

Parameter Estimation of Queueing System using Mixture Model and the EM  
Algorithm

by

Hang Li

B.Sc., Beijing University of Posts and Telecommunications, 2011

M.Sc., Beijing University of Posts and Telecommunications, 2014

A Thesis Submitted in Partial Fulfillment of the  
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

© Hang Li, 2016

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by  
photocopying or other means, without the permission of the author.

Parameter Estimation of Queueing System using Mixture Model and the EM  
Algorithm

by

Hang Li

B.Sc., Beijing University of Posts and Telecommunications, 2011

M.Sc., Beijing University of Posts and Telecommunications, 2014

Supervisory Committee

---

Dr. Kui Wu, Supervisor  
(Department of Computer Science)

---

Dr. Venkatesh Srinivasan, Departmental Member  
(Department of Computer Science)

## Supervisory Committee

---

Dr. Kui Wu, Supervisor  
(Department of Computer Science)

---

Dr. Venkatesh Srinivasan, Departmental Member  
(Department of Computer Science)

### ABSTRACT

Parameter estimation is a long-lasting topic in queueing systems and has attracted considerable attention from both academia and industry. In this thesis, we design a parameter estimation framework for a tandem queueing system that collects end-to-end measurement data and utilizes the finite mixture model for the maximum likelihood (ML) estimation. The likelihood equations produced by ML are then solved by the iterative expectation-maximization (EM) algorithm, a powerful algorithm for parameter estimation in scenarios involving complicated distributions.

We carry out a set of experiments with different parameter settings to test the performance of the proposed framework. Experimental results show that our method performs well for tandem queueing systems, in which the constituent nodes' service time follow distributions governed by exponential family. Under this framework, both the Newton-Raphson (NR) algorithm and the EM algorithm could be applied. The EM algorithm, however, is recommended due to its ease of implementation and lower computational overhead.

# Contents

<b>Supervisory Committee</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Acronyms</b>	<b>viii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>Dedication</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivating Examples . . . . .	1
1.2 Contributions of this Thesis . . . . .	4
1.2.1 Model the End-to-end Measurement Data . . . . .	5
1.2.2 Parameter Estimation of Mixture Model . . . . .	5
1.2.3 Simulation Experiments of Tandem Queuing System . . . . .	5
1.3 Outline . . . . .	6
<b>2 Background and Related Work</b>	<b>7</b>
2.1 Networks of Queues . . . . .	7
2.1.1 Single Queueing System . . . . .	8
2.1.2 Tandem Open Network . . . . .	8
2.2 Finite Mixture Model and Parameter Estimation . . . . .	9
2.2.1 Model Description . . . . .	9

2.2.2	ML Estimation . . . . .	10
2.2.3	EM Algorithm . . . . .	11
2.3	Previous Work . . . . .	15
<b>3</b>	<b>Parameter Estimation for Tandem Queueing Network</b>	<b>17</b>
3.1	Data Collection Procedure . . . . .	17
3.2	Model of Service time . . . . .	18
3.2.1	The Exponential Family of Distributions . . . . .	18
3.2.2	Density of the Sum . . . . .	19
3.3	The Likelihood Function . . . . .	20
3.4	EM Algorithm . . . . .	21
3.5	Modified EM Algorithm . . . . .	24
3.6	Illustrative Examples . . . . .	26
3.6.1	Parameter Estimation of Exponential Service Times . . . . .	26
3.6.2	Interchangeability of the Service Time . . . . .	27
3.6.3	Different Types of Service Times . . . . .	28
3.7	Tandem Queueing System with n Nodes . . . . .	29
3.7.1	PDF of the Sum of n Independent Random Variables . . . . .	30
3.7.2	Tandem Queueing System with Three Servers . . . . .	30
3.7.3	Interchangeability of Service Time in Tandem Queueing Network	33
3.8	Comparison with Basawa's Work . . . . .	34
3.9	Further Discussion . . . . .	34
<b>4</b>	<b>Experiments</b>	<b>36</b>
4.1	Data Acquisition . . . . .	36
4.1.1	Simulation of the Queueing System . . . . .	36
4.1.2	Sampling Plans . . . . .	38
4.2	Parameter Estimation Experiments . . . . .	39
4.2.1	Newton-Raphson Method . . . . .	39
4.2.2	Simulation Results . . . . .	41
4.3	Comparison Between Modified EM and NR . . . . .	54
<b>5</b>	<b>Conclusions and Future Work</b>	<b>55</b>
	<b>Bibliography</b>	<b>57</b>

# List of Tables

Table 3.1	Examples of exponential family distributions . . . . .	19
Table 4.1	An example of event list . . . . .	37
Table 4.2	True values of the parameters . . . . .	41
Table 4.3	Estimation of $\phi_1$ with different N values . . . . .	45
Table 4.4	Estimation of $\phi_2$ with different N values . . . . .	45
Table 4.5	Estimation of $(\phi_1, \phi_2)$ with different N values . . . . .	48
Table 4.6	Estimation of $\phi_1$ with different m values when N=1000 . . . . .	50
Table 4.7	Estimation of $\phi_2$ with different m values when N=1000 . . . . .	50

# List of Figures

Figure 1.1 NFV architecture [43] . . . . .	2
Figure 1.2 Virtualized environment representation [17] . . . . .	3
Figure 1.3 ISP's price vs. service speed [33] . . . . .	3
Figure 1.4 Priority queuing service [36] . . . . .	4
Figure 2.1 Single queueing system . . . . .	8
Figure 2.2 Tandem open network with two servers . . . . .	9
Figure 2.3 EM for data clustering[45] . . . . .	15
Figure 3.1 Two-stage tandem queueing system . . . . .	17
Figure 3.2 Tandem queueing system consisting of n servers . . . . .	29
Figure 4.1 Queueing system with two queues and two servers . . . . .	36
Figure 4.2 Parameter estimation of exponential service rates . . . . .	43
Figure 4.3 Parameter estimation of exponential and lognormal service rates	43
Figure 4.4 Parameter estimation of queueing system with three exponential servers . . . . .	44
Figure 4.5 Approximations of pdf with different lengths of observations . .	47
Figure 4.6 Comparison between NR and EM for different lengths of obser- vations . . . . .	49
Figure 4.7 Error with different m value . . . . .	51
Figure 4.8 Estimation of $\phi_1$ with different m values . . . . .	52
Figure 4.9 Estimation of $\phi_2$ with different m values . . . . .	53
Figure 4.10 Comparison between NR and EM with different m values . . .	53

# List of Acronyms

<b>CAPEX</b> .....	Capital expenditure
<b>EM</b> .....	Expectation-maximization algorithm
<b>ISP</b> .....	Internet service provider
<b>MLE</b> .....	Maximum likelihood estimation
<b>NFV</b> .....	Network function virtualization
<b>NR</b> .....	Newton-Raphson method
<b>OPEX</b> .....	Operating expenditure
<b>SDN</b> .....	Software-defined networking

## ACKNOWLEDGEMENTS

I would like to thank:

**Dr. Kui Wu**, for mentoring me in my research and supporting me during my graduate study at UVic.

**Dr. Srinivasan and Dr. Shi** for serving in my thesis examining committee.

**Feng Liu**, for his dedication to supporting and encouraging me in pursuing my degree.

**My friends**, for our true and long-lasting friendship.

*To learn without thinking is labor in vain.*

*To think without learning is desolation*

Confucius

DEDICATION

**To my parents,  
and Feng.**

# Chapter 1

## Introduction

Finite mixture models are widely used in various fields where data can be considered as a combination of different groups. When dealing with such models, parameter estimation, also known as the decomposition of the components, is always of great importance, since it is the parameters that capture the underlying structure of the whole system. Statistical tools like the maximum likelihood estimation are usually utilized to estimate the parameters based on the sample data.

One application of finite mixture models is to infer the performance of the servers in a queuing network, where only the end-to-end information is available. For each packet passing through the network, we use the total service time as the sample data, which can be viewed as the sum of service time at each single queue along the path. The service rate of each individual server can then be estimated based on this end-to-end information using the finite mixture model.

There are many practical applications where such a problem may arise from. We provide several examples to reveal the practical meaning of solving the above problem.

### 1.1 Motivating Examples

Network function virtualization (NFV) and software-defined networking (SDN), as shown in Figure 1.1 and Figure 1.2, are two promising technologies for telco operators that can reduce the growing capital expenditure (CAPEX) and operating expenditure (OPEX). For such networks, resource management should be flexible enough to make the network dynamically adjust its capacity to meet the actual demands. As a basic

principle, we need to assure network service performance by estimating the real-time service rate of all the virtualized functions, locating the elements with serious congestion, and reallocating resource to these elements at the VM level. The challenge is that monitoring the performance of individual virtualized functions is not easy: they usually work as an application in virtual machines (VM), but the performance metrics of VM is from the hypervisor that only provides VM-level performance such as the utilization of vCPU and vMemory. In this case, we can only infer the performance of virtualized functions from end-to-end measurement, i.e., from the time when a packet enters a chain of virtualized functions to the time when the packet leaves the service chain. In this scenario, the mixture model can be used to model the total service time of the packets obtained by end-to-end measurements, based on which we try to infer the service rate of each virtual function.

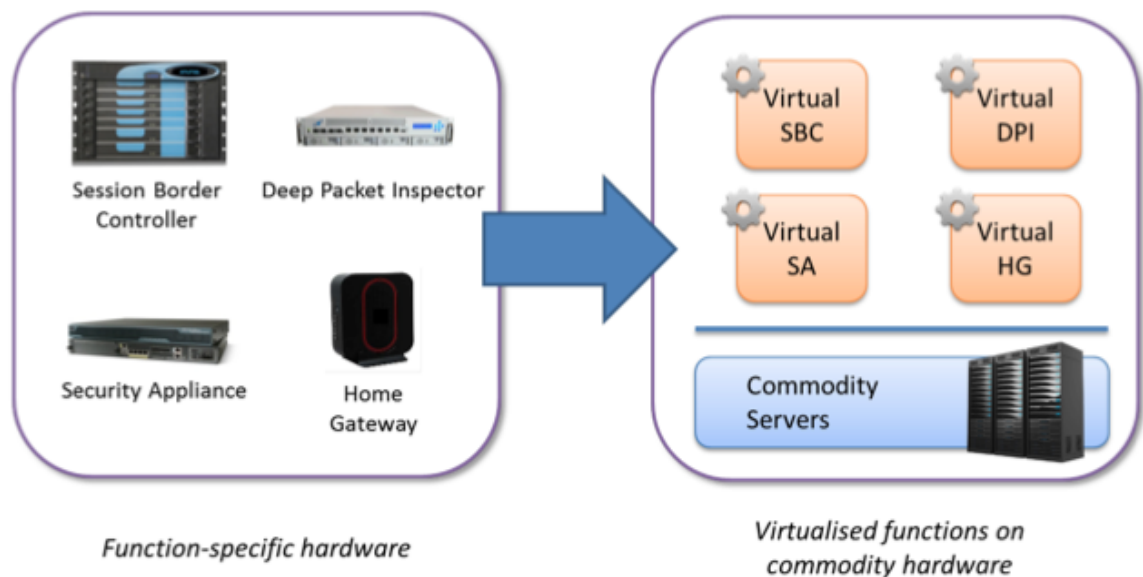


Figure 1.1: NfV architecture [43]

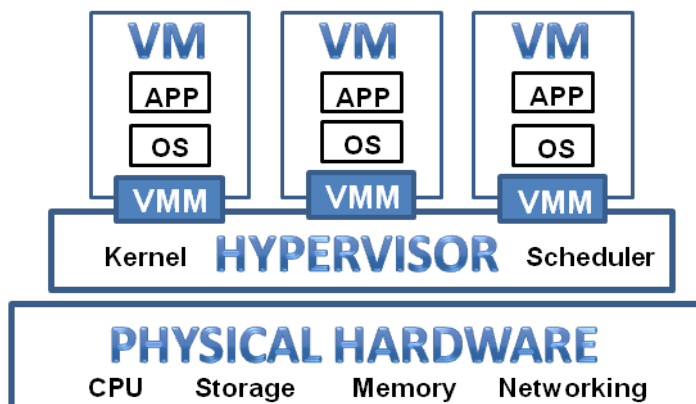


Figure 1.2: Virtualized environment representation [17]

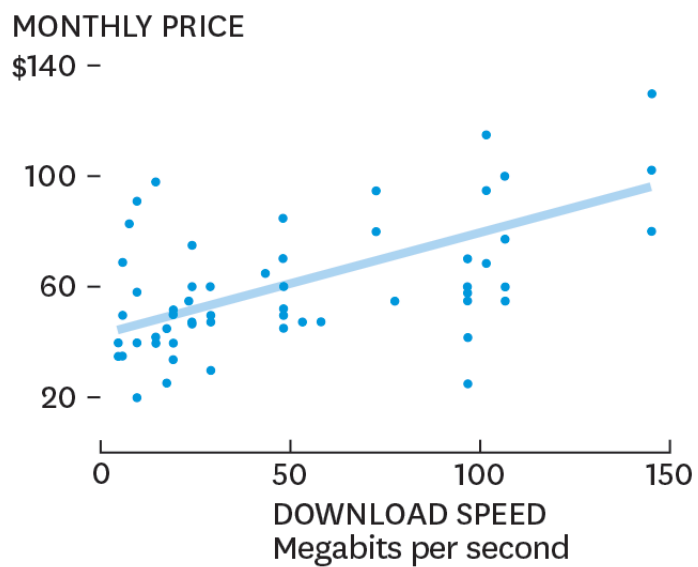


Figure 1.3: ISP's price vs. service speed [33]

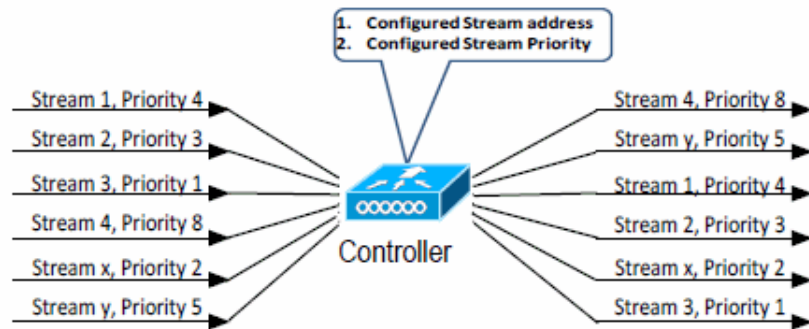


Figure 1.4: Priority queuing service [36]

Another area where the parameter estimation problem may arise is network quality of service (QoS) monitoring. Internet service providers (ISPs) always provide different service levels based on price, as shown in Figure 1.3 and Figure 1.4. Recently, a new type of ISPs, called virtual ISPs, start to gain more and more customers. A virtual ISP does not have its own network infrastructure, but depends on other traditional ISPs, also called physical ISPs (as they have own network infrastructure), to offer service to its customers. The virtual ISP pay its supporting physical ISPs for the offered service. To guarantee QoS to its customers, the virtual ISP needs to make sure its supporting physical ISPs actually provide the promised network capacity. However, physical ISPs normally do not share the performance monitoring data at link-level to the virtual ISP. In this case, the virtual ISP can monitor the end-to-end performance, based on which it can estimate the performance of physical ISPs. To be more specific, the total service time of the packets passing a particular chain of physical ISPs is captured and used to train the mixture model and estimate the service rate of each physical ISP.

## 1.2 Contributions of this Thesis

As shown in the above examples, we need a framework to infer the internal parameters based on some end-to-end measurement data. This task could be well handled by incorporating a finite mixture model.

