note that the initials for these two parameters, $\phi_3 = 0$ and $\phi_5 = 0$, are the same as the true value, and their estimates are always the same as the true values too. So in the figures, the estimated value and true value coincide, so the blue line which denotes the estimated value is covered by the green line which denotes the true value. Thus we could only see a green line in the figures of parameters ϕ_3 and ϕ_5.

Table 4.3: Bias and MSE in EM estimating a bivariate Markov chain of case C

Case C	θ_1	θ_2	θ_3	θ_4
True Values	-70	-55	-60	-30
Initial Values	-120	-8	-100	-3
Bias	-4.04	4.30	1.60	0.54
σ^2	18.69	21.72	5.21	1.72

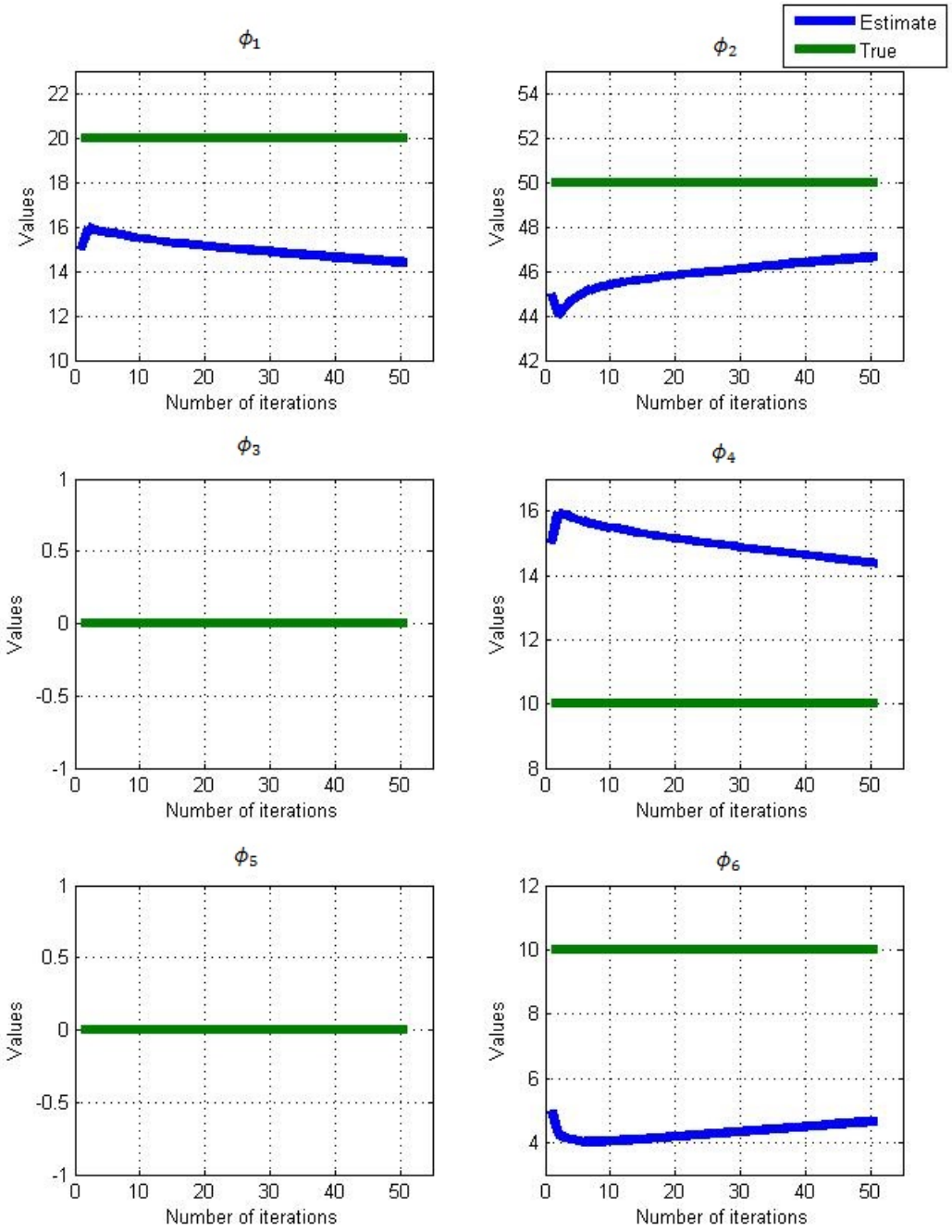

Figure 4.1: Estimation result versus number of iterations in EM estimating of case A part one

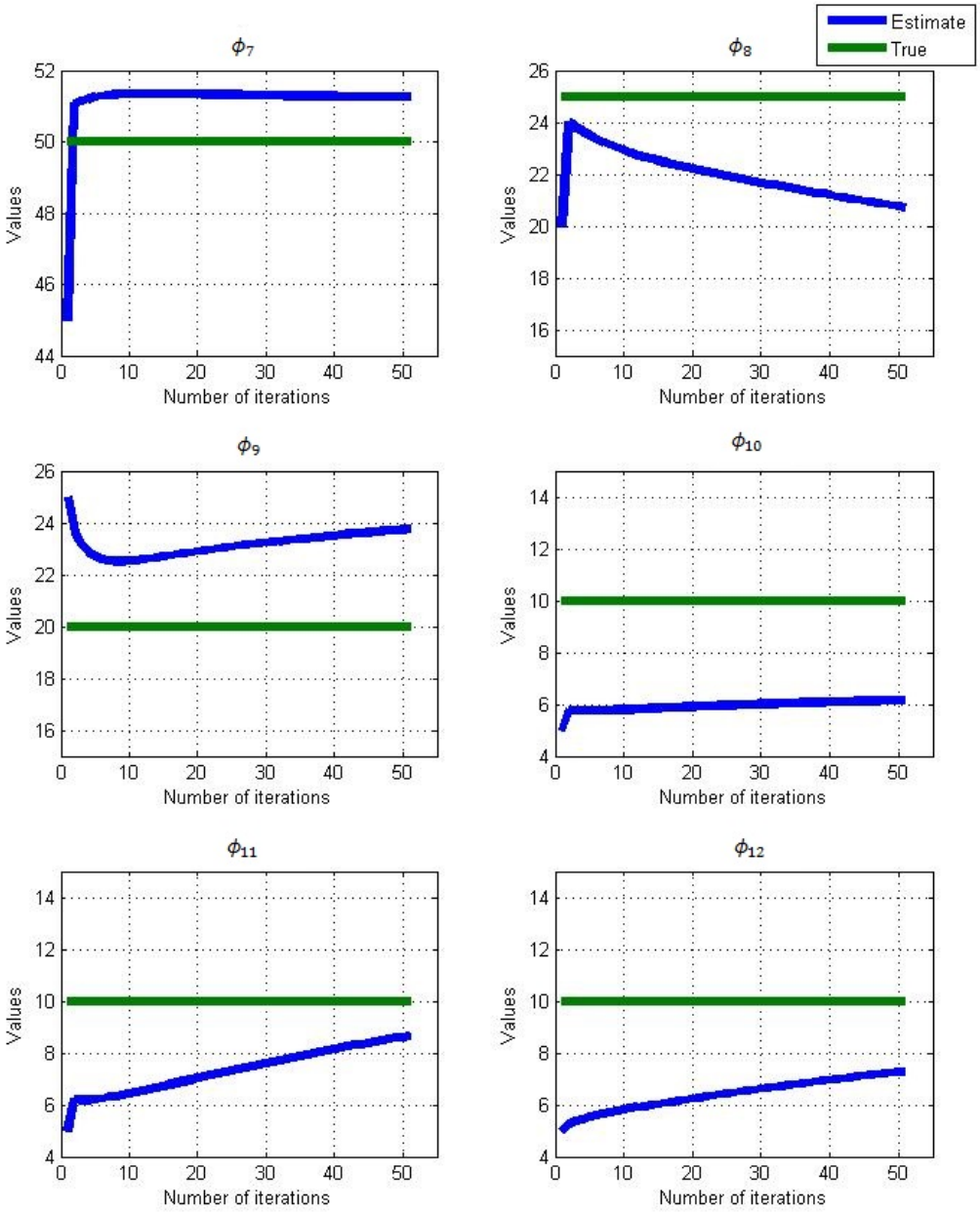

Figure 4.2: Estimation result versus number of iterations in EM estimating of case A part two

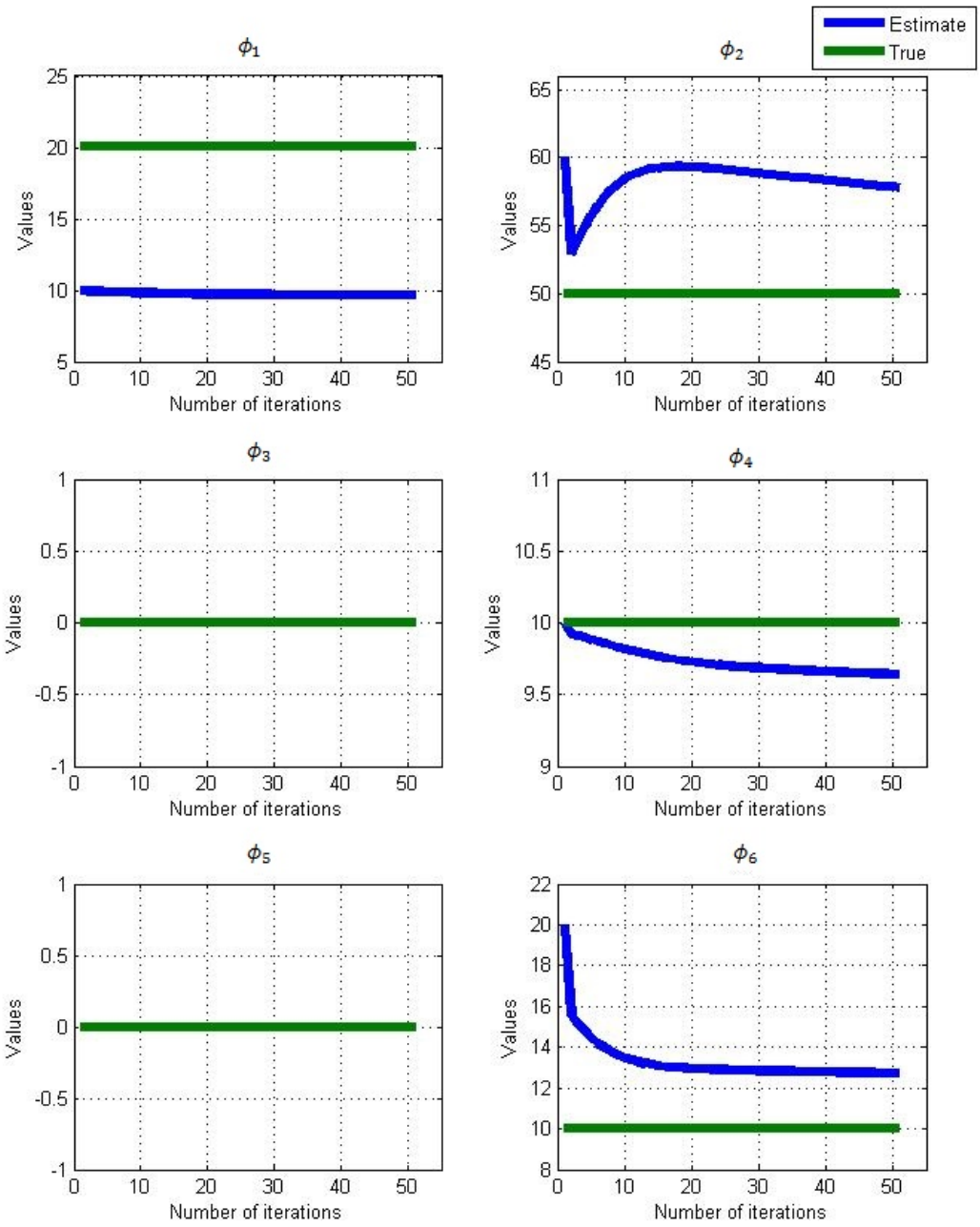

Figure 4.3: Estimation result versus number of iterations in EM estimating of case B part one