In this thesis, we provide a parameter estimation framework that uses end-to-end performance of a network of tandem queues to estimate the service rate of each server.

The thesis includes contributions on (1) mathematical model of measurement data, (2) algorithms for parameter estimation of finite mixture model, and (3) simulation of tandem queuing system and analysis of simulation results.

### **1.2.1 Model the End-to-end Measurement Data**

We propose a scheme to record the data from end-to-end measurements and model the data with a variant of finite mixture model. For packets passing through a network of queues, we collect the total service time of each packet spent in the network. According to the nature of a tandem queuing system, we make adjustments to the finite mixture model and apply it to our service time data. Then the parameters of this model directly reflect the service rate of each server in the queuing system.

### **1.2.2 Parameter Estimation of Mixture Model**

A great deal of effort has been spent on deriving an effective parameter estimation algorithm. For finite mixture model, the iterative Expectation Maximization algorithm provides a convenient way for solving the likelihood equation. Many researchers have applied this algorithm to their mixture model, and works on this topic always adopt model whose component densities belong to the same parametric family.

In our work, however, the packets' service time of each server in a tandem queue may follow to different distributions, some to exponential and some to normal for example. From this point, a modified iterative EM algorithm is developed for our specific mixture model whose component densities could come from the entire exponential family.

### **1.2.3 Simulation Experiments of Tandem Queuing System**

Another important theme of this thesis is building a simulation platform of tandem queuing system in R, a programming language popular in statistics and machine learning. Experiments are carried out to collect the end-to-end measurement data. We then implement our algorithm in R and apply it to the total service time data to estimate the service rate of the servers. We perform simulation experiments of different scenarios to test the feasibility of the proposed framework.

## 1.3 Outline

**Chapter 1** contains some real-world applications that motivate our research in the parameter estimation of finite mixture model. Main contributions of our thesis are listed with brief descriptions.

**Chapter 2** begins with brief introduction of each component that will appear in our parameter estimation framework, including queueing system, finite mixture model, ML estimation, and EM algorithm. We then provide a summary of the previous work in this field.

**Chapter 3** introduces the details of our new research, which is also the main contribution of this thesis. It begins with descriptions of data acquisition, the mixture model, formulation of ML estimation, and the EM algorithm. Illustrative examples are then provided to show how the framework is applied to specific queueing systems.

**Chapter 4** reports the results of our simulation experiments conducted with different settings. Details of the experimental setup are provided in this chapter. It also contains the analysis of evaluation results. To be specific, it discusses the evaluation of the proposed framework and the comparison of performances between different algorithms.

**Chapter 5** summarizes the thesis and provides future work along the line of this thesis research.

## Chapter 2

# Background and Related Work

In the preceding chapter, a glimpse at our parameter estimation problem was provided in the motivating examples. In order to estimate the available resource at each node (router, for example) for a particular packet flow, we model the computer network using tandem queuing network and then apply the finite mixture model to the total service time of the packets passing through the network. The parameters of the mixture model are the service rates at each node in the network of queues that reflect the actual resource obtained by the packet flow.

This chapter will first briefly go over the concepts of network of queues and finite mixture model which are used to model the data in our work. Then the maximum likelihood estimation and the EM algorithm will be introduced as techniques for solving the parameter estimation problem in the finite mixture model. Previous works in parameter estimation of different types of queueing system will be summarized in the end.

### 2.1 Networks of Queues

Network of queues is a classical model for computer networks which provides a general framework to model the limited resource at each node in a computer network. Queueing may occur at each node when the arrival rate of packets exceeds the processing rate of the node and the packets need to wait in the buffer before being transmitted. The processing rate of nodes here is an abstract concept that describes the overall function of real-world packet processing such as extracting destination address from the packet header, routing table look-up, input-output port switching

and so forth [18][26].

### 2.1.1 Single Queueing System

As shown in Figure 2.1, the simplest queueing model is a single queueing system with one queue and one server. In the computer networking context, the items moving through the queueing system are packets of data of a particular flow passing through the nodes and lines of such networks [35]. The queue is used to model the buffer available to the flow while the server is used to model the actual capacity obtained by this flow.

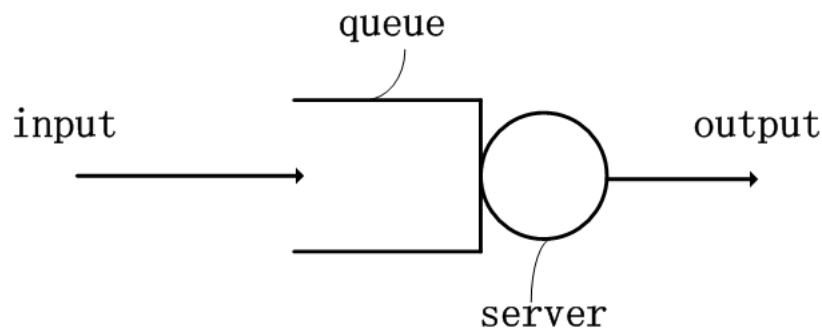


Figure 2.1: Single queueing system

The M/M/1 queueing system is the basic queueing system which assumes that the arrival process is Poisson process and the service time of the server is an exponentially distributed random variable. However, in real-world applications, service time doesn't always follow exponential distribution and some other probability distributions may be used to describe the servers' performance.

In our work, according to the property of service time, we resort to a M/G/1 system to model the nodes in the network, with an extra assumption that the distribution of service time belongs to the exponential family, due to their simplicity and their power to model real-world systems.

### 2.1.2 Tandem Open Network

A two-stage tandem network consisting of two nodes is shown in Figure 2.2. The two servers in the system have independent service rates. We use two random variables  $V_1$  and  $V_2$  to denote the service time of each server. Then the total service time of

each packet passing through this network could be denoted by  $X = V_1 + V_2$ . Based on this, for a general tandem open network consisting of  $n$  nodes, the total service time of each packet passing through this system could be denoted by  $X = V_1 + V_2 + \dots + V_n$ . Our task is to infer the service rate of each node based on the total service time. This could be considered as a variant of finite mixture model.

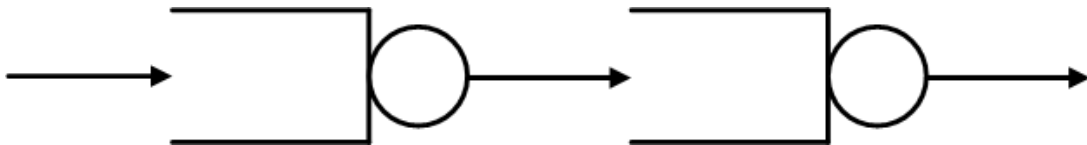


Figure 2.2: Tandem open network with two servers

## 2.2 Finite Mixture Model and Parameter Estimation

In statistics, a finite mixture model describes the presence of two or more subpopulations within the overall population, which is supposed to be a convex combination of a couple of probability density functions [44][29]. This model is a composition of several distributions and the decomposition of mixture distribution evolves estimating the parameters of each component.

### 2.2.1 Model Description

Here we follow the notations in [41] and [44] to describe the standard finite mixture model.

Suppose we have a random variable  $X$  of which the density function has the following form

$$g(x, \Psi) = \sum_{j=1}^k \pi_j(x, \theta_j).$$

Then  $X$  is said to follow a finite mixture distribution with  $k$  components and

$$\Psi = (\pi_1, \dots, \pi_k, \theta_1, \dots, \theta_k).$$

The components of the mixture model are denoted by  $f(x, \theta_j)_{j=1, \dots, k}$  where  $\theta_j$  is parameter of each component and  $\pi_j$  is the weight of each component.

In our application, for a tandem network of two nodes, the total service time was denoted by  $X = V_1 + V_2$ . We consider it as a variant of the standard mixture model which has equal weight for each component. The random variables  $V_1$  and  $V_2$ , which denote the service time at Node 1 and Node 2, respectively, follow two independent distributions  $f_{V_1}(v_1, \phi_1)$  and  $f_{V_2}(v_2, \phi_2)$ . Our task is to estimate the parameters  $\phi_1$  and  $\phi_2$  based on the sample data of the mixture model  $X$ .

## 2.2.2 ML Estimation

The maximum likelihood estimation can be used so solve the above parameter estimation problem and it's believed to have advantage over the other estimation methods like the minimum  $\chi^2$  and the Bayes's estimators [22][15][44].

Maximum likelihood estimation is a standard and general parameter estimation method in statistics [31][38]. Suppose  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  is the sample of a random variable  $X$  whose density function is denoted by  $f(x, \phi)$ . The likelihood function is the multiplication of the probability function for each sample data, which is written as

$$L(\phi, x) = \prod_{s=1}^n f(x_s, \phi).$$

Given the data  $\mathbf{x}$ , the likelihood function is a function of the parameter  $\phi$ . The principle of maximum likelihood estimation is to find a density function that makes the sample most likely to follow this distribution. In other words, we need a value of the parameter that can maximize the likelihood function. In many applications, for computational convenience, the log-likelihood function

$$\log L(\phi, x) = \sum_{s=1}^n \log f(x_s, \phi)$$

is used to get the estimate. The same result can be obtained by maximizing either of these two functions, since they are monotonically related to each other.

The maximum likelihood estimate of parameter  $\phi$  is then described as

$$\hat{\phi} = \arg \max_{\phi} \log L(\phi, x).$$

Suppose  $\hat{\phi}$  exists, it must satisfy the the following likelihood equation

$$\frac{\partial \log L(\phi, x)}{\partial \phi} = 0.$$

The maximum likelihood estimate can be obtained by solving this likelihood equation.

To summarize, the MLE can be acquired by taking the following three steps.

1. Multiplying the probability function to obtain likelihood function.
2. Calculating the derivative of the log-likelihood function with respect to  $\phi$  and setting it to 0.
3. Solving the likelihood equation for  $\hat{\phi}$ .

In some applications, the likelihood function is quadratic, leading to a linear likelihood equation that is easy to solve. However, in many cases, obtaining a closed-form solution of the likelihood equation could be rather difficult and an iterative method should be applied to obtain the maximum likelihood estimates.

Two typical examples of such iterative algorithms are the Newton-Raphson method and the expectation-maximization(EM) algorithm. In our work, we use the latter to estimate the parameter since it's easy to implement and numerically stable.

### 2.2.3 EM Algorithm

In this section, we first briefly introduce the idea of the EM algorithm and then discuss the formulation of the EM algorithm.

The expectation-maximization (EM) algorithm [16][32][12] is widely used in a variety of situations where the model depends on incomplete data. It provides an intuitive method to estimate the parameters given the existence of “missing data”. In each iteration, the missing values are replaced by estimated values and the parameters

are then estimated which will be used as true values in the next iteration.

To some extent, we can say that the most essential part of the EM algorithm is the incomplete data which makes it possible to build an iterative procedure alternating between an expectation (E) step and a maximization (M) step. In different situations, incomplete data may result from literal missing data or unobservable variables by definition.

In the conventional sense, “missing data” is a part of the complete data that should be observed but was not. Consider a random vector  $\mathbf{y}$  whose joint density function is denoted by  $f(y, \theta)$ . Suppose  $f(y, \theta)$  has a quadratic formula and thus has linear derivative with respect to the unknown parameter  $\theta$ . If data of the whole vector is observed, we can directly write out the log-likelihood function based on the density function  $f(y, \theta)$  and obtain the MLE of  $\theta$  by solving the likelihood equation, which will maximize the log-likelihood function.

Now suppose there exists missing data. For simplicity, we assume one element of the original vector is missing so that the complete data can be denoted as  $\mathbf{y} = (\mathbf{y}_{obs}, y_{mis})$ . This time the likelihood function cannot be directly evaluated or maximized due to the lack of knowledge of  $y_{mis}$ .

As described above, the EM algorithm can be used in this scenario. In the very beginning, we make a guess for the unknown parameter  $\theta$ . So the density function will have determinate formula with the parameter initialized by a constant  $\theta_0$ . Based on this density function and  $\mathbf{y}_{obs}$ , we can “fill in” the missing part and then obtain the likelihood function of complete data. Clearly, in this E-step, we are actually calculating the conditional expectation of the complete-data log-likelihood

$$E_{\theta_0}(\log L(\theta, y) | \mathbf{y}_{obs}). \quad (2.1)$$

This function will have both the current guess  $\theta_0$  which is a constant value, and the variable  $\theta$  which is the unknown parameter to be estimated in the M-step.

In the next step, the derivative of  $E_{\theta_0}(\log L(\theta, y) | \mathbf{y}_{obs})$  is calculated with respect to  $\theta_0$  and is set to 0. By solving this likelihood equation, we can obtain the current guess  $\theta_1$  for the unknown parameter. And this  $\theta_1$  will be used in the E-step of next iteration. We then iterate through this two steps until convergence.

Let’s take the exponential distribution as an example to illustrate the idea shown

above. Suppose

$$f(y, \theta) = \theta e^{-\theta y},$$

where  $\theta$  is the unknown parameter. The complete data vector  $\mathbf{y}$  consists of  $N$  independent realizations denoted by  $y_1, y_2, \dots, y_N$ . When the complete data is available, the log-likelihood function is

$$\begin{aligned} \log L(\theta, \mathbf{y}) &= \log \left( \prod_{i=1}^N f(y_i, \theta) \right) \\ &= \log \left( \theta^N e^{-\theta(y_1 + \dots + y_N)} \right) \\ &= N \log(\theta) - \theta \sum_{i=1}^N y_i. \end{aligned} \quad (2.2)$$

By maximizing Eq.(2.2), an MLE of the parameter  $\theta$  can be directly achieved as

$$\hat{\theta} = \frac{N}{\sum_{i=1}^N y_i}.$$

However, when there exists missing data, the EM algorithm needs to be incorporated in the standard ML method. We denote the observed data as  $\mathbf{y}_{obs} = (y_1, y_2, \dots, y_{N-1})$  and the complete data as  $\mathbf{y} = (\mathbf{y}_{obs}, y_N)$ . By calculating Eq.(2.1), we get the E-step of the first iteration as

$$E_{\theta_0} \left( N \log(\theta) - \theta \sum_{i=1}^N y_i \mid \mathbf{y}_{obs} \right) = N \log(\theta) - \theta \left( \sum_{i=1}^{N-1} y_i + \theta_0 \right). \quad (2.3)$$

In order to maximize Eq.(2.3), the derivative is calculated with respect to  $\theta$  and is set to 0, which leads to

$$\frac{N}{\theta} - \sum_{i=1}^{N-1} y_i + \theta_0 = 0.$$

Thus, the current guess of the parameter  $\theta$  is obtained by solving the above equation

as

$$\theta_1 = \frac{N}{\sum_{i=1}^{N-1} y_i + \theta_0}.$$

This temporary value of parameter  $\theta$  will then be used as a true value in the E-step of the next iteration.

The example shown above involves literal missing data that leads to the incomplete data in the EM algorithm. However, in some other cases, “missing data” is not obvious and formulating an incomplete-data problem becomes a challenge.

In [16] and [34], one example is given where the observed data  $\mathbf{y} = (125, 18, 20, 34)$  followed multinomial distribution with cell probabilities  $(\frac{1}{2} + \frac{1}{4}\theta, \frac{1}{4}(1 - \theta), \frac{1}{4}(1 - \theta), \frac{1}{4}\theta)$ . To formulate this problem into an incomplete-data framework, the first cell was split into two. Then the cell probabilities became  $(\frac{1}{2}, \frac{1}{4}\theta, \frac{1}{4}(1 - \theta), \frac{1}{4}(1 - \theta), \frac{1}{4}\theta)$  and the complete data was correspondingly denoted as  $\mathbf{y} = (y_1, y_2, y_3, y_4, y_5)$ . This time the observed data could be viewed as  $(y_1 + y_2, y_3, y_4, y_5)$  which is a function of the complete data.