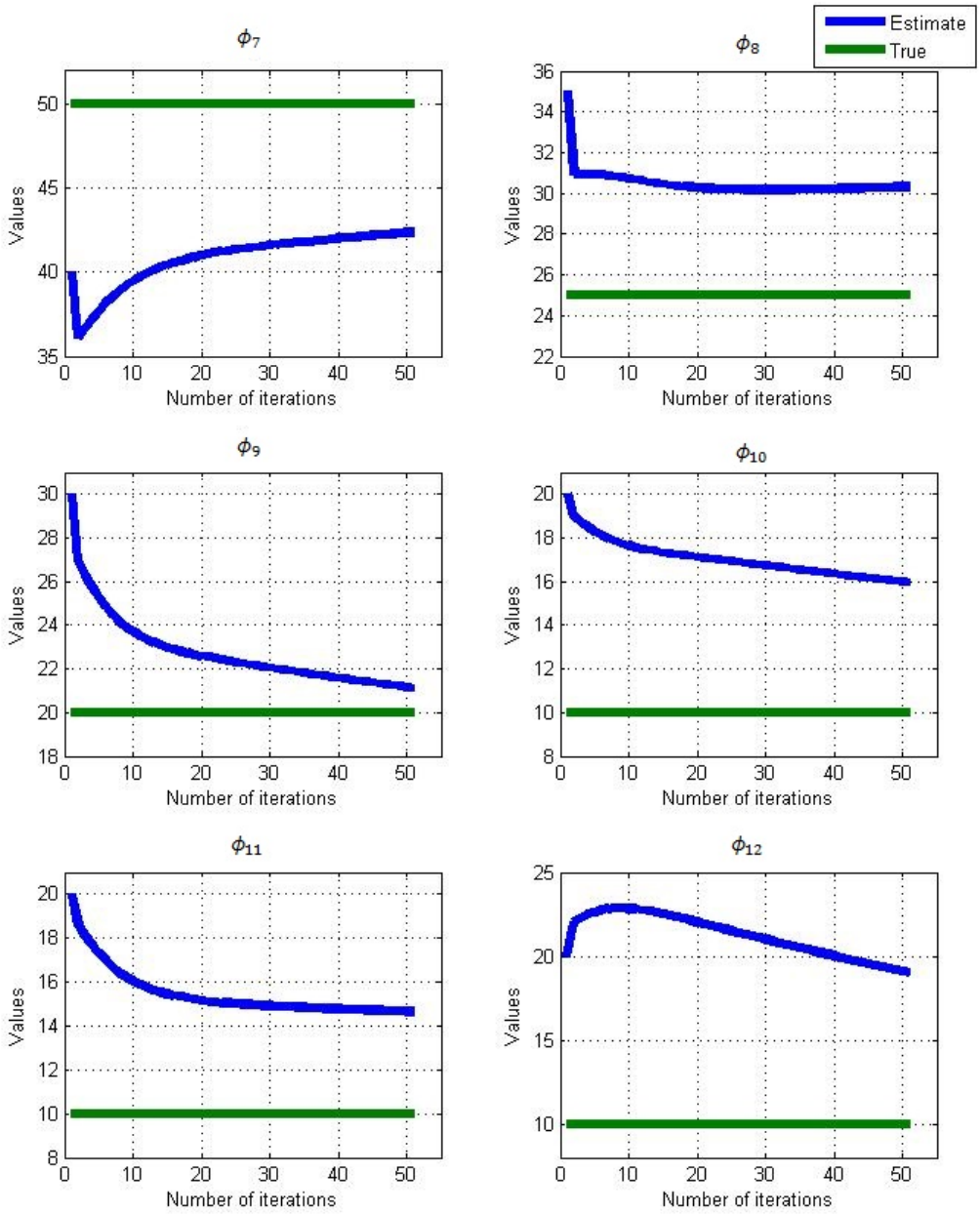

Figure 4.4: Estimation result versus number of iterations in EM estimating of case B part two

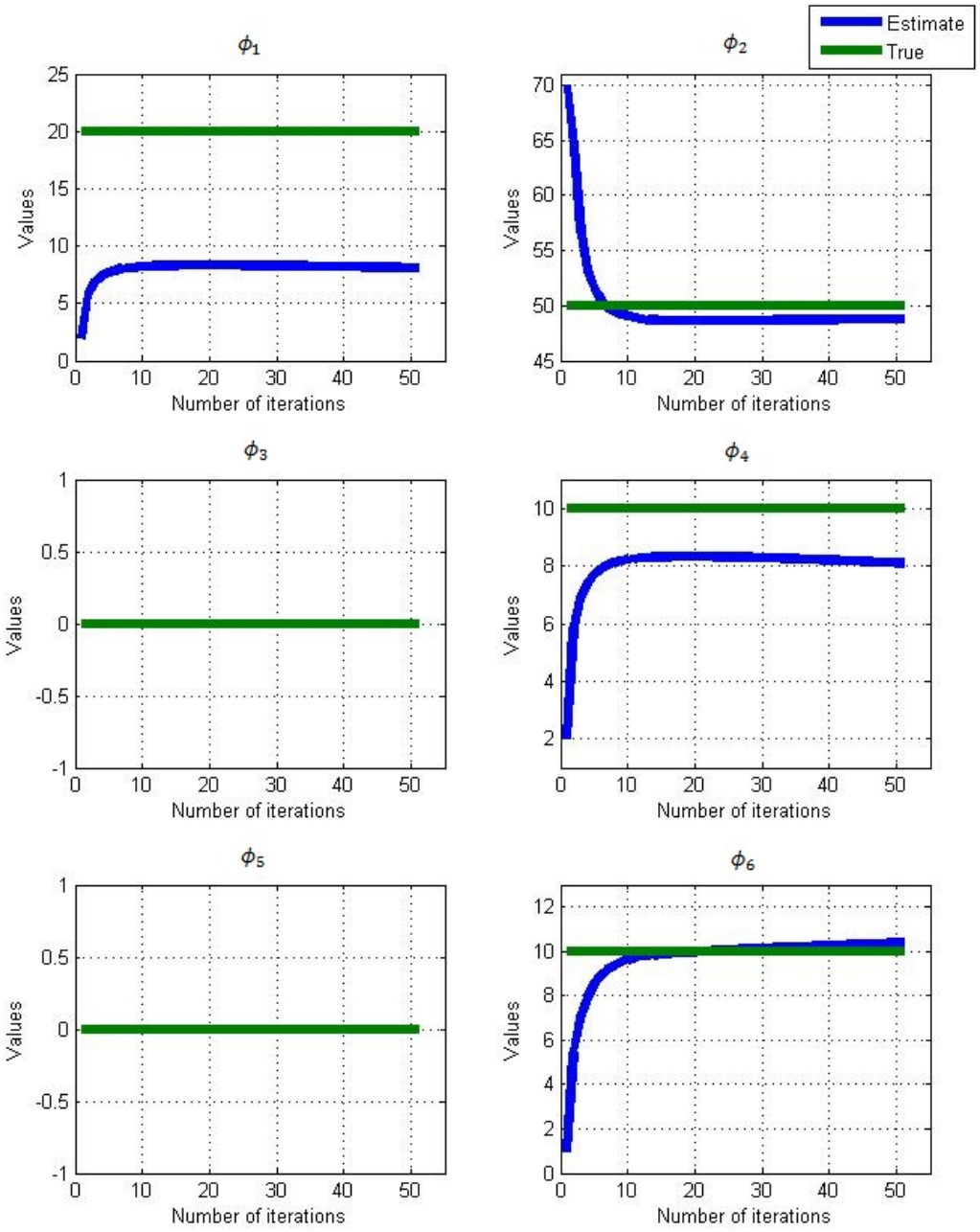

Figure 4.5: Estimation result versus number of iterations in EM estimating of case C part one

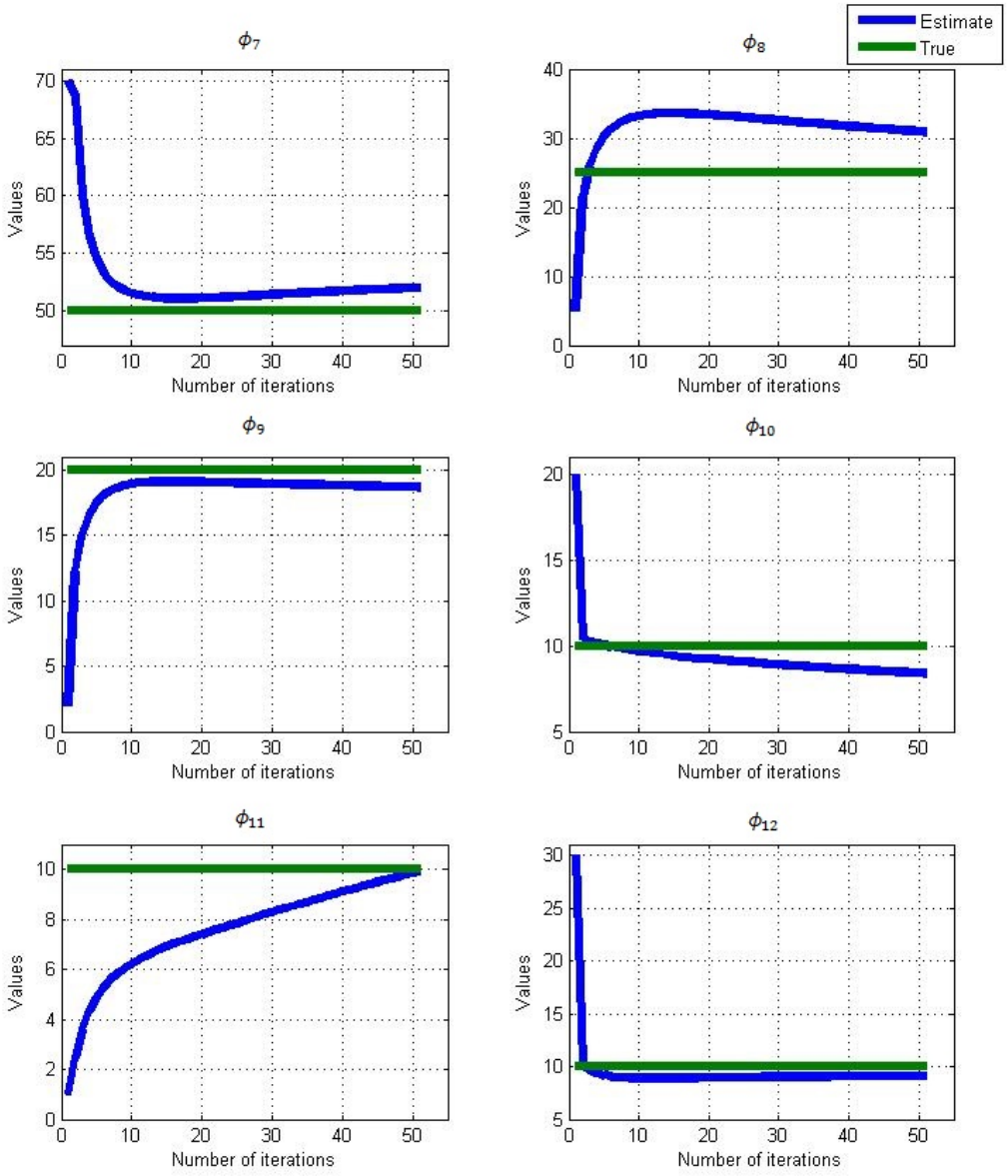

Figure 4.6: Estimation result versus number of iterations in EM estimating of case C part two

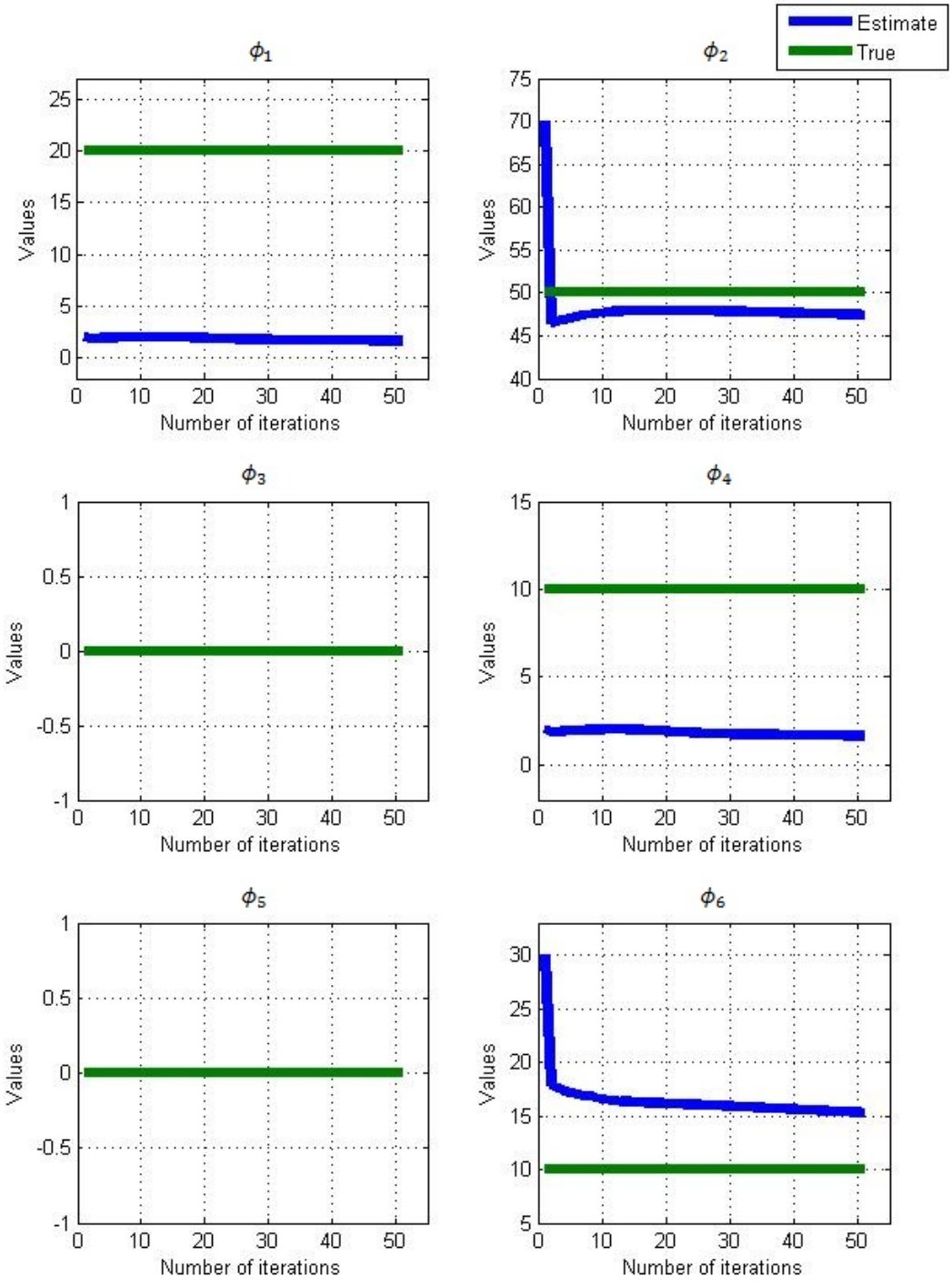

Figure 4.7: Estimation result versus number of iterations in EM estimating of case D part one

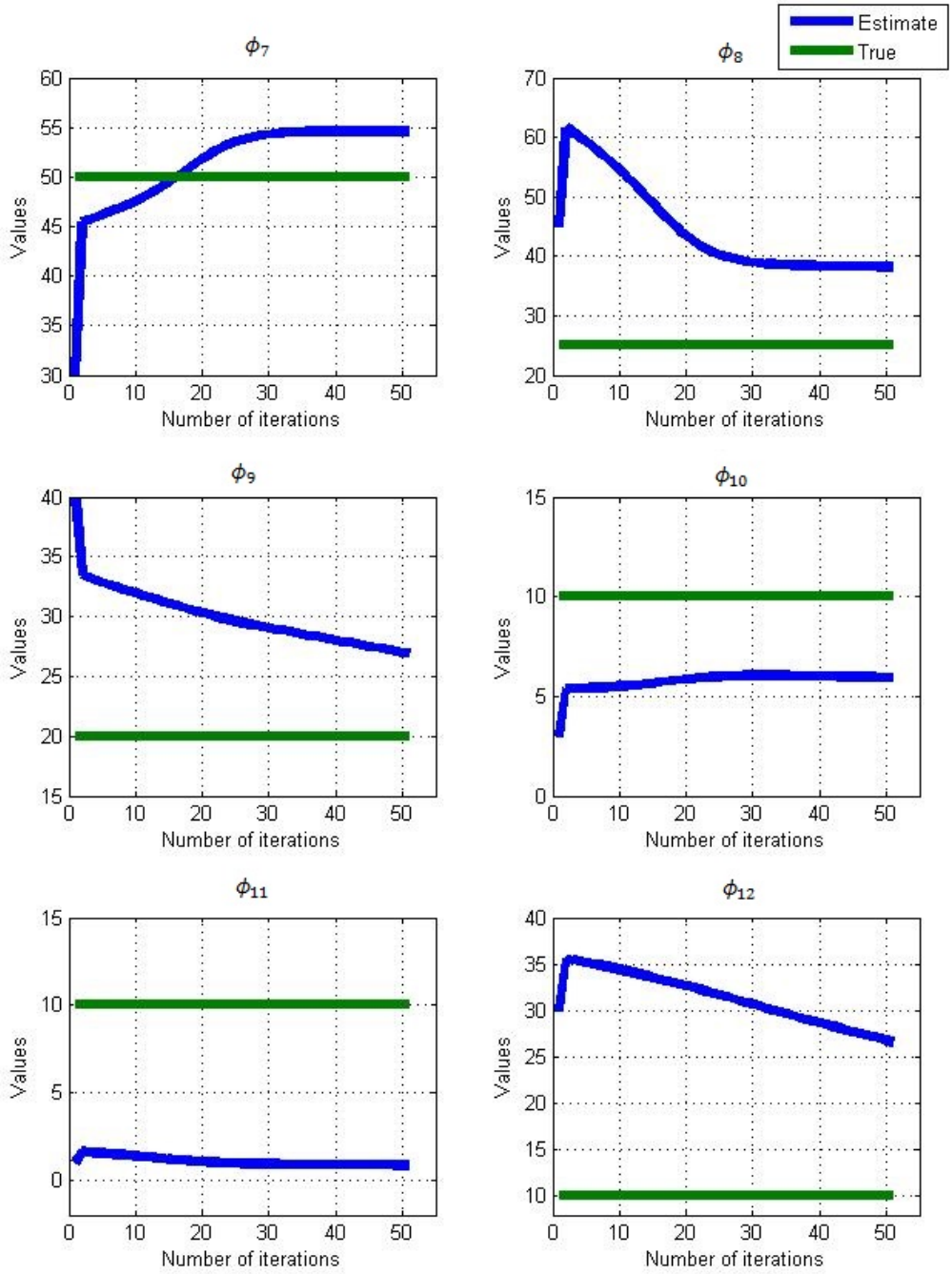

Figure 4.8: Estimation result versus number of iterations in EM estimating of case D part two

Table 4.4: Bias and MSE in EM estimating a bivariate Markov chain of case D

Case D	θ_1	θ_2	θ_3	θ_4
True Values	-70	-55	-60	-30
Initial Values	-36	-48	-100	-70
Bias	7.53	10.93	-17.12	-14.03
σ^2	59.11	125.61	295.10	200.20

4.4.2 Numerical Results for Recursive Algorithm

In this algorithm, we use the same true values and initial values as we used in the EM algorithm. Similarly to the EM algorithm, we just present the parameters that are located on the diagonal of the generator matrix G, which are $\theta_1 = -\phi_4 - \phi_7 - \phi_{10}$, $\theta_2 = -\phi_1 - \phi_8 - \phi_{11}$, $\theta_3 = -\phi_2 - \phi_5 - \phi_{12}$ and $\theta_4 = -\phi_3 - \phi_6 - \phi_9$. The performance of recursive algorithm is affected by some factors other than initial values. Those factors are the block size of the data $n+1$, and the two parameters of convergence factor γ_0 and α. We present the results for $n+1 = 5, 10, 20$, $\gamma_0 = 1, 0.1$, and $\alpha = 1, 0.7$ with the four cases of different initial values. Tables 4.5-4.8 show the numerical results for the four cases and different factors. Similarly to the EM algorithm, the bias shown in the tables represents the difference between the mean of the estimates and the true parameters, i.e., $\sum_{l=1}^{n}(\hat{\phi}_l - \phi_0)/n$. σ^2 is the mean squared error in this estimation, $\sum_{l=1}^{n}(\hat{\phi}_l - \phi_0)^2/n$, where n is the number of observations. Table 4.9-4.12 show the averaged results, after averaging, the mean square error σ^2 is decreased. The boundary here is $\phi_l \in [0, \infty)$, for $l = 1, \ldots, \tilde{r}$. Figures 4.9-4.16 show the estimation result versus the number of blocks of one replication. We could see the trend of the estimations from these figures.