The most common example seems to be using EM algorithm for clustering data [2][20] shown in Figure 2.3. In this scenario, the EM algorithm is used to solve the parameter estimation problem of finite mixture model that we stated in Section 2.2.1. For the given data set, a finite Gaussian mixture model is applied to fit the data. Each of the components represents a cluster and has mean and covariance matrix to be estimated. In this case, the essential part of EM algorithm, the unobserved data, is viewed as latent variables that describe from which component a particular data was generated.

We can find standard process for using EM to solve parameter estimation problems of Gaussian mixture model in many machine learning textbooks due to its extensive application in data clustering. In our work, as stated in the preceding section, a variant of the standard finite mixture model was taken to fit the sum of service time data. However, we cannot simply model the service time with Gaussian distribution. Hence the EM algorithm for Gaussian mixture model needs modifications before applied to our parameter estimation problem. The details of formulating our problem into incomplete data framework will be described in Chapter 3.

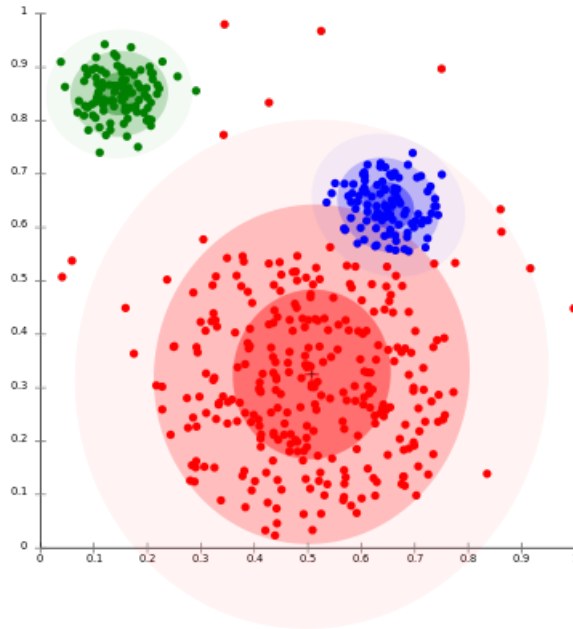


Figure 2.3: EM for data clustering[45]

## 2.3 Previous Work

In a queueing system, the elements supporting the underlying processes are represented by random variables with their corresponding distributions so that the parameter estimation problems in a queueing system are actually statistical inference problems [9].

The first attempt to estimate parameters in a single queue using a maximum likelihood estimation was carried out by Clarke in [13]. He studied an M/M/1 queue and recorded the busy intervals and free intervals along the time axis. Then the maximum likelihood estimates were derived for the parameters involved – arrival rate and service rate. This work was later extended in [1] using different data collection procedures. In [3] and [4], the Bayesian techniques were applied in parameter estimation problems in Markovian queues. Later, both maximum likelihood estimate and the Bayesian method were discussed in [30] for an M/M/1 queueing system, in which data was collected by recording the number of customers appearing at several observation instants. Meanwhile, parameter estimation problems for M/M/1 queues

during working vacations were solved by MLE in [25].

In addition, more research has been carried out to estimate the parameters in more complex queueing system. Goyal and Harris applied the maximum likelihood theory on the parameter estimation problem in a non-Markovian queueing system where the distributions of service times are state dependent [19]. Later, Basawa did a series of work investigating the parameter estimation problems in GI/G/1 queueing systems. Various estimation methods were discussed, including moment estimate [8] and maximum likelihood estimate [5]. In [40] and [39], Thiruvaiyaru and Basawa extended the maximum likelihood estimate for parameter estimation so that it could be applied in Jackson networks. In his later work, Basawa also tried other data collection methods to estimate the parameters in queueing system. In [10], he and Bhat recorded queue lengths and waiting time data while in [7] they recorded waiting time or system time of consecutive customers as well as idle time of servers.

In the next chapter, based on Basawa's previous work, we extend his framework for single M/G/1 system by incorporating finite mixture model, so that it can be used for parameter estimation problems in tandem queueing system.

## Chapter 3

# Parameter Estimation for Tandem Queueing Network

In this chapter, a parameter estimation framework is developed for a two-stage tandem queueing system, with service time of each server following exponential family distributions. Examples will be provided for specific service distributions (exponential service and log-normal service). Then we will show this framework can be extended to a general tandem queueing system with  $n$  servers. A three-stage tandem M/M/1 queueing system is used to illustrate its feasibility.

### 3.1 Data Collection Procedure

A two-stage tandem queueing system was introduced in Section 2.1.2. From the motivation part, we have seen that the parameter we care about in this queueing system is the service rate, which is the reciprocal of service time at each server. Let  $V_1$  and  $V_2$  denote the service times at each server, respectively. Our purpose is to estimate the parameters of these two random variables, whose densities belong to exponential families .

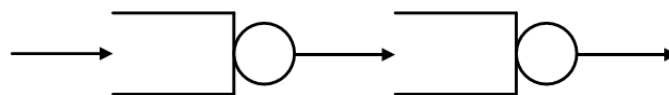


Figure 3.1: Two-stage tandem queueing system

## 3.2 Model of Service time

Let  $V_1^n$  and  $V_2^n$  denote the service times that the  $n$ th packet spend in the two servers of this system, respectively. Suppose the total waiting time  $\{W^n\}$  and total system time  $\{Y^n\}$  are available, we then have

$$Y^n = W^n + V_1^n + V_2^n.$$

Let  $X^n = V_1^n + V_2^n$ , we can get

$$X^n = Y^n - W^n. \quad (3.1)$$

Thus we can construct the i.i.d process  $\{X_n\}$  via Eq.(3.1) based on the  $\{W^n\}$  and  $\{Y^n\}$  processes we have collected.

### 3.2.1 The Exponential Family of Distributions

The probability density functions of  $V_1$  and  $V_2$  are assumed to belong to the exponential family. Let

$$\begin{aligned} f_{V_1}(v_1) &= h_1(v_1) \exp [T_1(v_1) \phi_1 - K_1(\phi_1)] \\ f_{V_2}(v_2) &= h_2(v_2) \exp [T_2(v_2) \phi_2 - K_2(\phi_2)] \end{aligned}$$

where  $\phi_1$  and  $\phi_2$  are the parameters of the distributions,  $T_1(v_1)$  and  $T_2(v_2)$  are sufficient statistics,  $K_1(\phi_1)$  and  $K_2(\phi_2)$  are log normalizers.

For log normalizer  $K(\phi)$ , we have

$$K(\phi) = \log \int h(v) \exp(T(v) \phi)$$

which ensures the integration of density is one.

Another property of exponential family is the moments of the sufficient statistics

Distribution	Density Function	$T(x)$	$K(\phi)$
Exponential	$\phi e^{-\phi x}$	$-x$	$-\log \phi$
Lognormal	$\frac{1}{x\sqrt{2\pi}} \exp(-(\log x - \phi)^2/2)$	$\log x$	$\phi^2/2$
Beta( $\phi, 1$ )	$\phi x^{\phi-1}$	$\log x$	$-\log \phi$
Gamma	$\frac{\phi(\phi x)^{k-1} e^{-\phi x}}{\Gamma(k)}$ ( $k$ known)	$-x$	$-k \log \phi$

Table 3.1: Examples of exponential family distributions

can be obtained by the derivatives of log normalizer, that is

$$\begin{aligned} \frac{d}{d\phi} K(\phi) &= E(T(v)), \\ \frac{d^2}{d\phi^2} K(\phi) &= \text{Var}(T(v)). \end{aligned}$$

In Table 3.1, we provide some examples of non-negative exponential family [6].

### 3.2.2 Density of the Sum

Let  $X = V_1 + V_2$ . Then  $X$  represents the total service time a packet experienced in the tandem queueing system.

Given the density functions of  $V_1$  and  $V_2$ , we know that the density function of the sum is the convolution of the two components. The density function of  $X$  is then given by

$$f_X(x) = \int_0^\infty f_{V_1}(v_1) f_{V_2}(x - v_1) dv_1, \quad x \geq v_1 > 0. \quad (3.2)$$

Since  $V_1 = X - V_2$ , the density function of  $X$  can also be written as

$$f_X(x) = \int_0^\infty f_{V_1}(x - v_2) f_{V_2}(v_2) dv_2, \quad x \geq v_2 > 0. \quad (3.3)$$

### 3.3 The Likelihood Function

We use Eq.(3.2) as the density function of  $X$ . Based on the sample data  $(x_1, \dots, x_N)$ , we can get the likelihood function  $L_N(v_1, v_2) = \prod_{i=1}^N f_X(x_i)$ . The ML estimation of  $\phi_1$  and  $\phi_2$  can be obtained by solving the following equations:

$$\frac{\partial \log L_N}{\partial \phi_1} = \sum_{i=1}^N \frac{\frac{\partial f_X(x_i)}{\partial \phi_1}}{f_X(x_i)} = 0 \quad \text{and} \quad \frac{\partial \log L_N}{\partial \phi_2} = \sum_{i=1}^N \frac{\frac{\partial f_X(x_i)}{\partial \phi_2}}{f_X(x_i)} = 0. \quad (3.4)$$

Similar to Eq.(3.7)-Eq.(3.10) in Basawa's paper [6], we have

$$\begin{aligned} \frac{\partial f_X(x)}{\partial \phi_1} &= \int_0^\infty \left( \frac{\partial f_{V_1}(v_1)}{\partial \phi_1} \right) f_{V_2}(x-v_1) dv_1 \\ &= \int_0^\infty [T_1(v_1) - K_1'(\phi_1)] f_{V_1}(v_1) f_{V_2}(x-v_1) dv_1, \end{aligned} \quad (3.5)$$

$$\begin{aligned} \frac{\partial f_X(x)}{\partial \phi_2} &= \int_0^\infty \left( \frac{\partial f_{V_2}(x-v_1)}{\partial \phi_2} \right) f_{V_1}(v_1) dv_1 \\ &= \int_0^\infty [T_2(x-v_1) - K_2'(\phi_2)] f_{V_1}(v_1) f_{V_2}(x-v_1) dv_1. \end{aligned}$$

Substituting Eq.(3.5) into Eq.(3.4), we solve the equation and get

$$K_1'(\phi_1) = \frac{1}{N} \sum_{i=1}^N \frac{\int_0^\infty T_1(v_1) f_{V_1}(v_1) f_{V_2}(x_i - v_1) dv_1}{\int_0^\infty f_{V_1}(v_1) f_{V_2}(x_i - v_1) dv_1}, \quad (3.6)$$

$$K_2'(\phi_2) = \frac{1}{N} \sum_{i=1}^N \frac{\int_0^\infty T_2(x_i - v_1) f_{V_1}(v_1) f_{V_2}(x_i - v_1) dv_1}{\int_0^\infty f_{V_1}(v_1) f_{V_2}(x_i - v_1) dv_1}. \quad (3.7)$$

Since Eq.(3.3) is equivalent to Eq.(3.2), then Eq.(3.5) can be equivalently written as

$$\begin{aligned}
\frac{\partial f_X(x)}{\partial \phi_1} &= \int_0^\infty \left( \frac{\partial f_{V_1}(x-v_2)}{\partial \phi_1} \right) f_{V_2}(v_2) dv_2 \\
&= \int_0^\infty [T_1(x-v_2) - K_1'(\phi_1)] f_{V_1}(x-v_2) f_{V_2}(v_2) dv_2, \\
\frac{\partial f_X(x)}{\partial \phi_2} &= \int_0^\infty \left( \frac{\partial f_{V_2}(v_2)}{\partial \phi_2} \right) f_{V_1}(x-v_2) dv_2 \\
&= \int_0^\infty [T_2(v_2) - K_2'(\phi_2)] f_{V_1}(x-v_2) f_{V_2}(v_2) dv_2.
\end{aligned} \tag{3.8}$$

Therefore, Eq.(3.6) and Eq.(3.7) are respectively equivalent to

$$K_1'(\phi_1) = \frac{1}{N} \sum_{i=1}^N \frac{\int_0^\infty T_1(x-v_2) f_{V_1}(x_i-v_2) f_{V_2}(v_2) dv_2}{\int_0^\infty f_{V_1}(x_i-v_2) f_{V_2}(v_2) dv_2}, \tag{3.9}$$

$$K_2'(\phi_2) = \frac{1}{N} \sum_{i=1}^N \frac{\int_0^\infty T_2(v_2) f_{V_1}(x_i-v_2) f_{V_2}(v_2) dv_2}{\int_0^\infty f_{V_1}(x_i-v_2) f_{V_2}(v_2) dv_2}. \tag{3.10}$$

From here on, we just use the equations derived from Eq.(3.2), which takes  $V_1$  as the independent variable.

For a certain distribution, the exponential distribution for example, the actual formula of the log normalizer  $K(\phi)$  could be found in Table 3.1. Then the parameters of the distribution could be calculated directly from the value of  $K(\phi)$  or  $K'(\phi)$  equivalently.

### 3.4 EM Algorithm

If the complete sample of  $V_1$  and  $V_2$  is available, as  $\{(v_1^i, v_2^i), i = 1, \dots, N\}$ , based on which we will have the complete likelihood

$$L^c(\phi_1, \phi_2) = \left( \prod_{i=1}^N f_{V_1}(v_1^i) \right) \left( \prod_{i=1}^N f_{V_2}(v_2^i) \right). \tag{3.11}$$

From the properties of the exponential families, the likelihood equations corresponding to the complete likelihood can be directly obtained as

$$\begin{aligned} K_1'(\phi_1) &= E(T_1(V_1)) = \frac{1}{N} \sum_{i=1}^N T_1(v_1^i), \\ K_2'(\phi_2) &= E(T_2(V_2)) = \frac{1}{N} \sum_{i=1}^N T_2(v_2^i). \end{aligned} \quad (3.12)$$

It's quite straightforward to solve this problem given the complete data. Nevertheless, in our problem, the only observed sample is  $(x_i, i = 1, \dots, N)$  where  $x_i = v_1^i + v_2^i$ . The observed data is a function of the two variables and we consider it as a scenario that involves "missing data". Therefore we need to use the conditional expectations of  $T_1(V_1)$  and  $T_2(V_2)$  with respect to the sample  $x_i$  in Eq.(3.12). The likelihood equations are then seen to be

$$\begin{aligned} K_1'(\phi_1) &= \frac{1}{N} \sum_{i=1}^N E(T_1(V_1) | X = x_i), \\ K_2'(\phi_2) &= \frac{1}{N} \sum_{i=1}^N E(T_2(V_2) | X = x_i). \end{aligned} \quad (3.13)$$

We now proceed to calculate the conditional expectations in Eq.(3.13)

$$\begin{aligned} E(T_1(V_1) | X = x) &= \int_0^\infty T_1(v_1) f_{V_1|X}(v_1|x) dv_1 \\ &= \int_0^\infty -v_1 f_{V_1|X}(v_1|x) dv_1 \\ &= - \int_0^\infty v_1 \frac{f_{V_1,X}(v_1, x)}{f_X(x)} dv_1. \end{aligned} \quad (3.14)$$

Here we suppose  $V_1$  and  $V_2$  are independent and both follow exponential distribution, then

$$V_1, V_2 \sim f_{V_1, V_2}(v_1, v_2) = f_{V_1}(v_1) f_{V_2}(v_2) = \phi_1 \phi_2 e^{-\phi_1 v_1 - \phi_2 v_2}.$$

Let  $Y_1 = g_1(V_1, V_2) = V_1$  and  $Y_2 = g_2(V_1, V_2) = V_1 + V_2$ . Since the mapping from

$(V_1, V_2)$  to  $(Y_1, Y_2)$  is one-to-one, the system  $\begin{cases} y_1 = g_1(v_1, v_2) \\ y_2 = g_2(v_1, v_2) \end{cases}$  can be inverted to get  $\begin{cases} v_1 = h_1(y_1, y_2) = y_1 \\ v_2 = h_2(y_1, y_2) = y_2 - y_1 \end{cases}$ .