4.4.3 Summary of Numerical Results

From the figures of the performance of each parameter, we could see that some parameters are always estimated better than others. This is because some of the parameters are related to the observable process, and the others are related to the underlying process. The figures

Table 4.5: Bias and MSE in recursive estimating a bivariate Markov chain of case A

Case A				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-55	-40	-50	-30
γ_0	$n+1$	α	Estimation results				
1	10	1	Bias	8.07	12.14	3.25	0.91
			σ_s^2	68.10	148.38	13.21	2.84
1	10	0.7	Bias	-3.11	-0.58	-8.62	-9.72
			σ_s^2	47.56	39.74	97.85	127.45
0.1	10	0.7	Bias	7.29	11.85	4.98	0.41
			σ_s^2	54.99	141.07	26.71	1.88
1	20	1	Bias	8.07	12.11	3.25	0.87
			σ_s^2	68.14	147.85	13.21	2.78
1	20	0.7	Bias	-2.14	1.53	-5.61	-9.08
			σ_s^2	33.15	29.20	48.04	113.60
0.1	20	0.7	Bias	7.37	11.95	6.23	0.84
			σ_s^2	55.64	143.15	40.15	1.85
1	5	1	Bias	8.07	12.14	3.25	0.91
			σ_s^2	68.13	148.39	13.22	2.84
1	5	0.7	Bias	-3.91	-2.45	-13.07	-10.55
			σ_s^2	61.14	55.37	207.17	147.16
0.1	5	0.7	Bias	7.28	11.82	3.49	0.25
			σ_s^2	55.59	140.70	14.71	2.39

Table 4.6: Bias and MSE in recursive estimating a bivariate Markov chain of case B

Case B				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-80	-65	-80	-50
γ_0	$n+1$	α	Estimation results				
1	10	1	Bias	-5.12	-2.13	-21.77	-8.40
			σ_s^2	27.81	6.70	474.61	73.09
1	10	0.7	Bias	-14.71	5.69	-19.44	-6.10
			σ_s^2	225.38	51.57	393.11	58.99
0.1	10	0.7	Bias	-4.83	-2.88	-20.44	-4.85
			σ_s^2	24.54	9.66	418.00	24.48
1	20	1	Bias	-5.12	-2.12	-21.76	-8.40
			σ_s^2	27.81	6.68	474.59	73.14
1	20	0.7	Bias	-15.26	4.28	-21.51	-10.37
			σ_s^2	241.49	34.12	471.65	132.44
0.1	20	0.7	Bias	-5.35	-3.83	-20.06	-4.68
			σ_s^2	29.62	15.74	402.67	23.02
1	5	1	Bias	-5.13	-2.13	-21.76	-8.40
			σ_s^2	27.85	6.72	474.54	73.05
1	5	0.7	Bias	-14.03	4.90	-17.41	-5.79
			σ_s^2	208.57	48.54	326.15	48.09
0.1	5	0.7	Bias	-4.46	-1.91	-21.09	-6.34
			σ_s^2	21.33	5.45	445.54	41.99

Table 4.7: Bias and MSE in recursive estimating a bivariate Markov chain of case C

Case C				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-120	-8	-100	-3
γ_0	$n+1$	α	Estimation results				
1	10	1	Bias	-23.86	-9.54	-31.55	-0.83
			σ_s^2	575.39	96.82	996.63	3.75
1	10	0.7	Bias	-9.53	-67.41	-18.50	-39.16
			σ_s^2	194.53	4715.32	458.74	1963.67
0.1	10	0.7	Bias	-26.38	-26.49	-29.36	-1.00
			σ_s^2	697.62	706.27	863.48	5.73
1	20	1	Bias	-23.62	-9.65	-31.56	-1.15
			σ_s^2	564.43	98.97	997.65	4.74
1	20	0.7	Bias	-16.24	-66.11	-14.83	-29.83
			σ_s^2	305.94	4531.60	284.98	976.49
0.1	20	0.7	Bias	-31.13	-23.29	-31.03	-5.29
			σ_s^2	970.07	544.93	963.80	34.53
1	5	1	Bias	-23.49	-9.57	-30.98	-0.26
			σ_s^2	557.51	97.59	960.75	2.82
1	5	0.7	Bias	0.41	-67.15	-19.13	-40.13
			σ_s^2	99.15	4800.22	654.16	2482.94
0.1	5	0.7	Bias	-20.59	-28.70	-28.27	2.36
			σ_s^2	427.09	829.78	800.66	9.67

Table 4.8: Bias and MSE in recursive estimating a bivariate Markov chain of case D

Case D				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-36	-48	-100	-70
γ_0	$n+1$	α	Estimation results				
1	10	1	Bias	1.32	1.74	-32.60	-15.74
			σ_s^2	5.93	4.29	1063.56	250.63
1	10	0.7	Bias	-11.82	-2.82	-38.21	-25.00
			σ_s^2	152.35	33.13	1479.19	654.31
0.1	10	0.7	Bias	-6.45	-1.24	-30.56	-8.78
			σ_s^2	44.54	2.03	934.30	77.82
1	20	1	Bias	1.28	1.76	-32.60	-15.74
			σ_s^2	5.84	4.35	1063.73	250.76
1	20	0.7	Bias	-10.48	-2.67	-35.04	-26.11
			σ_s^2	120.80	41.63	1237.32	706.05
0.1	20	0.7	Bias	-4.37	-1.85	-30.90	-9.42
			σ_s^2	20.45	3.86	955.00	89.48
1	5	1	Bias	1.35	1.73	-32.59	-15.73
			σ_s^2	6.06	4.26	1063.14	250.34
1	5	0.7	Bias	-6.62	-4.13	-39.55	-15.66
			σ_s^2	6.06	4.26	1063.14	250.34
0.1	5	0.7	Bias	-6.92	0.61	-30.70	-11.81
			σ_s^2	95.79	80.05	1599.73	269.61

Table 4.9: Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case A

Case A				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-55	-40	-50	-30
γ_0	$n+1$	α	Averaged estimation results				
1	10	1	Bias	8.68	12.53	6.40	1.05
			σ_s^2	77.16	157.74	42.61	1.93
1	10	0.7	Bias	1.61	6.45	-4.07	-4.46
			σ_s^2	18.83	48.32	25.75	26.29
0.1	10	0.7	Bias	9.07	12.77	7.82	1.34
			σ_s^2	83.08	163.32	61.94	2.20
1	20	1	Bias	8.36	12.37	6.10	0.98
			σ_s^2	71.95	153.73	38.96	1.90
1	20	0.7	Bias	2.44	7.63	-3.66	-3.71
			σ_s^2	20.67	63.74	21.97	20.37
0.1	20	0.7	Bias	9.04	12.79	8.13	1.55
			σ_s^2	82.45	163.75	66.82	2.83
1	5	1	Bias	8.89	12.62	6.56	1.04
			σ_s^2	80.76	159.97	44.56	1.81
1	5	0.7	Bias	0.67	5.03	-5.39	-5.35
			σ_s^2	19.36	34.18	40.59	35.23
0.1	5	0.7	Bias	8.96	12.70	7.32	1.14
			σ_s^2	81.30	161.54	54.50	1.76

Table 4.10: Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case B

Case B				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-80	-65	-80	-50
γ_0	$n+1$	α	Averaged estimation results				
1	10	1	Bias	-6.25	-4.75	-20.42	-8.05
			σ_s^2	40.23	23.91	417.35	65.78
1	10	0.7	Bias	-11.40	0.92	-21.99	-8.27
			σ_s^2	133.83	7.00	488.05	72.20
0.1	10	0.7	Bias	-6.98	-6.09	-19.92	-9.11
			σ_s^2	49.40	37.77	396.91	84.00
1	20	1	Bias	-5.99	-4.34	-20.49	-7.35
			σ_s^2	37.17	20.32	420.16	55.04
1	20	0.7	Bias	-11.20	0.50	-22.48	-8.48
			σ_s^2	129.76	6.82	509.44	75.14
0.1	20	0.7	Bias	-7.15	-6.38	-19.84	-9.23
			σ_s^2	51.75	41.21	393.80	86.40
1	5	1	Bias	-6.40	-4.97	-20.39	-8.50
			σ_s^2	42.03	25.94	416.17	73.11
1	5	0.7	Bias	-11.73	1.35	-21.38	-7.97
			σ_s^2	141.73	7.19	462.66	68.31
0.1	5	0.7	Bias	-6.71	-5.64	-20.06	-8.88
			σ_s^2	45.81	32.54	402.45	79.72

Table 4.11: Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case C

Case C				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-120	-8	-100	-3
γ_0	$n+1$	α	Averaged estimation results				
1	10	1	Bias	-34.79	-5.25	-34.22	-1.71
			σ_s^2	1212.91	30.63	1171.66	4.37
1	10	0.7	Bias	-9.76	-48.14	-16.78	-24.02
			σ_s^2	114.63	2388.84	320.25	670.12
0.1	10	0.7	Bias	-37.95	-8.30	-33.78	-13.24
			σ_s^2	1441.17	70.48	1141.50	179.24
1	20	1	Bias	-33.51	-7.28	-33.86	-0.85
			σ_s^2	1125.79	56.40	1147.37	2.20
1	20	0.7	Bias	-6.92	-47.41	-18.17	-21.25
			σ_s^2	58.88	2348.26	348.94	480.05
0.1	20	0.7	Bias	-39.16	-7.77	-34.32	-15.11
			σ_s^2	1533.88	61.62	1177.99	233.10
1	5	1	Bias	-35.19	-3.94	-33.95	-1.94
			σ_s^2	1240.66	18.26	1153.46	4.44
1	5	0.7	Bias	-10.13	-49.46	-17.51	-27.33
			σ_s^2	129.44	2536.49	361.58	923.68
0.1	5	0.7	Bias	-36.20	-9.47	-33.37	-9.68
			σ_s^2	1311.43	91.79	1114.36	98.71