We define the Jacobian matrix  $J_h = \begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_1}{\partial y_2} \\ \frac{\partial h_2}{\partial y_1} & \frac{\partial h_2}{\partial y_2} \end{bmatrix}$ . The absolute value of the determinant of  $J_h$  is  $|J_h| = \begin{vmatrix} 1 & 0 \\ -1 & 1 \end{vmatrix} = 1$ .

Then we can get the joint density of  $V_1$  and  $X$  as

$$\begin{aligned} f_{V_1, X}(v_1, x) &= f_{Y_1, Y_2}(y_1, y_2) \\ &= f_{V_1, V_2}(h_1(y_1, y_2), h_2(y_1, y_2)) * |J_h| \\ &= \phi_1 \phi_2 e^{-\phi_1 v_1 - \phi_2(x - v_1)}. \end{aligned}$$

Back to Eq.(3.14), we have

$$\begin{aligned} E(T_1(V_1) | X = x) &= -\frac{1}{f_X(x)} \int_0^\infty v_1 \phi_1 \phi_2 e^{-\phi_1 v_1 - \phi_2(x - v_1)} dv_1 \\ &= -\frac{\int_0^\infty v_1 \phi_1 \phi_2 e^{-\phi_1 v_1 - \phi_2(x - v_1)} dv_1}{\int_0^\infty f_{V_1}(v_1) f_{V_2}(x - v_1) dv_1} \\ &= -\frac{\phi_1 \phi_2 e^{-\phi_2 x} \int_0^x v_1 e^{(-\phi_1 + \phi_2)v_1} dv_1}{\phi_1 \phi_2 e^{-\phi_2 x} \int_0^x e^{(-\phi_1 + \phi_2)v_1} dv_1} \\ &= -\frac{\int_0^x v_1 e^{(-\phi_1 + \phi_2)v_1} dv_1}{\int_0^x e^{(-\phi_1 + \phi_2)v_1} dv_1} \\ &= \frac{1}{\phi_2 - \phi_1} - \frac{x}{1 - e^{(\phi_1 - \phi_2)x}}. \end{aligned} \tag{3.15}$$

Similarly, we calculate the conditional expectation of  $T_2(V_2)$  as

$$\begin{aligned}
E(T_2(V_2) | X = x) &= \int_0^\infty T_2(x - v_1) f_{V_1|X}(v_1|x) dv_1 \\
&= \frac{1}{f_X(x)} \int_0^\infty (v_1 - x) \phi_1 \phi_2 e^{-\phi_1 v_1 - \phi_2(x-v_1)} dv_1 \\
&= \frac{\phi_1 \phi_2 e^{-\phi_2 x} \int_0^\infty (v_1 - x) e^{(-\phi_1 + \phi_2)v_1} dv_1}{\phi_1 \phi_2 e^{-\phi_2 x} \int_0^x e^{(-\phi_1 + \phi_2)v_1} dv_1} \\
&= \frac{\int_0^x (v_1 - x) e^{(-\phi_1 + \phi_2)v_1} dv_1}{\int_0^x e^{(-\phi_1 + \phi_2)v_1} dv_1} \\
&= \frac{x}{e^{(\phi_2 - \phi_1)x} - 1} - \frac{1}{\phi_2 - \phi_1}. \tag{3.16}
\end{aligned}$$

Based on this, our EM algorithm is then described as follows.

**Step 1. (E step)** Let  $(\phi_1^0, \phi_2^0)$  be the initial value of  $(\phi_1, \phi_2)$ . The conditional expectations of  $T_1(V_1)$  and  $T_2(V_2)$  are then calculated using  $\phi_1^0, \phi_2^0$  and the sample data  $x$  based on Eq.(3.15) and Eq.(3.16).

**Step 2. (M step)** From the likelihood equation Eq.(3.13) and the formula of  $K(\phi)$  according to the particular distribution, we can directly obtain  $(\phi_1^1, \phi_2^1)$ .

**Step 3.** Use  $(\phi_1^1, \phi_2^1)$  obtained in M step as the starting value and repeat the E step and M step until convergence.

### 3.5 Modified EM Algorithm

From the previous section we see that in the EM algorithm, conditional expectation of  $T_1(V_1)$  and  $T_2(V_2)$  need to be calculated based on the density functions of the variables. However, in some cases, due to the complexity of the density function, the calculation of conditional expectation could be rather difficult and time-consuming. Thus, we use a modified EM algorithm where the conditional expectation is replaced by their empirical approximation.

Based on Eq.(3.2), we have

$$\begin{aligned}
E(T_1(V_1) | X = x) &= \frac{\int_0^\infty T_1(v_1) f_{V_1}(v_1) f_{V_2}(x - v_1) dv_1}{\int_0^\infty f_{V_1}(v_1) f_{V_2}(x - v_1) dv_1}, \\
E(T_2(V_2) | X = x) &= \frac{\int_0^\infty T_2(x - v_1) f_{V_1}(v_1) f_{V_2}(x - v_1) dv_1}{\int_0^\infty f_{V_1}(v_1) f_{V_2}(x - v_1) dv_1}. \tag{3.17}
\end{aligned}$$

Regarding the numerators and denominators as the expectations of functions of  $v_1$ , we then have

$$\begin{aligned} E(T_1(V_1) | X = x) &= \frac{E(T_1(v_1) f_{V_2}(x - v_1))}{E(f_{V_2}(x - v_1))}, \\ E(T_2(V_2) | X = x) &= \frac{E(T_2(x - v_1) f_{V_2}(x - v_1))}{E(f_{V_2}(x - v_1))}. \end{aligned} \quad (3.18)$$

Here instead of calculating the theoretical expectation, we use Monte Carlo method to calculate the mean value. The details of this modified EM algorithm are described below.

**Step 1.** Let  $(\phi_1^0, \phi_2^0)$  be the initial value of  $(\phi_1, \phi_2)$ . Simulate i.i.d. observations  $(v_{1j}^0, j = 1, \dots, m)$  from  $f_{V_1}(v_1; \phi_1^0)$  that satisfy  $v_{1j}^0 \leq x_i$ . Compute the numerical mean value for the expectations as follow.

$$\begin{aligned} \hat{E}(T_1(V_1) | X = x_i) &= \frac{\sum_{j=1}^m T_1(v_{1j}^0) f_{V_2}(x_i - v_{1j}^0; \phi_2^0)}{\sum_{j=1}^m f_{V_2}(x_i - v_{1j}^0; \phi_2^0)}, \\ \hat{E}(T_2(V_2) | X = x_i) &= \frac{\sum_{j=1}^m T_2(x_i - v_{1j}^0) f_{V_2}(x_i - v_{1j}^0; \phi_2^0)}{\sum_{j=1}^m f_{V_2}(x_i - v_{1j}^0; \phi_2^0)} \end{aligned} \quad (3.19)$$

**Step 2.** Calculate  $K_1'(\phi_1)$  and  $K_2'(\phi_2)$  by solving  $K_1'(\phi_1) = \frac{1}{N} \sum_{i=1}^N \hat{E}(T_1(V_1) | X = x_i)$  and  $K_2'(\phi_2) = \frac{1}{N} \sum_{i=1}^N \hat{E}(T_2(V_2) | X = x_i)$ . Then we can find  $(\phi_1^1, \phi_2^1)$  according to the form of  $K_1'(\phi_1)$  and  $K_2'(\phi_2)$ .

**Step 3.** Use  $(\phi_1^1, \phi_2^1)$  obtained in Step 2 as the starting value to calculate the probability distribution function of  $V_1$  and  $V_2$ . Repeat the process until the estimations of  $\phi_1$  and  $\phi_2$  converge.

If we take  $V_2$  as the independent variable, just replace Eq.(3.19) in the EM algorithm by

$$\begin{aligned}
\hat{E} [T_2 (V_2) | X = x_i] &= \frac{\sum_{j=1}^m T_2 (v_{2j}^0) f_{V_1} (x_i - v_{2j}^0; \phi_1^0)}{\sum_{j=1}^m f_{V_1} (x_i - v_{2j}^0 (j); \phi_1^0)}, \\
\hat{E} [T_1 (V_1) | X = x_i] &= \frac{\sum_{j=1}^m T_1 (x_i - v_{2j}^0) f_{V_1} (x_i - v_{2j}^0 (j); \phi_1^0)}{\sum_{j=1}^m f_{V_1} (x_i - v_{2j}^0 (j); \phi_1^0)}
\end{aligned} \tag{3.20}$$

## 3.6 Illustrative Examples

### 3.6.1 Parameter Estimation of Exponential Service Times

In this section, we consider a queueing system in which both of the two servers have exponential service times. The density functions respectively are

$$\begin{aligned}
f_{V_1} (v_1) &= \phi_1 e^{-\phi_1 v_1}, \\
f_{V_2} (v_2) &= \phi_2 e^{-\phi_2 v_2}.
\end{aligned}$$

Estimate the parameters of  $V_1$  and  $V_2$  through a normal EM algorithm has been discussed in Section 3.4. We now proceed to show how to formulate the modified EM algorithm.

For exponential distributions, we can get the sufficient statistics and log normalizer from Table 3.1 as

$$\begin{aligned}
T (v) &= -v, \\
K (\phi) &= -\log \phi.
\end{aligned} \tag{3.21}$$

Substituting Eq.(3.21) in Eq.(3.19), we have

$$\begin{aligned}
\hat{E} (T_1 (V_1) | X = x_i) &= \frac{\sum_{j=1}^m (-v_{1j}) \phi_2 e^{-\phi_2 (x_i - v_{1j})}}{\sum_{j=1}^m \phi_2 e^{-\phi_2 (x_i - v_{1j})}}, \\
\hat{E} (T_2 (V_2) | X = x_i) &= \frac{\sum_{j=1}^m - (x_i - v_{1j}) \phi_2 e^{-\phi_2 (x_i - v_{1j})}}{\sum_{j=1}^m \phi_2 e^{-\phi_2 (x_i - v_{1j})}}.
\end{aligned} \tag{3.22}$$

In the first iteration, let  $\phi_1^0$  and  $\phi_2^0$  be the initial values of the parameters. Then we can get the density functions as

$$\begin{aligned} f_{V_1}(v_1) &= \phi_1^0 e^{-\phi_1^0 v_1}, \\ f_{V_2}(v_2) &= \phi_2^0 e^{-\phi_2^0 v_2}. \end{aligned} \tag{3.23}$$

Based on Eq.(3.23), we simulate a set of observations  $(v_{1j}^0, j = 1, \dots, m)$  for each  $x_i$  that satisfy  $v_{1j}^0 \leq x_i$  and substitute them in Eq.(3.22). The values of  $K_1'(\phi_1)$  and  $K_2'(\phi_2)$  are then calculated by solving Eq.(3.13).

From Eq.(3.21), we can get

$$K'(\phi) = -\frac{1}{\phi}. \tag{3.24}$$

Therefore, the current estimation of the parameters can be obtained by solving

$$\begin{aligned} \phi_1^1 &= -\frac{1}{K'_1(\phi_1)}, \\ \phi_2^1 &= -\frac{1}{K'_2(\phi_1)}. \end{aligned} \tag{3.25}$$

In the second iteration, let  $\phi_1^1$  and  $\phi_2^1$  be the initial values and repeat the process described in the first iteration, we will get  $\phi_1^2$  and  $\phi_2^2$ . After several iterations the result will converge<sup>1</sup> and we will obtain the estimates of  $\phi_1$  and  $\phi_2$ .

### 3.6.2 Interchangeability of the Service Time

Note that the scheme described in the previous section itself cannot finally determine the values of the parameters of  $V_1$  and  $V_2$ . The reason is that within the whole scheme,  $V_1$  and  $V_2$  are interchangeable. We get two estimates from the EM algorithm but we don't know whether the larger one represents  $V_1$  or  $V_2$ . Only with some prior knowledge like  $V_1 > V_2$  for example, can we make a final conclusion.

This problem can also be seen in the moment estimation. Suppose  $a$  and  $b$  are

---

<sup>1</sup>The convergence of the EM algorithm has been proved in [16]

known values and

$$E(X) = E(V_1 + V_2) = E(V_1) + E(V_2) = \frac{1}{\phi_1} + \frac{1}{\phi_2} = a,$$

$$D(X) = D(V_1 + V_2) = D(V_1) + D(V_2) = \frac{1}{\phi_1^2} + \frac{1}{\phi_2^2} = b.$$

We get two possible solutions without any prior knowledge:

$$\phi_1 = \frac{2}{a + \sqrt{2b - a^2}}, \quad \phi_2 = \frac{2}{a - \sqrt{2b - a^2}}$$

or

$$\phi_1 = \frac{2}{a - \sqrt{2b - a^2}}, \quad \phi_2 = \frac{2}{a + \sqrt{2b - a^2}}$$

### 3.6.3 Different Types of Service Times

As discussed in Section 3.2, the service time is modeled by exponential family of distributions, thus our mixture model of total service time may contain components following different types of distributions.

Here is another example to illustrate our modified EM algorithm where service times of the two servers follow exponential distribution and lognormal distribution, respectively. We then have the density functions as

$$f_{V_1}(v_1) = \frac{1}{v_1 \sqrt{2\pi}} e^{-(\log v_1 - \phi_1)^2 / 2},$$

$$f_{V_2}(v_2) = \phi_2 e^{-\phi_2 v_2}.$$
(3.26)

For lognormal distributions, we get

$$T(v) = \log v,$$

$$K(\phi) = \phi^2 / 2.$$
(3.27)

Applying Eq.(3.21) and Eq.(3.27) in Eq.(3.19), we have

$$\begin{aligned}\hat{E}(T_1(V_1)|X=x_i) &= \frac{\sum_{j=1}^m \log v_{1j} \phi_2 e^{-\phi_2(x_i-v_{1j})}}{\sum_{j=1}^m \phi_2 e^{-\phi_2(x_i-v_{1j})}}, \\ \hat{E}(T_2(V_2)|X=x_i) &= \frac{\sum_{j=1}^m \log(x_i-v_{1j}) \phi_2 e^{-\phi_2(x_i-v_{1j})}}{\sum_{j=1}^m \phi_2 e^{-\phi_2(x_i-v_{1j})}}.\end{aligned}\tag{3.28}$$

In the first iteration, we use the initial values of the parameters  $(\phi_1^0, \phi_2^0)$  in Eq.(3.26) and generate the observations  $(v_{1j}^0, j = 1, \dots, m)$  just as what we did in the first example. For lognormal distributions, we can get  $K'(\phi) = \phi$  from Eq.(3.27). In the end of this iteration, based on the values of  $K'_1(\phi_1)$  and  $K'_2(\phi_2)$ , we can calculate the parameters by solving

$$\begin{aligned}\phi_1^1 &= -\frac{1}{K'_1(\phi_1)}, \\ \phi_2^1 &= K'_2(\phi_2).\end{aligned}\tag{3.29}$$

The current estimation of the parameters is then used in the next iteration as the initial value. The whole process repeats until convergence.