Table 4.12: Averaged result of bias and MSE in recursive estimating a bivariate Markov chain of case D

Case D				θ_1	$\theta 2$	$\theta 3$	θ_4
True value				-70	-55	-60	-30
Initial value				-36	-48	-100	-70
γ_0	$n+1$	α	Averaged estimation results				
1	10	1	Bias	4.21	0.04	-33.06	-15.88
			σ_s^2	20.16	0.36	1093.59	252.99
1	10	0.7	Bias	-5.69	6.18	-31.25	-23.02
			σ_s^2	42.10	39.91	984.23	539.94
0.1	10	0.7	Bias	5.12	0.06	-33.95	-19.28
			σ_s^2	27.44	0.13	1153.04	372.80
1	20	1	Bias	2.73	-0.25	-32.60	-14.23
			σ_s^2	10.26	0.50	1063.20	203.43
1	20	0.7	Bias	-5.95	8.40	-31.39	-22.05
			σ_s^2	55.34	72.81	995.03	498.50
0.1	20	0.7	Bias	5.68	-0.03	-34.05	-20.16
			σ_s^2	32.91	0.11	1159.27	407.02
1	5	1	Bias	5.22	0.27	-33.34	-16.85
			σ_s^2	29.41	0.41	1111.73	284.69
1	5	0.7	Bias	-5.53	3.23	-32.87	-22.27
			σ_s^2	40.15	18.29	1087.59	507.98
0.1	5	0.7	Bias	4.35	0.32	-33.51	-18.77
			σ_s^2	19.88	0.28	1122.86	352.77

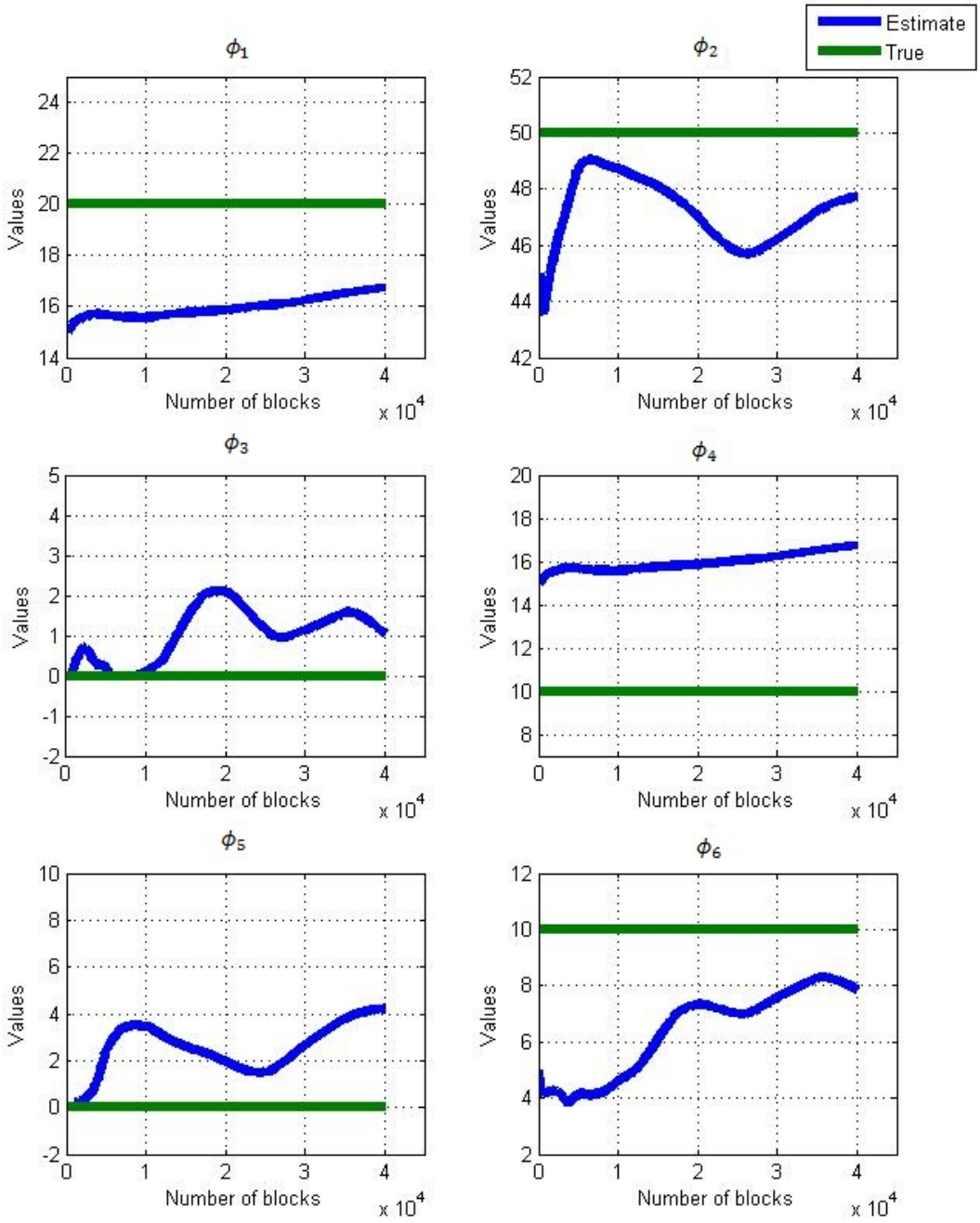

Figure 4.9: Estimation result versus number of blocks in recursive estimating of case A part one

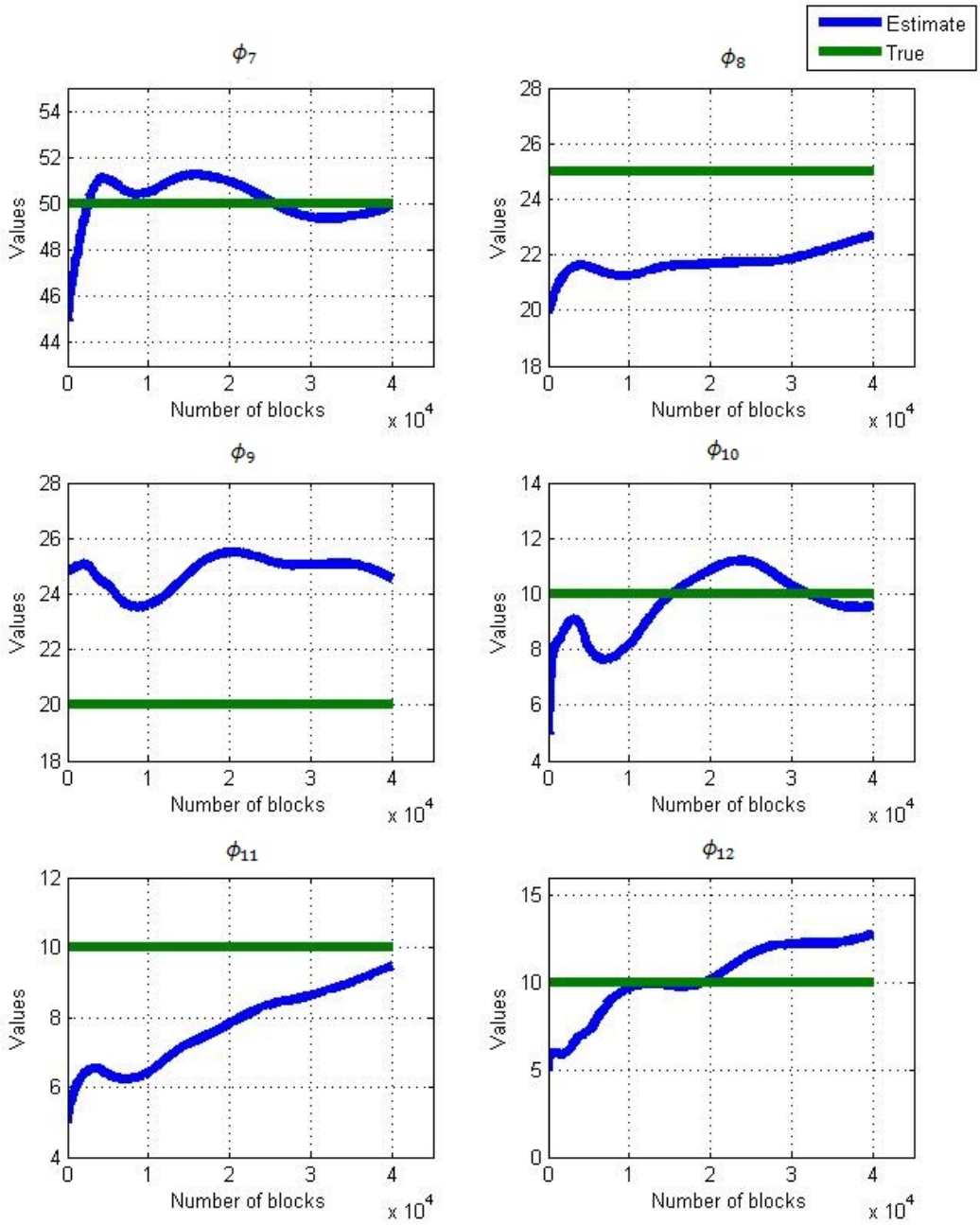

Figure 4.10: Estimation result versus number of blocks in recursive estimating of case A part two

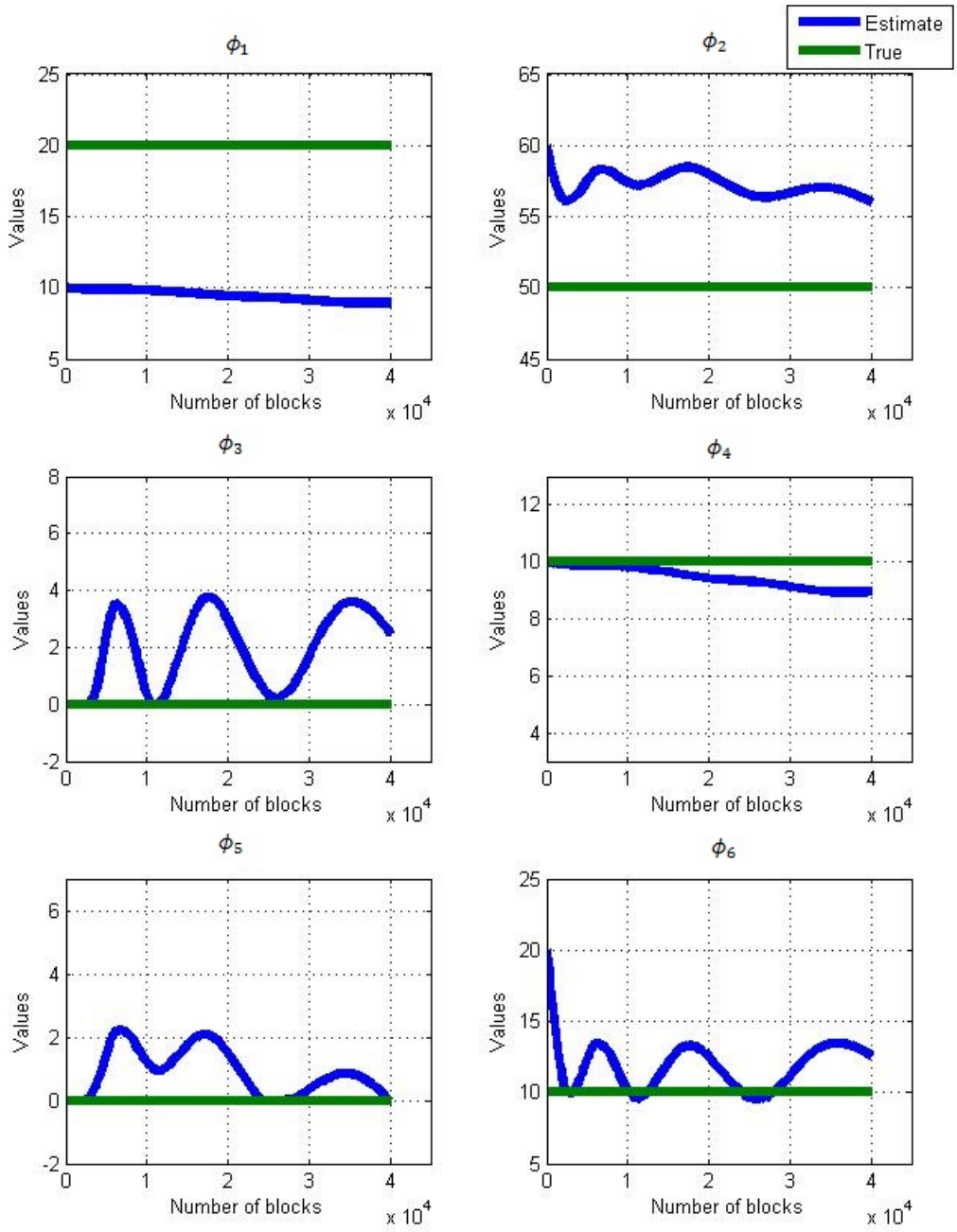

Figure 4.11: Estimation result versus number of blocks in recursive estimating of case B part one

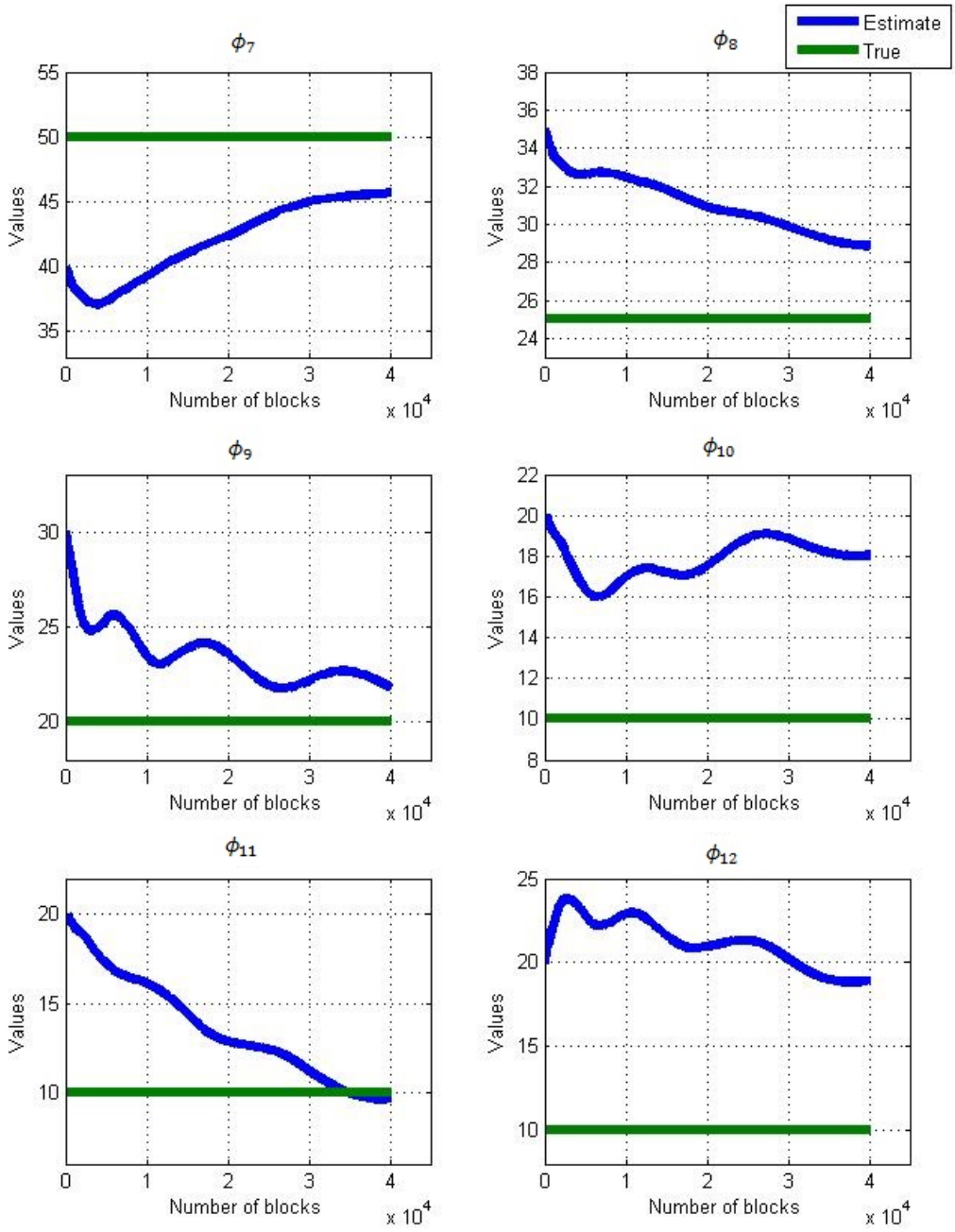

Figure 4.12: Estimation result versus number of blocks in recursive estimating of case B part two

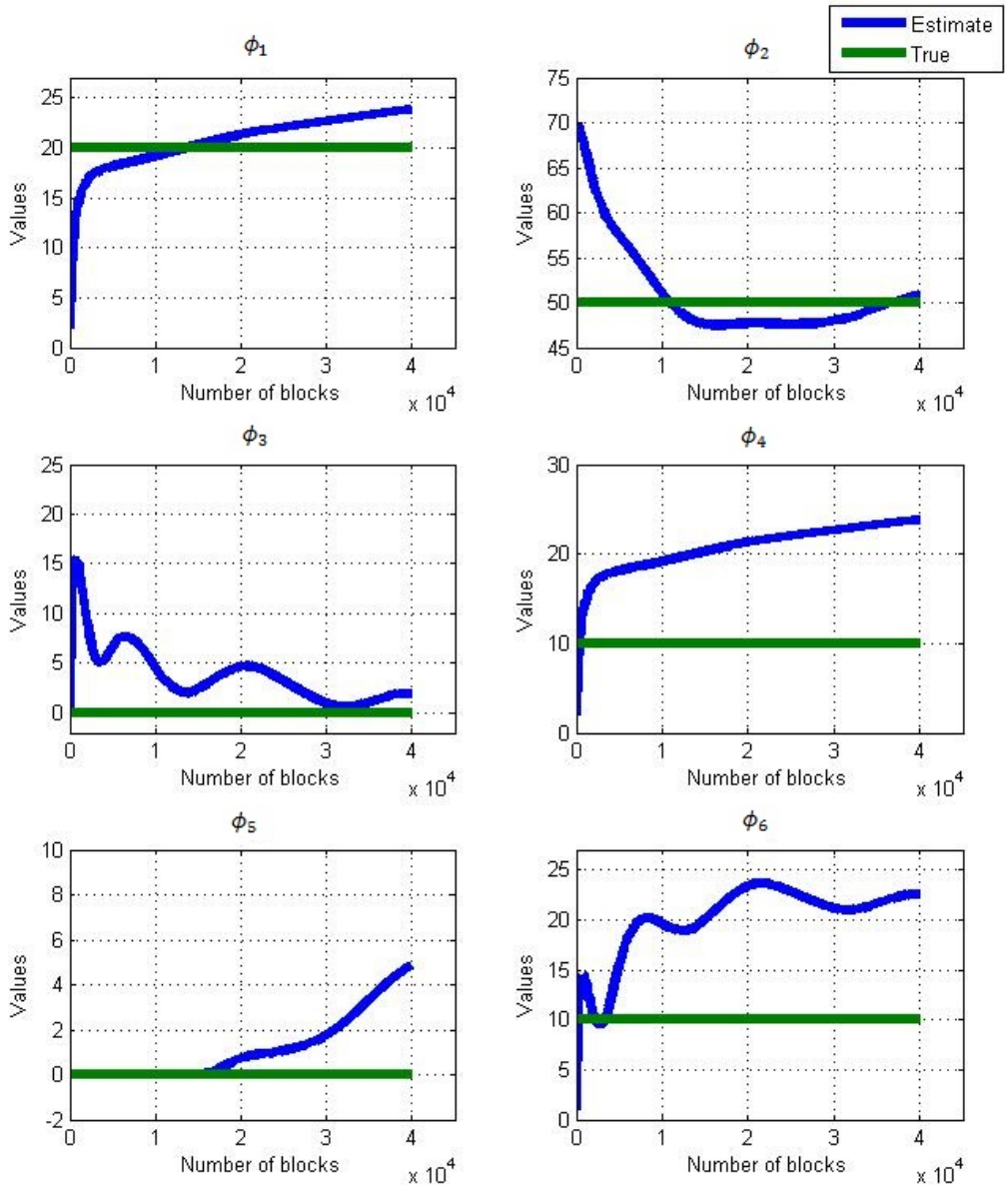

Figure 4.13: Estimation result versus number of blocks in recursive estimating of case C part one