### 3.7 Tandem Queueing System with n Nodes

In this section of this chapter, we consider the parameter estimation problem for a tandem queueing system consisting of n nodes in series, shown in Figure 3.2.

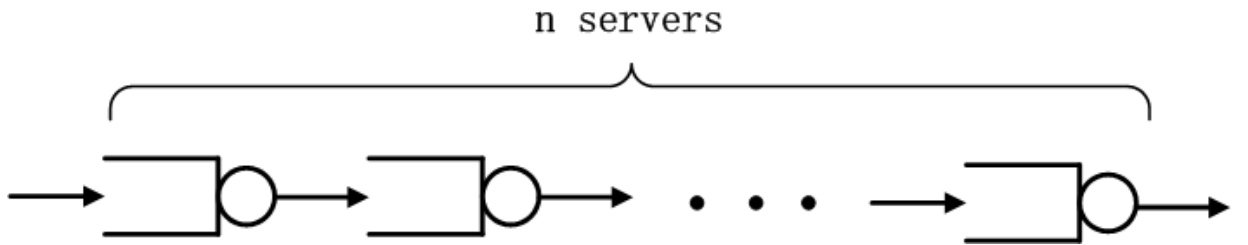


Figure 3.2: Tandem queueing system consisting of n servers

### 3.7.1 PDF of the Sum of $n$ Independent Random Variables

We have discussed in our previous section how to derive the probability distribution function of the sum of two independent variables. Here we use a recursive method to derive the distribution of the sum of  $n$  mutually independent random variables.

Let  $X = V_1 + V_2 + \dots + V_n$ . The probability distribution function of  $X$  can be recursively derived using the results of the probability density function of the sum of two independent variables [37]. The details are described as follows.

**Step 1.**

We first set

$$Y_2 = V_1 + V_2.$$

The distribution of  $Y_2$  can be easily obtained by calculating the convolution of the PDFs of  $V_1$  and  $V_2$ .

**Step 2.**

Then we set

$$Y_3 = Y_2 + V_3,$$

and calculate the distribution of  $Y_3$ .

**Step 3.**

Repeat the previous process and finally we will get the distribution of  $X$  based on

$$X = Y_n = Y_{n-1} + V_n.$$

### 3.7.2 Tandem Queueing System with Three Servers

In the following, we use a tandem queueing system consisting of three servers as an example to illustrate how parameters are estimated using our general framework for  $n$  servers.

As described in Section 3.2, data of total service time  $X$  is acquired by the difference of total system time and total waiting time. This variable  $X$  is considered as the sum of  $V_1$ ,  $V_2$  and  $V_3$ , which represent the service times of each server, respectively.

We first calculate the density function of  $X$ .

Let  $Y = V_2 + V_3$ , the density function of  $Y$  is then written as

$$f_Y(y) = \int_0^y f_{V_2}(v_2) f_{V_3}(y - v_2) dv_2. \quad (3.30)$$

Since  $X = V_1 + Y$ , we can get the density function of  $X$  as

$$\begin{aligned} f_X(x) &= \int_0^x f_{V_1}(v_1) f_Y(x - v_1) dv_1 \\ &= \int_0^x f_{V_1}(v_1) \int_0^{x-v_1} f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1. \end{aligned} \quad (3.31)$$

Based on Eq.(3.31), we get get

$$\begin{aligned} E(T_1(v_1) | X = x) &= \frac{\int_0^x T_1(v_1) f_{V_1}(v_1) \int_0^{x-v_1} f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1}{\int_0^\infty f_{V_1}(v_1) \int_0^{x-v_1} f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1} \\ &= \frac{\int_0^x \int_0^{x-v_1} T_1(v_1) f_{V_1}(v_1) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1}{\int_0^\infty \int_0^{x-v_1} f_{V_1}(v_1) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1} \end{aligned}$$

$$\begin{aligned} E(T_2(v_2) | X = x) &= \frac{\int_0^x f_{V_1}(v_1) \int_0^{x-v_1} T_2(v_2) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1}{\int_0^\infty f_{V_1}(v_1) \int_0^{x-v_1} f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1} \\ &= \frac{\int_0^x \int_0^{x-v_1} T_2(v_2) f_{V_1}(v_1) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1}{\int_0^\infty \int_0^{x-v_1} f_{V_1}(v_1) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1} \end{aligned}$$

$$\begin{aligned} E(T_3(v_3) | X = x) &= \frac{\int_0^x f_{V_1}(v_1) \int_0^{x-v_1} T_3(v_3) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1}{\int_0^\infty f_{V_1}(v_1) \int_0^{x-v_1} f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1} \\ &= \frac{\int_0^x \int_0^{x-v_1} T_3(v_3) f_{V_1}(v_1) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1}{\int_0^\infty \int_0^{x-v_1} f_{V_1}(v_1) f_{V_2}(v_2) f_{V_3}(x - v_1 - v_2) dv_2 dv_1} \end{aligned}$$

Based on the modified EM algorithm, we use the Monte Carlo method to calculate the integration appeared in Eq.(3.32). The details are described as follows:

**Step 1.** Let  $(\phi_1^0, \phi_2^0, \phi_3^0)$  be the initial value of  $(\phi_1, \phi_2, \phi_3)$ . Simulate i.i.d. observations  $(v_{1j}^0, j = 1, \dots, m)$  from  $f_{V_1}(v_1; \phi_1^0)$  and  $(v_{2j}^0, j = 1, \dots, m)$  from  $f_{V_2}(v_2; \phi_2^0)$  that satisfy  $v_{1j}^0 + v_{2j}^0 \leq x_i$ . Then  $(v_{3j}^0, j = 1, \dots, m)$  can be represented as  $x_i - v_{1j}^0 - v_{2j}^0$ . The

conditional expectations of  $T_1$ ,  $T_2$  and  $T_3$  are calculated using Monte Carlo method as

$$\begin{aligned}\hat{E}(T_1(V_1)|X=x_i) &= \frac{\sum_{j=1}^m T_1(v_{1j}^0) f_{V_3}(x_i - v_{1j}^0 - v_{2j}^0; \phi_3^0)}{\sum_{j=1}^m f_{V_3}(x_i - v_{1j}^0 - v_{2j}^0; \phi_3^0)}, \\ \hat{E}(T_2(V_2)|X=x_i) &= \frac{\sum_{j=1}^m T_2(v_{2j}^0) f_{V_3}(x_i - v_{1j}^0 - v_{2j}^0; \phi_3^0)}{\sum_{j=1}^m f_{V_3}(x_i - v_{1j}^0 - v_{2j}^0; \phi_3^0)}, \\ \hat{E}(T_3(V_3)|X=x_i) &= \frac{\sum_{j=1}^m T_3(x_i - v_{1j}^0 - v_{2j}^0) f_{V_3}(x_i - v_{1j}^0 - v_{2j}^0; \phi_3^0)}{\sum_{j=1}^m f_{V_3}(x_i - v_{1j}^0 - v_{2j}^0; \phi_3^0)}.\end{aligned}$$

Calculate  $K_1'(\phi_1)$ ,  $K_2'(\phi_2)$  and  $K_3'(\phi_3)$  by solving

$$\begin{aligned}K_1'(\phi_1) &= \frac{1}{N} \sum_{i=1}^N \hat{E}(T_1(V_1)|X=x_i), \\ K_2'(\phi_2) &= \frac{1}{N} \sum_{i=1}^N \hat{E}(T_2(V_2)|X=x_i), \\ K_3'(\phi_3) &= \frac{1}{N} \sum_{i=1}^N \hat{E}(T_3(V_3)|X=x_i).\end{aligned}\tag{3.32}$$

Then we can obtain the current estimation of  $(\phi_1^1, \phi_2^1, \phi_3^1)$  according to Eq.(3.32).

**Step 2.** Use the value  $(\phi_1^1, \phi_2^1, \phi_3^1)$  obtained in Step 1 as the starting value to calculate the probability distribution function of  $V_1$ ,  $V_2$  and  $V_3$ . Repeat the process until the estimations of  $\phi_1, \phi_2$  and  $\phi_3$  converge.

To further illustrate this framework, let's take exponential distribution as an example. Suppose  $V_1, V_2, V_3$  are independent random variables and follow exponential distributions with parameter  $\phi_1, \phi_2$  and  $\phi_3$ , respectively. According to our modified EM algorithm, we simulate  $(v_{1j}^0, j = 1, \dots, m)$  and  $(v_{2j}^0, j = 1, \dots, m)$  based on

$$\begin{aligned}f_{V_1}(v_1) &= \phi_1^0 e^{-\phi_1^0 v_1}, \\ f_{V_2}(v_2) &= \phi_2^0 e^{-\phi_2^0 v_2},\end{aligned}$$

where  $(\phi_1^0, \phi_2^0, \phi_3^0)$  is the initial value of the parameters to be estimated.

Then the conditional expectations are computed by

$$\begin{aligned}\hat{E}(T_1(V_1)|X=x_i) &= \frac{\sum_{j=1}^m (-v_{1j}^0) \phi_3^0 e^{-\phi_3^0(x_i-v_{1j}^0-v_{2j}^0)}}{\sum_{j=1}^m \phi_3^0 e^{-\phi_3^0(x_i-v_{1j}^0-v_{2j}^0)}}, \\ \hat{E}(T_2(V_2)|X=x_i) &= \frac{\sum_{j=1}^m (-v_{2j}^0) \phi_3^0 e^{-\phi_3^0(x_i-v_{1j}^0-v_{2j}^0)}}{\sum_{j=1}^m \phi_3^0 e^{-\phi_3^0(x_i-v_{1j}^0-v_{2j}^0)}}, \\ \hat{E}(T_3(V_3)|X=x_i) &= \frac{\sum_{j=1}^m -(x_i-v_{1j}^0-v_{2j}^0) \phi_3^0 e^{-\phi_3^0(x_i-v_{1j}^0-v_{2j}^0)}}{\sum_{j=1}^m \phi_3^0 e^{-\phi_3^0(x_i-v_{1j}^0-v_{2j}^0)}}.\end{aligned}$$

Based on Eq.(3.32), we can get the values of  $(K'_i(\phi), i = 1, 2, 3)$  and this result can be further used to obtain the estimate of  $(\phi_1^1, \phi_2^1, \phi_3^1)$  since

$$\begin{aligned}\phi_1^1 &= -\frac{1}{K'_1(\phi_1)}, \\ \phi_2^1 &= -\frac{1}{K'_2(\phi_1)}, \\ \phi_3^1 &= -\frac{1}{K'_3(\phi_1)}.\end{aligned}$$

In the next iteration, the current estimate  $(\phi_1^1, \phi_2^1, \phi_3^1)$  is used as the initial value to simulate  $(v_{1j}^1, j = 1, \dots, m)$  and  $(v_{2j}^1, j = 1, \dots, m)$ , which makes it possible to build the iterative procedure<sup>2</sup>. The final estimate of parameter will be obtained when the procedure converges.

### 3.7.3 Interchangeability of Service Time in Tandem Queueing Network

The interchangeability of service time we discussed in Section 3.6.2 is also true for a tandem queueing network with more than two servers.<sup>3</sup>

In our work, a tandem queueing network is comprised of several nodes whose service times follow exponential family distributions. In order to estimate the service rate of each server, basically we are going to estimate the parameters of several random variables based on their sum data. Since the service times at each server

<sup>2</sup>We don't need to simulate  $(v_{3j}^1, j = 1, \dots, m)$  since it can be represented as  $x_i - v_{1j}^1 - v_{2j}^1$ .

<sup>3</sup>The interchangeability of service time doesn't imply that servers in the queueing network are interchangeable.

are independent and share equal weight when constructing the sum data, they are interchangeable, thus we need extra knowledge to make a final conclusion.

### 3.8 Comparison with Basawa's Work

In this section, we compare our work with the previous research [7] by Basawa. In [7], the author investigated the parameter estimation problem based on a sample of the differences of two positive random variables  $U$  and  $V$ , where  $U$  and  $V$  follow exponential family distributions. In other words, given the data of  $V - U$ , the parameters of  $U$  and  $V$  could be estimated through a modified EM algorithm involving Monte Carlo simulation. Instead of working on the differences of two variables, our work needs to solve parameter estimation problem based on a sample of the sum of two random variables  $V_1$  and  $V_2$  whose densities belong to exponential family. Inspired by Basawa's work, we use a very similar way to formulate the ML estimation. With modifications, the EM algorithm stated in [7] can be applied to solve the likelihood equations in our work.

One of the differences between these two works is the scalability. The work done in [7] is applied to a single queueing system but not queueing network with more than one node. In contrast, as described in the preceding section, our framework can solve the parameter estimation problem based on sum data of more than two random variables, and thus can be applied to a tandem queueing network with several nodes.

Another difference is that the work in [7] estimates both arrival rate and service rate of a queue while our work only focuses on the service rate estimation. Besides, the method proposed in [7] collects system time data or waiting time data together with the servers idle time to obtain the sample data of the differences. In our work, we collect both system time and waiting time data but no need for idle time data.

### 3.9 Further Discussion

Same as Basawa's work [7], we assume that waiting times are available. This assumption is generally true and is justified that the waiting time in a queue is easy to render a direct measurement [14].

Clearly, this assumption may be too strong, as it requires the involvement of intermediate nodes. In scenarios where only end-to-end delay is available, our method can be still applied to obtain the system time at each individual node, if we re-define

the service time at each node to also include the waiting time at the node. Note that in the traditional meaning, system time of a node is equal to the waiting time in the queue plus the service time of the node. In other words, the re-defined “service time” is the same as the system time in the traditional meaning.

Nevertheless, this re-definition of service time poses a constraint of our method. The distribution of the re-defined service time may not belong to exponential family anymore, and only few limited queueing systems have such a property, e.g., M/M/1 system. Indeed, this limitation roots from the widely-known difficulty in the network of queue: the input and the output of a node may not follow the same distribution, making it extremely hard to adopt a unified model to analyze the system time of all nodes. We cannot avoid this problem, and we conclude that our method can only be applied in scenarios where total waiting time is available or in scenarios where the distribution of system times of each node belong to the exponential family.

# Chapter 4

## Experiments

In this chapter, we carry out simulation experiments and analyze the simulation results. We will first describe how our data is acquired from a two-stage queueing system simulated in R. Based on the data, the parameters of each server are estimated by our EM algorithm as well as the Newton-Raphson algorithm. The results of each experiment and the analysis in detail are presented in the second part of this chapter .

### 4.1 Data Acquisition

#### 4.1.1 Simulation of the Queueing System

As described in the previous chapter, we need to collect the total service time and total waiting time of each package passing through a tandem queueing system. In our work, instead of building a real computer network, we use simulation to study this discrete-event system [24][28].

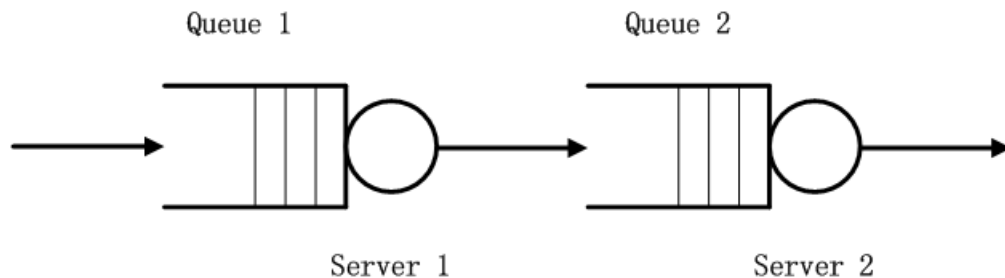


Figure 4.1: Queueing system with two queues and two servers

The essential part of a discrete-event simulation is the maintenance of the event list. In our problem, the event list is a list of scheduled events consisting of three types of events: arrival, departure and intermediate. Different from a single-server queue, the intermediate events are defined to represent the customer's departure from the first server and the arrival at the second server in a two-stage queueing system. We use a data frame to store the event list with each row representing an arrival event, an intermediate event or an departure event. The events are stored in the order of time so that the earliest event will always be handled first.

<b>Event Time</b>	<b>Event Type</b>
0.97	arrival
1.19	intermediate
1.52	arrival
1.60	intermediate
1.61	departure
1.89	departure

Table 4.1: An example of event list

The operation of the queueing system is implemented with the main loop. In each iteration, the head event which is the first event with earliest time is pulled off the event list. The system clock is updated to reflect the event's occurrence. Then based on the type of this head event, different operations may be triggered.

1. If the head event is marked with arrival, a next arrival event is created based on the system's inter-arrival time and inserted into the event list. In the meanwhile, the status of the first queue is checked:
  - If it's empty, the arrival event is directly processed so that an intermediate event is created and inserted into the event list.
  - Otherwise, it enters the first queue and wait to be served.
2. If the head event is marked with intermediate, we need to deal with both of the first queue and the second queue:

- For the first queue, the event is removed. If the queue is still not empty, another intermediate event will be scheduled and inserted into the event list, which represents the next departure from the first queue.
  - For the second queue, we check its status. If the second queue is not empty, the head event entered the queue and wait to be served. If the queue is empty, a departure event is scheduled and inserted into the event list.
3. If the head event is marked with departure, it will be removed from the second queue. The status of the second queue is then checked:
- If it's not empty, another departure event will be created and inserted into the event list.

### 4.1.2 Sampling Plans

For each packet passing through the queueing system, the total system time and total waiting time are recorded. This accounting job is done by incorporating three global vectors: *sysTimeVec*, *waitTimeVec1* and *waitTimeVec2*. The capacities of all these vectors are equal to the number of the packets.

We use *sysTimeVec* to record the total time that each packet spent in the queueing system. The total system time of the *i*th packet is stored in *sysTimeVec*[*i*], which locates at the *i*th position in the vector.

The value of *sysTimeVec*[*i*] is initialized with 0 when the arrival event of the *i*th packet occurs. Later when the departure event of this packet is scheduled, *sysTimeVec*[*i*] will be updated by the difference between the departure time and the arrival time. As described in the previous section, in our simulation, the departure event may be scheduled in two scenarios:

- When the packet enters the second queue and find the server idle, it will get served immediately and the departure event will be scheduled.
- When the packet enters the second queue and find the server busy, it will wait in the queue until the packet before it departs. This time, the departure event of the *i*th packet is scheduled when the departure of the (*i* - 1)th packet occurs.

The two vectors *waitTimeVec1* and *waitTimeVec2* are used to record the packets' waiting time in each server respectively. For the *i*th packet, *waitTimeVec1*[*i*] is

initialized with 0 when the arrival event occurs. If the first queue is empty, the packet gets served immediately and the value of  $waitTimeVec1[i]$  remains 0. Otherwise, it will wait in the queue until the  $(i-1)$ th packet's departure from the first queue. When the  $(i-1)$ th packet leaves the first queue, the intermediate event of this customer is scheduled and  $waitTimeVec1[i]$  will be updated with the difference of the current time and the arrival time.

As the intermediate event of the  $i$ th packet occurs, the value of  $waitTimeVec2[i]$  is initialized with 0. If the second server is idle,  $waitTimeVec2[i]$  will be set to 0. Otherwise, we will start timing from the time that the packet enters the second queue to the time that the departure event is scheduled. And this time period will be assigned to  $waitTimeVec2[i]$ .

## 4.2 Parameter Estimation Experiments

Based on the data representing the total service time of each customer, we carry out parameter estimation experiments using our modified EM algorithm. We also test the Newton-Raphson method.

### 4.2.1 Newton-Raphson Method

Let's first have a brief review of the Newton-Raphson method, which is always considered as a standard approach to solve the likelihood function in an ML estimation.

Newton-Raphson method, also known as Newton's method, is an optimization algorithm for finding approximations to the roots of real-valued functions[46]. For a function  $f(x)$ , based on an initial value  $x_0$ , we can get a better guess of the root as

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},$$

where  $f'$  is the function's derivative. Then the process can be repeated by calculating

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

until we get an estimate of the root that is accurate enough.

Newton-Raphson method can also be applied to multivariate systems. Suppose

that we have an  $N$  dimensional multivariate equation system as

$$\begin{cases} f_1(x_1, \dots, x_N) = f_1(x) = 0 \\ f_2(x_1, \dots, x_N) = f_2(x) = 0 \\ \vdots \\ f_N(x_1, \dots, x_N) = f_N(x) = 0, \end{cases}$$

where  $\mathbf{x} = (x_1, \dots, x_N)^T$ . For convenience, we can also define a vector of functions as

$$\mathbf{f}(\mathbf{x}) = (f_1(x), f_2(x), \dots, f_N(x))^T.$$

Based on the Newton-Raphson method, we will have

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}(\mathbf{x}_n)^{-1} \mathbf{f}(\mathbf{x}_n),$$

where  $\mathbf{J}(\mathbf{x})$  is an  $N$ -dimensional matrix defined as

$$\mathbf{J}(\mathbf{x}) = \frac{d}{d\mathbf{x}} \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \dots & \frac{\partial f_N}{\partial x_N} \end{bmatrix}.$$

We can also get the matrix form of this iteration:

$$\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}_{n+1} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}_n - \begin{bmatrix} & & \\ & \mathbf{J}(\mathbf{x}_n) & \\ & & \end{bmatrix}^{-1} \begin{bmatrix} f_1(\mathbf{x}_n) \\ \vdots \\ f_N(\mathbf{x}_n) \end{bmatrix}.$$

For our problem, in order to estimate the service times' parameters  $\phi_1$  and  $\phi_2$ , we want to use the Newton-Raphson method to obtain the roots of the likelihood equations

$$\frac{\partial \log L_N}{\partial \phi_1} = \sum_{i=1}^N \frac{\frac{\partial f_X(x_i)}{\partial \phi_1}}{f_X(x_i)} = 0 \quad \text{and} \quad \frac{\partial \log L_N}{\partial \phi_2} = \sum_{i=1}^N \frac{\frac{\partial f_X(x_i)}{\partial \phi_2}}{f_X(x_i)} = 0.$$

Based on the solution obtained in the  $i$ th iteration, we can get a better estimate in

the  $(i + 1)th$  iteration as

$$\begin{pmatrix} \phi_1^{i+1} \\ \phi_2^{i+1} \end{pmatrix} = \begin{pmatrix} \phi_1^i \\ \phi_2^i \end{pmatrix} - \begin{bmatrix} \frac{\partial^2 \log L_N}{\partial \phi_1^2} & \frac{\partial^2 \log L_N}{\partial \phi_1 \phi_2} \\ \frac{\partial^2 \log L_N}{\partial \phi_2 \phi_1} & \frac{\partial^2 \log L_N}{\partial \phi_2^2} \end{bmatrix} \begin{pmatrix} \frac{\partial \log L_N}{\partial \phi_1} \\ \frac{\partial \log L_N}{\partial \phi_2} \end{pmatrix}.$$

This process is continued until an estimate is obtained that is accurate enough.

Since the Newton-Raphson method is popular in solving likelihood equations, it has been implemented in many R packages. In our simulation code, a package named **maxLik**[21][42] is used for estimating the parameters.

## 4.2.2 Simulation Results

Experiments are carried out to evaluate the performance of the parameter estimation framework. In each experiment, we use both modified EM algorithm and Newton-Raphson method to estimate the parameters of each server in the queueing system.

For each experiment, the simulated queueing system is set up using the following parameters:

<b>Arrival rate</b>	0.4
<b>Mean service rate of server1</b>	1.4
<b>Mean service rate of server2</b>	0.9

Table 4.2: True values of the parameters

The initial mean service rates of server1 and server2 are set to 1.5 and 1 respectively, which can be considered as a prior knowledge of the system. This is similar to a real world scenario where we are informed with one set of service rates, however, the true values of service rates are different. Note that the arrival rate can be set to any value as long as the system is stable. It only affects the waiting time of packets in the queueing system and has nothing to do with the parameter estimation since our model depends only on total service time.

For the modified EM algorithm, we need to generate random numbers in the E-step, which will cause uncertainty in the estimation result of the parameters  $\phi_1$  and  $\phi_2$ . For example, in Experiment 1 described below, after running the modified EM algorithm for 100 times, the variances of estimates are  $7.9026e - 4$  and  $1.1396e - 4$  for  $\phi_1$  and  $\phi_2$ , respectively. Thus, in our experiments, each estimation process is repeated 100 times and the mean value is taken as the final estimate.

We also define the normalized error of our estimates as

$$error = \frac{|\phi - \hat{\phi}|}{\phi} \quad (4.1)$$

which will be used for measuring the performance of the estimation result.

### **A. Feasibility Study of the Framework**

In the first two experiments, we want to demonstrate the feasibility of our proposed parameter estimation framework. We first build a queueing system consisting of two servers. The servers are set to follow exponential distributions with mean service rates specified in Table 4.2. We generate  $N = 1000$  observations from the queueing system and use  $m = 500$  for the expectation calculation in Eq.(3.19) to estimate the parameters. The estimation result is shown in Figure 4.2.

We then change the second server to follow lognormal distribution with the same mean service rate. The result in Figure 4.3 shows that the EM algorithm works well for the cases that the service time follows either exponential distribution or lognormal distribution.

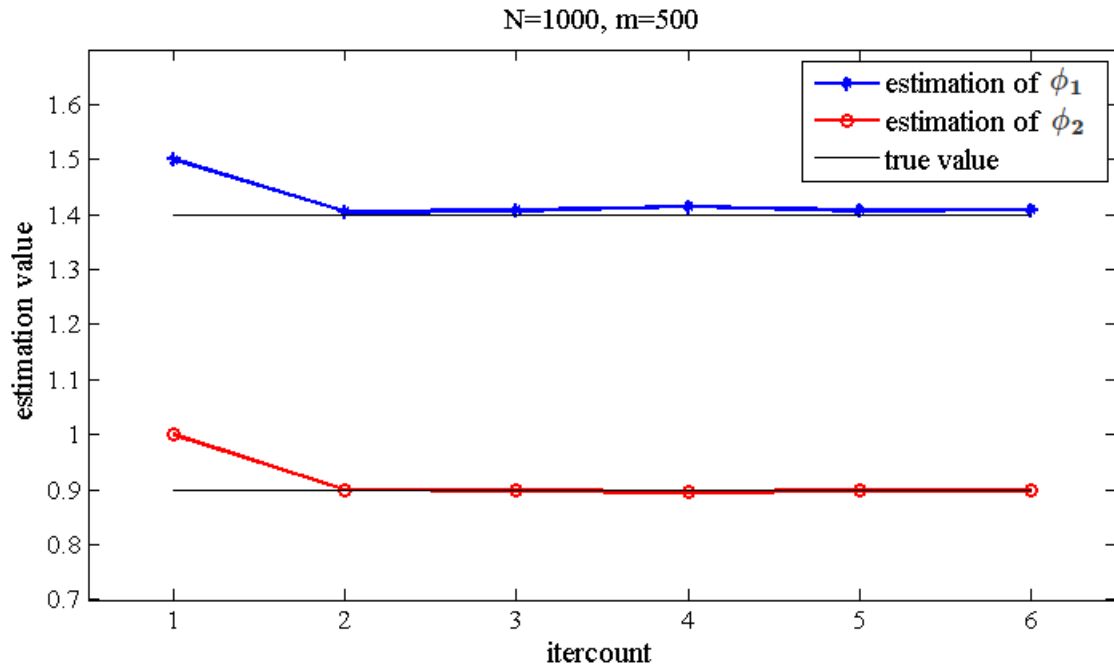


Figure 4.2: Parameter estimation of exponential service rates

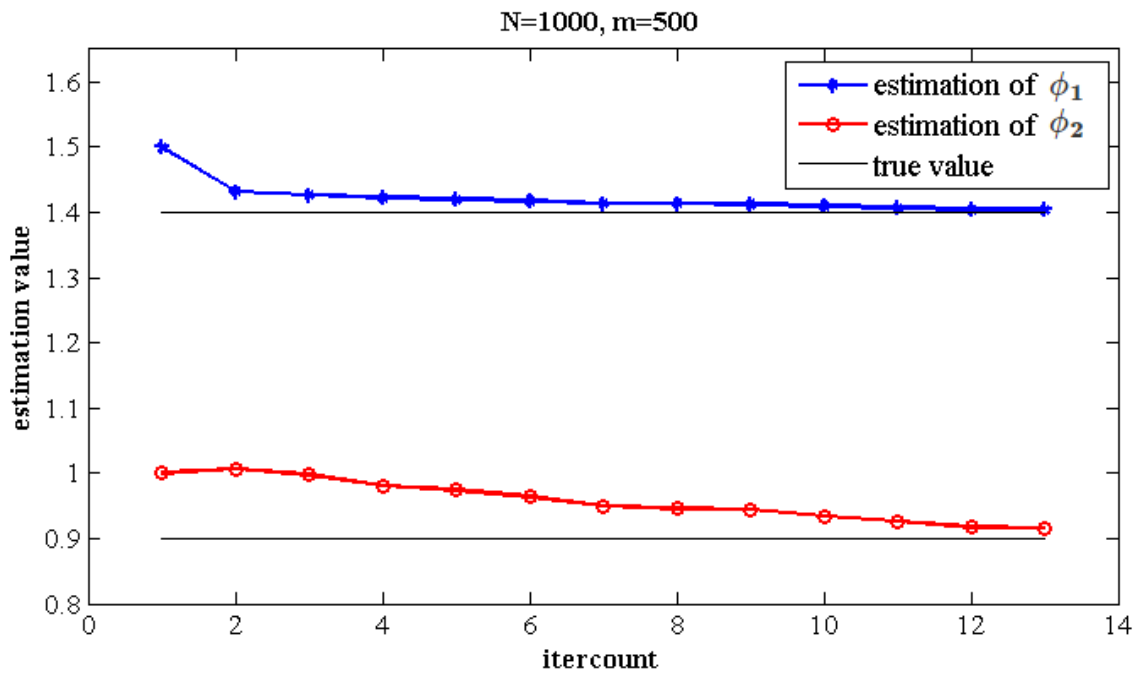


Figure 4.3: Parameter estimation of exponential and lognormal service rates

In the second experiment, we want to test the feasibility of our framework in processing mixture data generated from three components. This time the mixture data is obtained as the sum of three exponential components with  $\phi_1 = 3, \phi_2 = 2, \phi_3 = 1$ . The estimation results are shown in Figure 4.4.