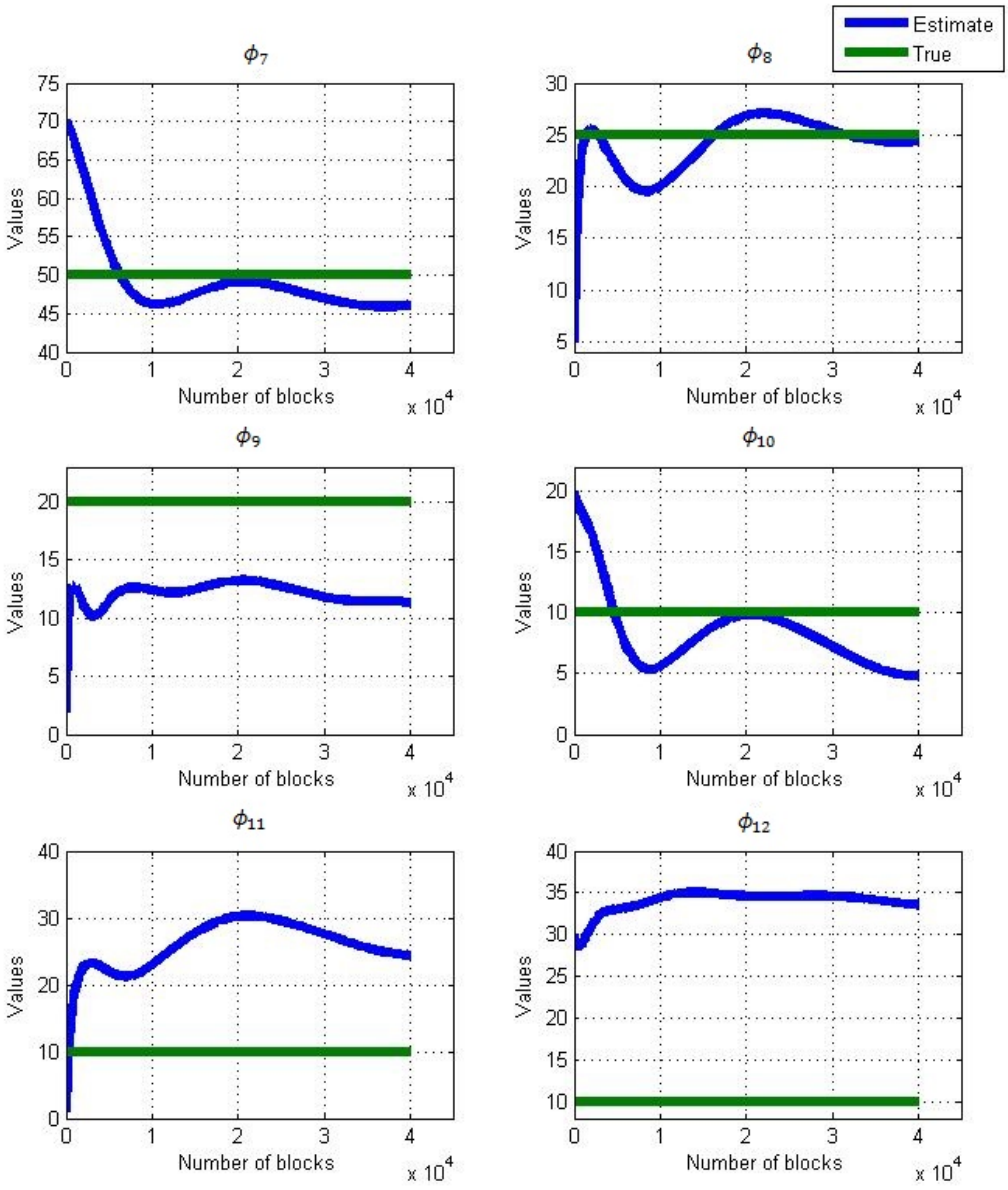

Figure 4.14: Estimation result versus number of blocks in recursive estimating of case C part two

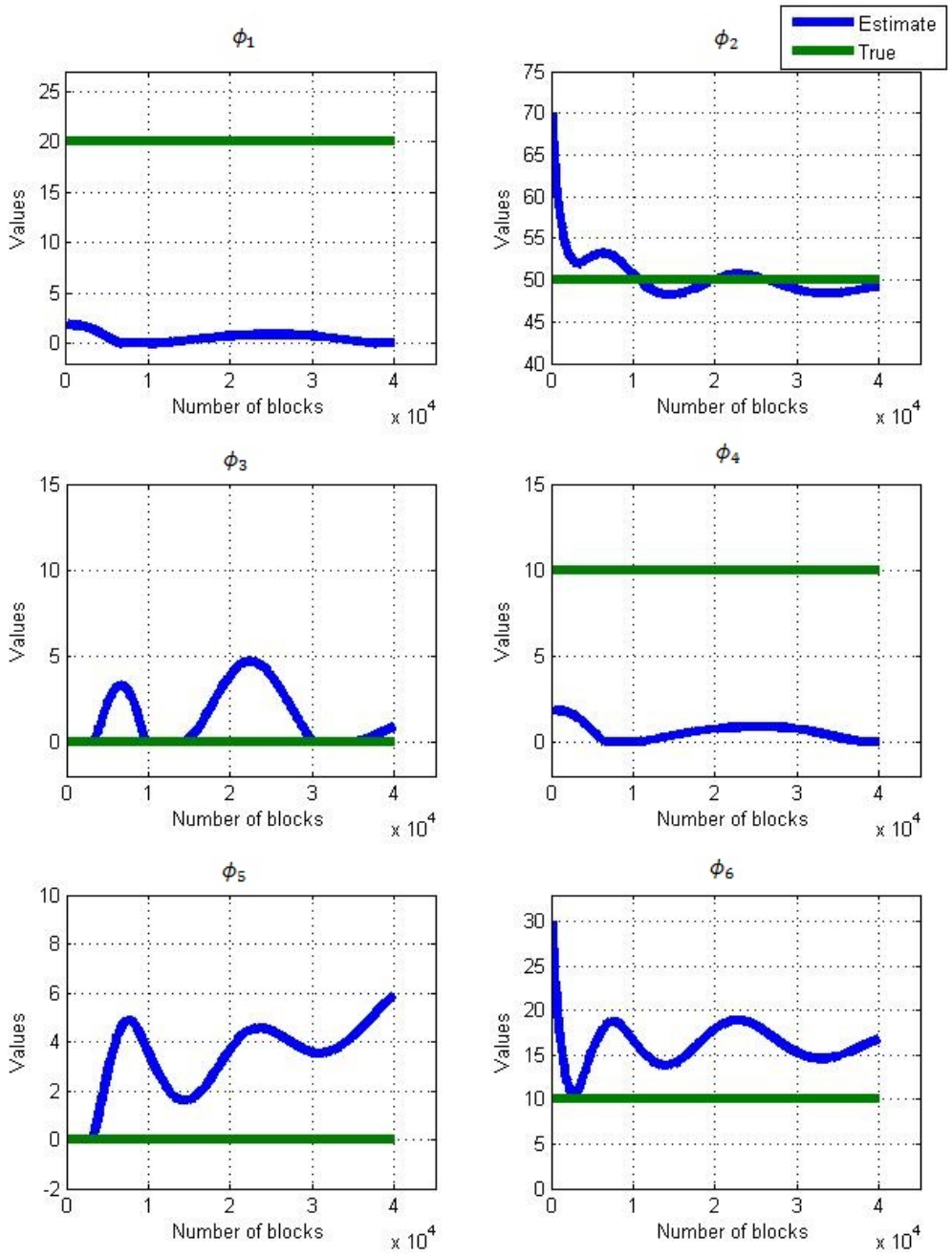

Figure 4.15: Estimation result versus number of blocks in recursive estimating of case D part one

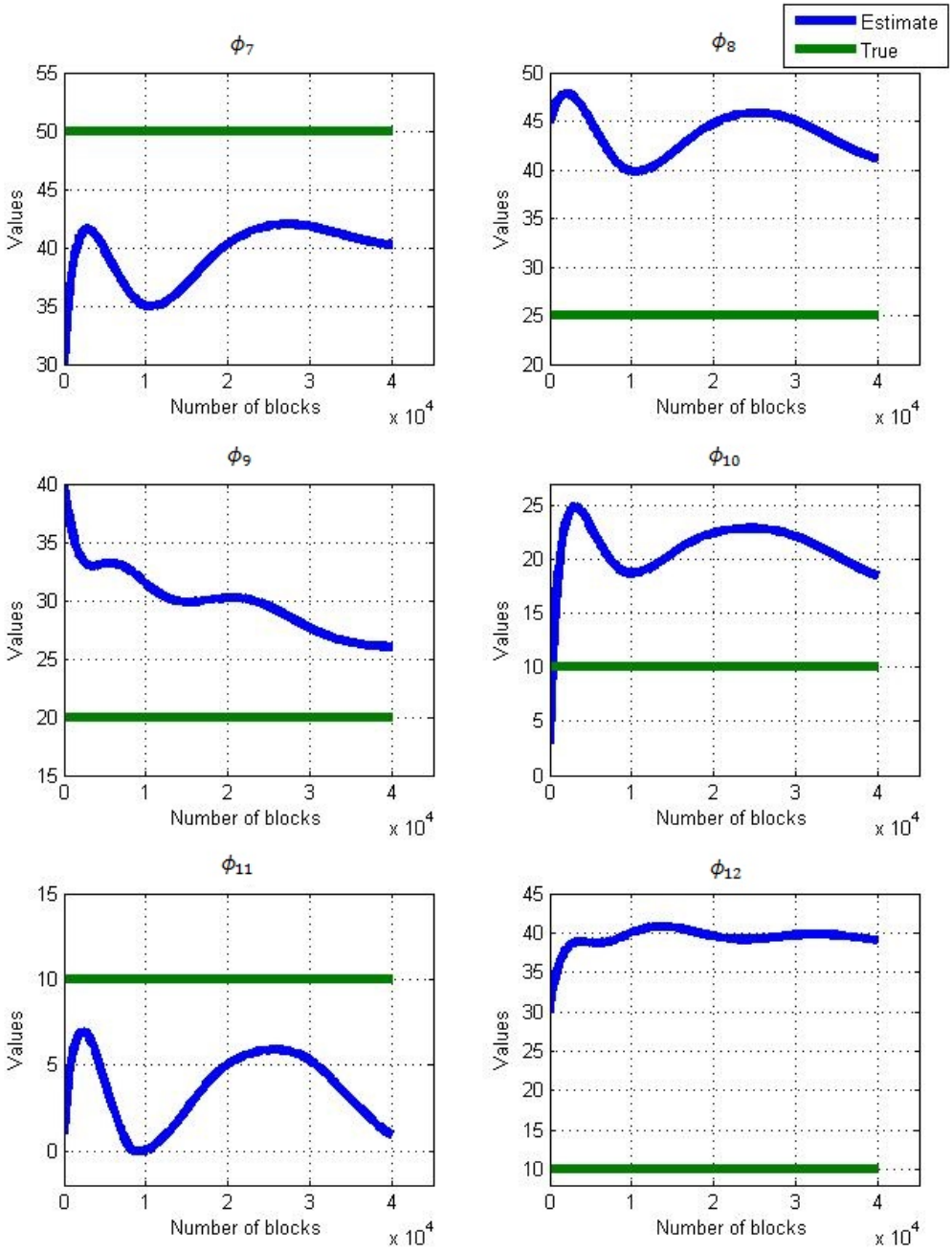

Figure 4.16: Estimation result versus number of blocks in recursive estimating of case D part two

show the estimation procedure for one replica while the tables show the numerical results of 50 replicas, so the results in the figures are not as good as the ones in the tables. But from the figures, we could see the trend of each algorithm, the EM algorithm convergences faster while the recursive algorithm converges slower. In these figures, there are only 51 iterations in the EM algorithm but there are 40,000 iterations in the recursive algorithm.