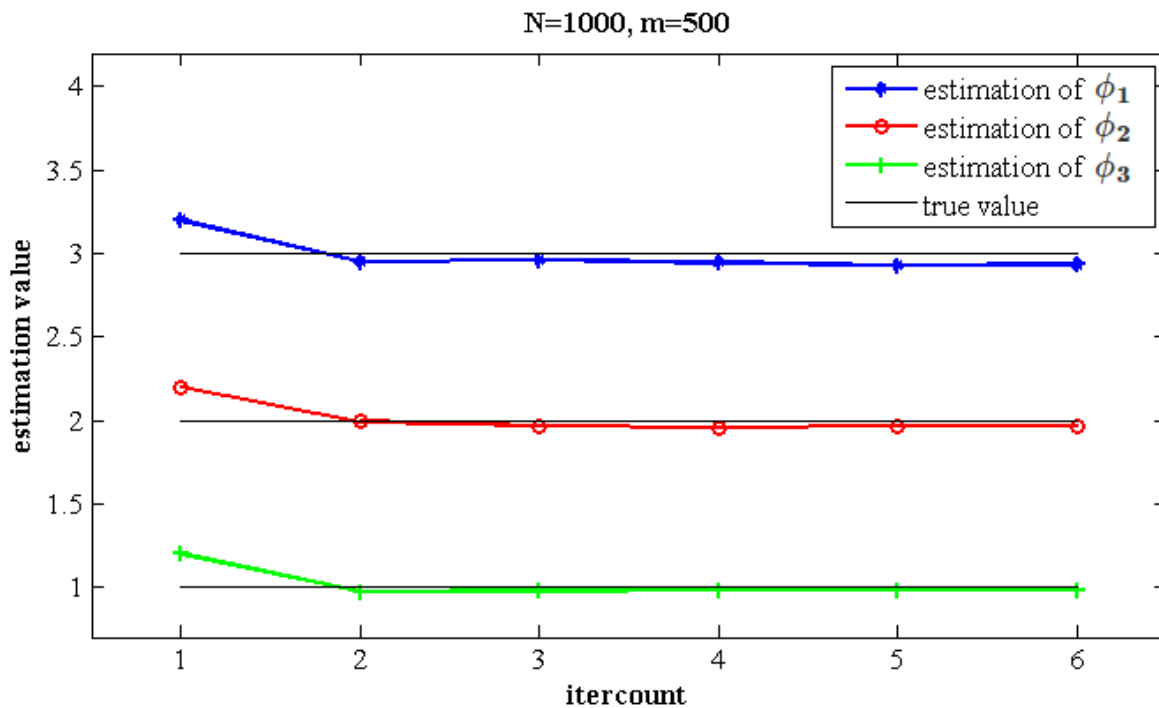


Figure 4.4: Parameter estimation of queueing system with three exponential servers

Through these two experiments, we demonstrate the feasibility of our parameter estimation framework: build a mixture model, derive the likelihood equation then use the modified EM algorithm to solve it (can use Newton-Raphson method as well). The simulation results show that this framework works properly in estimating parameters of different distributions from exponential family and can be potentially applied to queueing systems with  $N$  (more than three) servers.

## B. Influence of the Size of Observations

In this experiment, we compare the performance of the parameter estimation framework with observations of different lengths. A two-stage queueing system is built and both the service times of the two servers are set to follow exponential distributions. We use  $m = 500$  for the approximation of integration. Different set of observations are obtained from the queueing system with the lengths set to  $N = 100, 300, 500, 700, 1000, 2000$ , respectively. Estimation results are shown in Table 4.3 and Table 4.4.

Length	<b>EM(m=500)</b>		<b>NR</b>	
	Value	Error	Value	Error
N=2000	1.4298	0.0213	1.4102	0.0073
N=1000	1.4430	0.0307	1.4159	0.0114
N=700	1.4637	0.0455	1.4395	0.0282
N=500	1.4839	0.0599	1.5169	0.0835
N=300	1.5341	0.0958	1.5971	0.1408
N=100	1.6136	0.1526	1.6507	0.1791

Table 4.3: Estimation of  $\phi_1$  with different N values

Length	<b>EM(m=500)</b>		<b>NR</b>	
	Value	Error	Value	Error
N=2000	0.8927	0.0081	0.9003	0.0003
N=1000	0.9070	0.0078	0.9187	0.0208
N=700	0.9118	0.0131	0.9211	0.0234
N=500	0.9116	0.0129	0.8990	0.0011
N=300	0.9207	0.0230	0.8984	0.0018
N=100	0.9253	0.0281	0.9116	0.0129

Table 4.4: Estimation of  $\phi_2$  with different N values

From the results shown in Table 4.3 and Table 4.4, we find that both EM and NR gave reasonable estimates of  $\phi_1$  and  $\phi_2$ . Before carrying out further analysis of the estimation results, we will examine the observations first. For each  $N$  value, we create the histogram based on the observations and compare it to the theoretical probability density function of the distribution. Comparisons are shown when  $N = 2000, 1000, 500, 100$ . From Figure 4.5, we find that when  $N = 2000$  and  $N = 1000$ , the empirical density estimates approximate the theoretical pdf well. However, when the value of  $N$  decreases to 500 and 100, we will find a large gap between estimation and theoretical pdf. Apparently, a larger sample size will result in a better approximation of the distribution, thus inference that is made from a larger sample size will lead to a better estimation of the parameters.

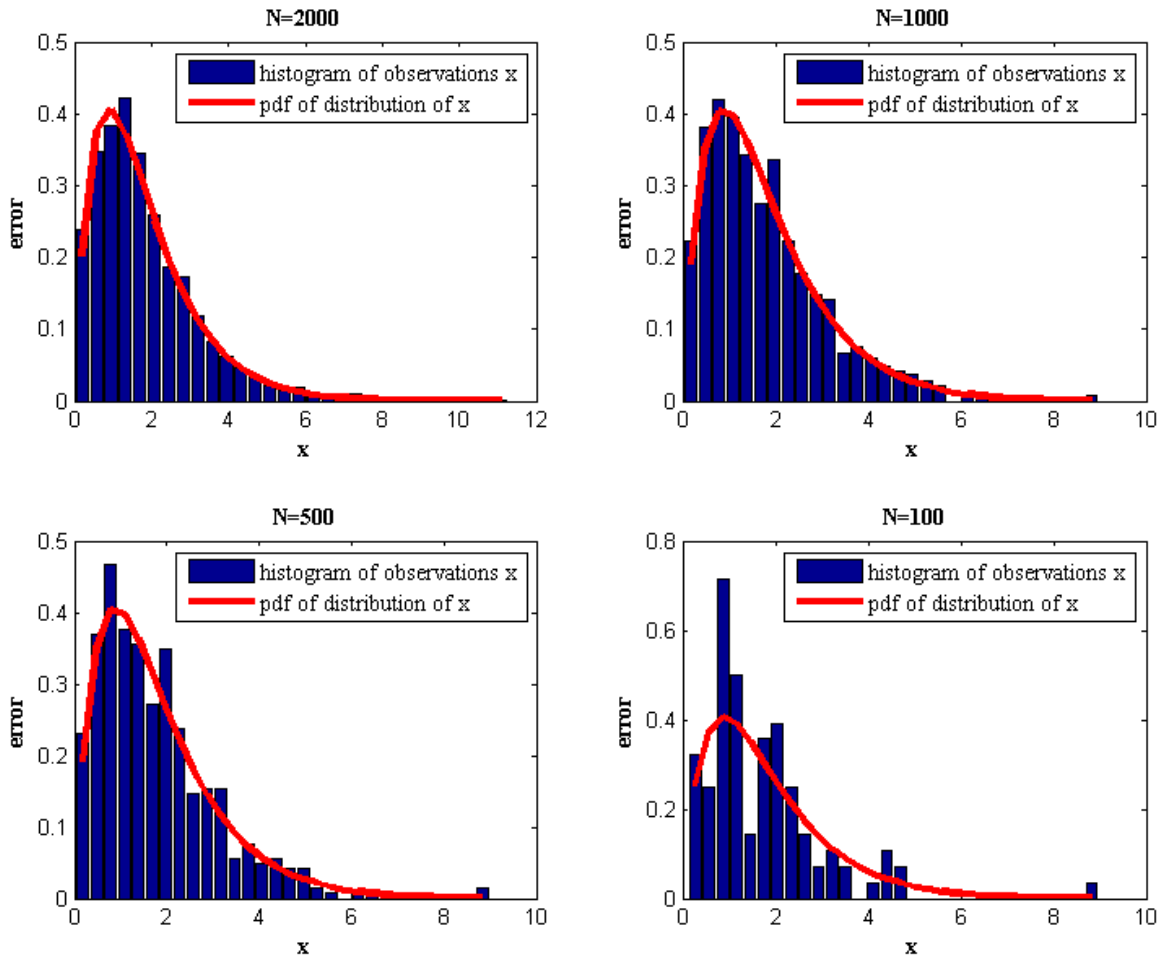


Figure 4.5: Approximations of pdf with different lengths of observations

Now let's take a closer look at the estimation results to see if we really get smaller errors when increasing the size of observations. We use Figure 4.6 to show the errors when we take different lengths of observations. The errors of  $\phi_1$  and  $\phi_2$  are represented by blue lines and red lines, respectively. From the results, we see that EM and NR give comparable estimates for particular  $N$  value. When the size increases from 100 to 1000, both of the two blue lines show that the error decreases.

Nevertheless, the red lines showed an inconsistency. The reason is that even though a larger size of observations will result in a better approximation of the mixture

model as a whole, it's possible that not both of the two components can obtain a better estimation.

This can be demonstrated by the result obtained by NR. By using the **maxLik** package, we get relatively accurate solution to the likelihood equation, thus the NR results are believed to be super close to the true ML estimates. In Figure 4.6, the errors of NR reveal the insufficiency of the two sets of observations when  $N = 700$  and  $N = 1000$  in describing the component with parameter  $\phi_2$ .

To further test our hypothesis, we regard  $\phi_1$  and  $\phi_2$  as a two dimensional vector and calculate the error of  $(\phi_1, \phi_2)$  using

$$error = \frac{\sqrt{(\phi_1 - \hat{\phi}_1)^2 + (\phi_2 - \hat{\phi}_2)^2}}{\sqrt{\phi_1^2 + \phi_2^2}}. \quad (4.2)$$

	<b>EM(m=500)</b>	<b>NR</b>
Length	Error	Error
N=1000	0.0262	0.0147
N=700	0.0389	0.0269
N=500	0.0509	0.0702
N=300	0.0815	0.1184
N=100	0.1292	0.1508

Table 4.5: Estimation of  $(\phi_1, \phi_2)$  with different N values

The results shown in the above table are represented by the two green lines in Figure 4.6, based on which we can conclude that both NR and EM get better results with larger size of observations when the estimations of the two parameters are considered as a whole.

In summary, an increasing size of observations will reduce the systematic error<sup>1</sup>

<sup>1</sup>The systematic error here refers to the error brought by sample data compared to the ideal distribution.

of our framework. From Figure 4.6, we also notice that when we fix the value of  $m$  to  $m = 500$ , there is a gap between the results of NR and the modified EM. We will analyze it in the next part.

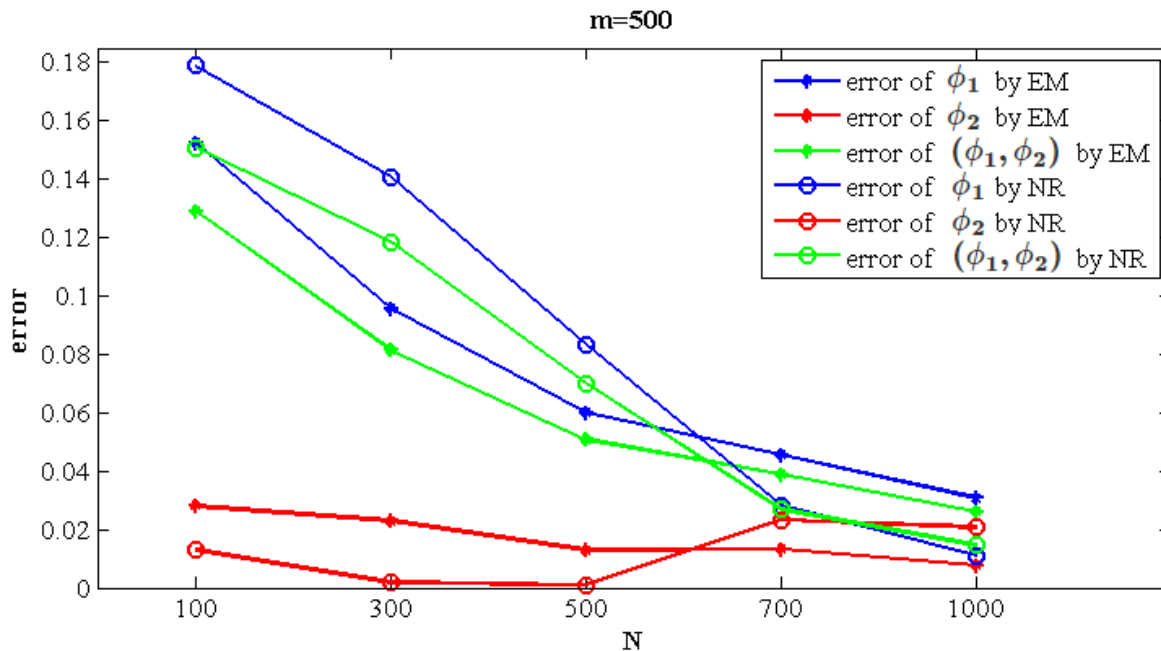


Figure 4.6: Comparison between NR and EM for different lengths of observations

### C. Influence of the Value of $m$

In this part, we evaluate the influence of different  $m$  values when approximating the theoretical expectation. As described in Chapter 3, the Monte Carlo method is used in calculating the expectation in Eq.(3.19). Clearly it yields approximate results and the accuracy depends on the value of  $m$ , which is the number we use for computing the average.

In our experiment, the size of observations is fixed to  $N = 1000$ , since it's good enough to approximate the theoretical pdf of our mixture model. We change the

value of  $m$  from 100 to 500 and estimate the two parameters based on the specific set of observations. The estimation results are shown in Table 4.6 and Table 4.7.

<b>Method</b>	<b>Value</b>	<b>Error</b>
EM(m=100)	1.6169	0.1549
EM(m=200)	1.5048	0.0749
EM(m=300)	1.4778	0.0556
EM(m=400)	1.4524	0.0374
EM(m=500)	1.4430	0.0307
NR	1.4159	0.0114
True Value	1.4	

Table 4.6: Estimation of  $\phi_1$  with different  $m$  values when  $N=1000$

<b>Method</b>	<b>Value</b>	<b>Error</b>
EM(m=100)	0.8521	0.0532
EM(m=200)	0.8860	0.0156
EM(m=300)	0.8950	0.0056
EM(m=400)	0.9044	0.0049
EM(m=500)	0.9070	0.0078
NR	0.9187	0.0208
True Value	0.9	

Table 4.7: Estimation of  $\phi_2$  with different  $m$  values when  $N=1000$

In order to find the trends in our results, we plot the data shown in the above tables in Figure 4.7. We know that increasing the number of simulations in the

Monte Carlo method helps to reduce the random error. And the blue lines and green lines indeed show a decreasing trend. However, the red line in Figure 4.7 seems to contradict our knowledge.