Comparing the EM algorithm with the recursive algorithm, under similar conditions to those used in the analysis of the MMPP, with the same amount of data, the EM algorithm also performs better than the recursive algorithm. The performance of the EM algorithm is weakly dependent on the initial values while the recursive algorithm is much more sensitive to the initial value. The performance of the recursive algorithm also depends on the chosen convergence factors γ_0 and α, and the block size $n+1$. Note that the block size of the recursive algorithm for the MMPP is one.

For the recursive algorithm, we presented two kinds of estimation results, original estimation result $\hat{\phi}$ and averaged result $\bar{\phi}$. We could see that most of the variance figures for the original estimation results are larger than that for the averaged results. The reason is the same as that discussed in chapter 3.

Chapter 5: Conclusion and Future Directions

The main contribution of this thesis is to provide a numerical evaluation of a recursive algorithm for bivariate Markov chain, and to compare that algorithm with the batch EM algorithm. The recursive and batch EM algorithms were applied for estimating the parameter of an MMPP and a general bivariate Markov chain. In this chapter, we will summarize our main findings and point to possible future research directions.

5.1 Conclusion

We elucidated the estimation procedures of the EM algorithm and the stochastic approximation recursive algorithm for the MMPP and the bivariate Markov chain, and provided a numerical comparison of the performances of the two approaches. Comparing the results, we found that the EM algorithm achieves relatively accurate estimates for both the MMPP and the bivariate Markov chains which is its advantage. The EM algorithm is not as sensitive to the initial parameter value, while the performance of the recursive algorithm is highly dependent on the choice of the initial parameter value assumed for the model. There is no other factors like convergence factor or the block size that could affect the estimation result of the EM algorithm while these factors affect the performance of the recursive algorithm greatly. The EM algorithm also provides a reasonable estimate using a smaller amount of data, but that data is iterated upon many times so the effective size of the data is somewhat equivalent to the number of data points times the number of iterations. In our experiments, we used 25,000 data points and 51 EM iterations, while the recursive algorithm was applied to 200,000 data points. The main advantage of the recursive algorithm is that it does not require storage of a training data, and the parameter is updated online as data becomes available.

A hybrid approach could be used in which the EM algorithm is initially applied to some data, to provide a reasonable initial estimate of the parameter, and then estimation is switched to the recursive algorithm.

For the EM algorithm and the recursive algorithm, the parameter of the observable process is easier to estimate while the parameter of the underlying process is harder to estimate. For example, parameter ϕ_1 of the bivariate Markov chain is harder to estimate than ϕ_2, regardless of the algorithm used.

5.2 Future Directions

5.2.1 Increase Number of States

We could compare with other recursive algorithms such as an algorithm which performs alternate maximization over the parameter and sufficient statistics [31]. In my research, Markov chains with only two states were studied. In the MMPP, I chose $r = 2$, and the number of parameters is $r^2 = 4$. In the bivariate Markov chain, I chose $r = d = 2$, and the number of parameters is $(r + d)^2 - (r + d) = 12$. Both of them are the simplest cases. If we increase the number of states, the number of parameters will increase greatly, and estimation would become much harder. To further test the performance of the EM algorithm and recursive algorithm for the MMPP and the general bivariate Markov chain, we should increase the number of states.

We could also increase the length of the data. For the MMPP, I used between 5,000 and 50,000 data points, each representing a jump of the observable process of the MMPP. For the bivariate Markov chain, I used between 20,000 and 200,000 data points, each representing a jump of the observable process of the bivariate Markov chain. The convergence speed of the EM algorithm is much faster than that of the recursive algorithm. Increasing the length of the data will improve the accuracy of the parameter estimate obtained by the recursive algorithm, and to a lesser extent, the accuracy of the estimate obtained by the EM algorithm. The number of iterations affect the performance of the EM algorithm a

lot which we can see from Figures 4.1-4.8. Asymptotic analysis of the performance of the algorithm when the number of observations goes to infinity. Such analysis was done for the HMM and MMPP by Rydén [4].

The main problem I have encountered is the computation time. To test the performance of the algorithms, I always had to increase the length of data, but this also increased the computation time. If we would like to further increase the length of data, we need to take the computation time into our consideration.

5.2.2 Estimation with Actual Data

In my research, the data was generated by a computer simulation, and the parameter values were either determined arbitrarily or taken from [8] and [1]. It would be interesting to apply and compare these algorithms on real data obtained in a specific application such as network traffic modeling or ion-channel characterization [32], [20].

Bibliography

Bibliography

[1] B. L. Mark and Y. Ephraim, "An EM algorithm for continuous-time bivariate Markov chains," *Computational Statistics and Data Analysis*, vol. 57, pp. 504–517, 2013.

[2] P. J. Bickel, Y. Ritov, and T. Rydén, "Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models," *Ann. of Statist*, vol. 26, no. 4, pp. 1614–1635, 1998.

[3] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1518–1569, 2002.

[4] T. Rydén, "On recursive estimation for hidden Markov models," *Stochastic Processes and their Applications*, vol. 66, pp. 79–96, 1997.

[5] W. Fischer and K. Meier-Hellstern, "The Markov-modulated Poisson process (mmpp) cookbook," *Perform. Eval.*, vol. 18, pp. 149–171, 1992.

[6] K. S. Meier, "A statistical procedure for fitting Markov-modulated Poisson processes," Ph.D. Dissertation, University of Delaware, 1984.

[7] T. Rydén, "Parameter estimation for Markov modulated Poisson processes," *Commun. Statist. Stoch. Models*, vol. 10, no. 4, pp. 795–829, 1994.

[8] ——, "An EM algorithm for estimation in Markov-modulated Poisson processes," *Computational Statistics and Data Analysis*, vol. 21, pp. 431–447, 1996.

[9] G. Lindgren and U. Holst, "Recursive estimation of parameters in Markov-modulated Poisson processes," *IEEE Trans. Commun.*, vol. 43, no. 11, pp. 2812–2820, Nov. 1995.

[10] U. Holst and G. Lindgren, "Recursive estimation in mixture models with Markov regime," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1683–1690, Nov. 1991.

[11] Y. Ephraim and B. L. Mark, "Bivariate Markov processes and their estimation," *Foundations and Trends in Signal Processing*, vol. 6, no. 1, pp. 1–95, 2013.

[12] M. Rausand and A. Høyland, *System Reliability Theory: Models, Statistical Methods, and Applications*. Wiley-Interscience, 2004.

[13] S. M. Ross, *Introduction to probability models*. Boston, MA: Academic Press, 2003.

[14] E. Çinlar, "Markov renewal theory: A survey," *Management Science*, vol. 21, no. 7, pp. 727–752, 1975.

[15] W. J. J. Roberts, Y. Ephraim, and E. Dieguez, "On Rydén's EM algorithm for estimating MMPPs," *IEEE Sig. Proc. Let.*, vol. 13, pp. 373–377, 2006.

[16] D. S. Freed and L. A. Shepp, "A Poisson process whose rate is a hidden Markov chain," *Adv. Appl. Prob.*, vol. 14, pp. 21–36, 1982.

[17] P. Todorovic, *An Introduction to Stochastic Processes and Their Applications.* New York: Springer-Verlag, 1994.

[18] L. Breiman, *Probability.* SIAM, Philadelphia, 1992.

[19] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes.* Oxford: Oxford Science Publications, 2001.

[20] D. Colquhoun and A. G. Hawkes, "On the stochastic properties of bursts of single ion channel openings and of clusters of bursts," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 300, no. 1098, pp. 1–59, 1982.

[21] F. Qin, A. Auerbach, and F. Sachs, "Maximum likelihood estimation of aggregated Markov processes," *Proc. Royal Society Lond. B*, vol. 264, pp. 375–383, 1997.

[22] Y. Ephraim and B. L. Mark, "Explicit forward recursive estimators for Markov modulated Markov processes," *Stoch. Models*, vol. 28, no. 3, pp. 359–387, 2012.

[23] L. R. R. S. E. Levinson and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, pp. 1035–1074, 1983.

[24] P. A. Devijver, "Baum's forward-backward algorithm revisited," *Pattern Recognit. Lett.*, vol. 3, pp. 369–373, 1985.

[25] C. F. Van Loan, "Computing integrals involving the matrix exponential," *IEEE Trans. Automat. Control*, vol. 23, pp. 395–404, 1978.

[26] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty five years later," *SIAM Review*, vol. 45, pp. 3–49, 2003.

[27] C. J. Willy, W. J. J. Roberts, T. A. Mazzuchi, and S. Sarkani, "Recursions for the MMPP score vector and observed information matrix," *Stoch. Models*, vol. 26, no. 4, pp. 649–665, 2010.

[28] D. R. Fredkin and J. A. Rice, "On aggregated Markov processes," *J. Appl. Prob.*, vol. 23, no. 1, pp. 208–214, 1986.

[29] W. J. Vetter, "Derivative operations on matrices," *IEEE Trans. Automatic Control*, vol. AC-15, pp. 241–244, April 1970.

[30] B. Chen and P. A. Zadrozny, "Analytic derivatives of the matrix exponential for estimation of linear continuous-time models," *Journal of Economic Dynamics & Control*, vol. 25, pp. 1867–1879, 2001.

[31] C. Stiller and G. Radons, "Online estimation of hidden Markov models," *IEEE Sig. Proc. Let.*, vol. 6, no. 8, pp. 213–215, 1999.

[32] D. M. Lucantoni, "New results on the single server queue with a batch Markovian arrival process," *Stochastic Models*, vol. 7, no. 1, pp. 1–46, 1991.

Curriculum Vitae

Zhuxuan Li was born in Beijing, China. She graduated from Beijing No. 4 high school, then attended Tsinghua University in 2007 and earned her BS in Electrical Engineering in 2011. Then she came to the US to continue her graduate study. She is now an MS student in the department of Electrical and Computer Engineering at George Mason University. Her thesis title is "Recursive parameter estimation for continuous-time bivariate Markov chains", which was conducted under the supervision of Professor Yariv Ephraim. Her research interests are in bivariate Markov chains, and in estimation of their parameters using recursive algorithms and the EM algorithm.