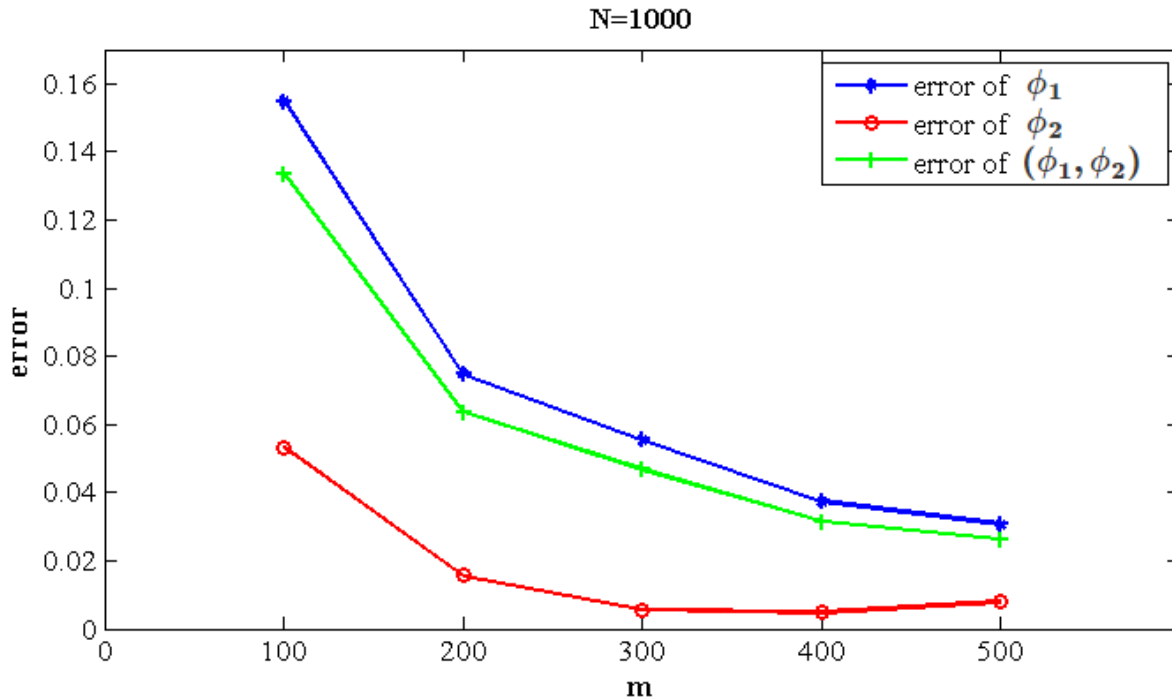


Figure 4.7: Error with different m value

To explain this phenomena, let's take a closer look at the estimation results as shown in Figure 4.8 and Figure 4.9. In each figure, red lines represent the results obtained by NR which are believed to reveal the actual distribution of the samples. Thus, the margins between red lines and black lines can be regarded as the systematic errors introduced by the observations.

As the value of  $m$  increases, the gaps between the blue lines and the red lines become smaller and smaller, indicating that the estimated values obtained by the modified EM get closer to the NR results, but not to the true values. This result consists with our guess: increasing the number of simulations can help to reduce the random error but is not able to reduce the systematic error.

When the estimation of the two parameters are considered as a whole, we get the

results in Figure 4.10. It's clear that as the value of  $m$  becomes larger, the result of modified EM gets closer to that of NR. This also explains why we had the gap in Figure 4.6 when  $m$  was fixed to 500.

In summary, a larger  $m$  will lead to a smaller discrepancy between the EM result and NR result, but not always a better estimate compared to the true value.

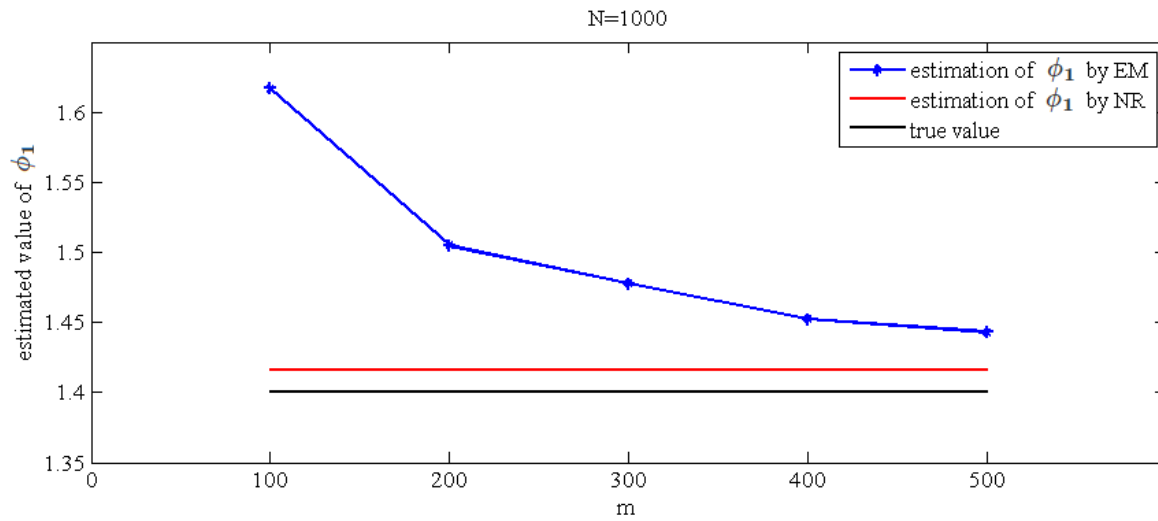


Figure 4.8: Estimation of  $\phi_1$  with different  $m$  values

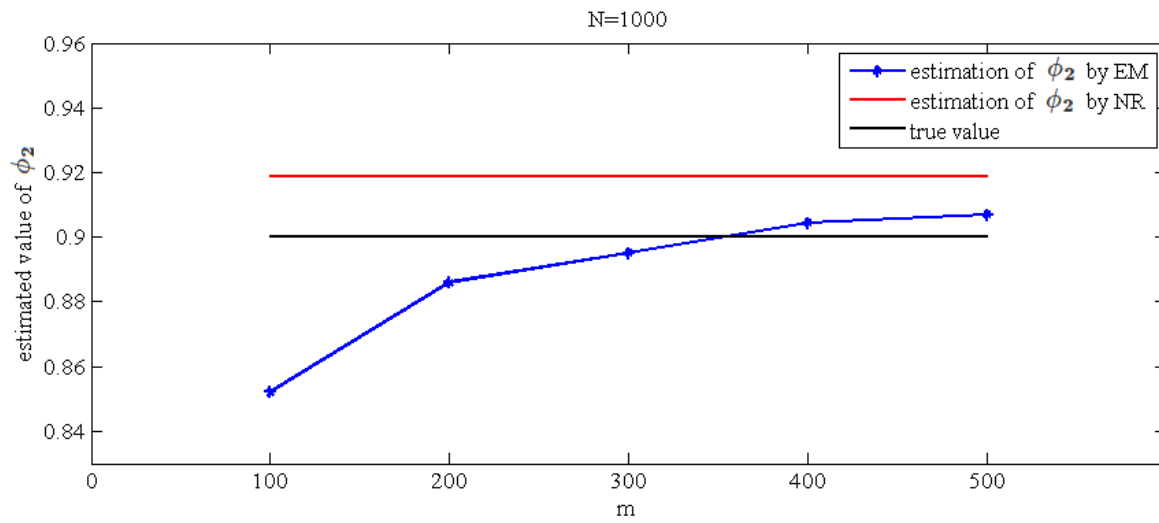


Figure 4.9: Estimation of  $\phi_2$  with different  $m$  values

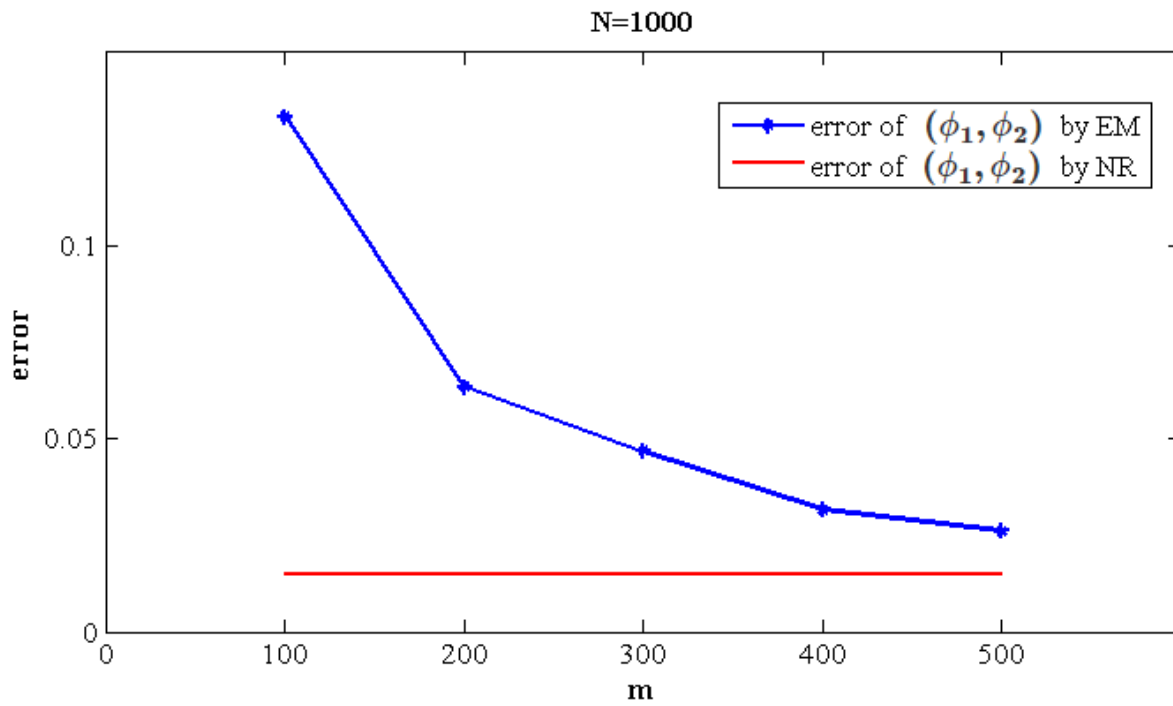


Figure 4.10: Comparison between NR and EM with different  $m$  values

### 4.3 Comparison Between Modified EM and NR

As described in the previous section, we used **maxLik** to implement the NR method and it produced pretty accurate solution to the likelihood equation in our framework. Thus, the gaps between the NR results and the true values are believed to be caused mainly by systematic error that cannot be reduced any more based on this specific sample.

Our modified EM algorithm uses expectation maximization method to solve the likelihood equation. The Monte Carlo simulation was applied for calculating the expectation and the result approaches that of NR when we increase the number of samples in the Monte Carlo method.

From our experiment results shown in Figure 4.6, we can find that NR does not always outperform the EM. From Figure 4.9, we can see that when  $m = 300$  and 400, the random errors are larger than that when  $m = 500$ . However, the random errors in these two scenarios may offset the systematic error by chance, which leads to estimates better than that with NR.

To summarize, we can conclude that:

1. Both EM and NR work properly in our parameter estimation framework.
2. Increasing number of observations will describe the distribution better and result in smaller systematic error.
3. A larger  $m$  value in the Monte Carlo method can help reduce random errors and lead to a better approximation to the EM estimate.
4. NR does not always outperform EM since different sources of errors may neutralize each other by chance.

Finally, we note that the NR method is more complex to implement and is numerically intensive [11], since it involves computation of Hessian matrix (the matrix of second-order partial derivatives of the likelihood with respect to parameter pairs). In contrast, the modified EM is much easier to implement and involves no integral computation.

## Chapter 5

# Conclusions and Future Work

In this work, we investigated the parameter estimation problem for finite mixture model based on additive metrics like end-to-end delay and total waiting time, in the context of queueing network. A general framework for exponential family distributions was proposed, including data acquisition, data modeling and ML estimation. We also provided a modified EM algorithm that can be used to solve the likelihood equation iteratively.

To evaluate the actual performance of the servers in a queueing network, the intuitive method is to collect data of packets' service time at each server and estimate the parameters of each server respectively. However, this becomes impossible in many applications as introduced in Chapter 1, we thus resort to a different approach: collecting each packet' total system time and total waiting time, then the difference will be total service time data. In order to estimate the parameters of every server in the queueing system based on this total service time data, a finite mixture model is used and the ML method is then applied for estimating the parameters.

Extensions to this framework is a modified EM algorithm for solving the likelihood equation in the ML method. Since the distributions of the servers are assumed to vary across the entire exponential family and some may have complicated density functions, we are not able to get explicit expressions for the ML estimates all the time. In this case, iterative methods such as EM and NR can be used to get the ML estimates. Inspired by the work in [7], a modified EM algorithm is applied in this scenario to avoid the main drawbacks of standard EM and NR in terms of computational complexity. In each E-step, the expectation calculation is replaced by Monte Carlo simulation of mean value, which leads to an approximation of standard EM estimates.

Simulation experiments were carried out to test the feasibility and to evaluate the performance of our work. From the results, we conclude that:

1. Our general parameter estimation framework can be applied to a tandem queueing system consisting of a couple of servers that follow different distributions from exponential family.
2. Estimates from both EM and NR are satisfactory enough for our parameter estimation framework.
3. A larger size of observation data will contribute to improving the accuracy of the parameter estimation.

What we have done in this thesis provides basis for future work in several directions.

The first possible extension is to extend the tandem queueing system to a queueing network. For a tandem queueing system, the mixture model with equal weight is used to model the total service time data. However, in a queueing network with more complex topology, a weighted mixture model may be needed and the weight vector should be estimated at the same time.

Another interesting direction is to design accelerators for the modified EM algorithm that we used for estimating parameters in our framework. It's known that the EM algorithm sometimes converges slow and many methods have been proposed to accelerate it [27][23]. For this project, next step for future work may involve combining EM algorithm and Quasi-Newton method to achieve faster convergence.

For the parameters' interchangeability problem, we assumed that prior knowledge is available to help map the estimates to the servers. However, further research may be needed to work out how to create this mapping information from observation data independently.

In the end, the data acquisition part can be improved by setting buffer capacity for each server. Our current simulation platform built in R can simulate a tandem queueing system quite well, however, the buffer capacity is set to infinite implicitly, which is not realistic in real world applications. This also lead to a slow simulation when the arrival rate is high. By incorporating finite capacity buffer and dropping packages as needed, we will get a better simulation tool of a queueing system.

# Bibliography

- [1] Dennis J Aigner. Technical note-parameter estimation from cross-sectional observations on an elementary queuing system. *Operations Research*, 22(2):422–428, 1974.
- [2] Neil Alldrin, Andrew Smith, and Doug Turnbull. Clustering with em and k-means. *University of San Diego, California, Tech Report*, pages 261–95, 2003.
- [3] Carmen Armero. Bayesian inference in markovian queues. *Queueing Systems*, 15(1-4):419–426, 1994.
- [4] Carmen Armero and David Conesa. Prediction in markovian bulk arrival queues. *Queueing Systems*, 34(1-4):327–350, 2000.
- [5] Ishwar V Basawa and Narahari Umanath Prabhu. Large sample inference from single server queues. *Queueing Systems*, 3(4):289–304, 1988.
- [6] IV Basawa, UN Bhat, and J Zhou. Parameter estimation in queueing systems using partial information. *Ohio State University, Tech. Rep.*, 2006.
- [7] IV Basawa, UN Bhat, and J Zhou. Parameter estimation using partial information with applications to queueing and related models. *Statistics & Probability Letters*, 78(12):1375–1383, 2008.
- [8] IV Basawa and NU Prabhu. Estimation in single server queues. *Naval Research Logistics Quarterly*, 28(3):475–487, 1981.
- [9] U Narayan Bhat. *An introduction to queueing theory: modeling and analysis in applications*. Birkhäuser, 2015.
- [10] UN Bhat and IV Basawa. Maximum likelihood estimation in queueing systems. *Advances on Methodological and Applied Aspects of Probability and Statistics*, pages 13–29, 2002.

- [11] Paul P Biemer. *Latent class analysis of survey error*, volume 571. John Wiley & Sons, 2011.
- [12] Charles Byrne and Paul PB Eggermont. Em algorithms. In *Handbook of Mathematical Methods in Imaging*, pages 271–344. Springer, 2011.
- [13] A Bruce Clarke. Maximum likelihood estimates in a simple queue. *The Annals of Mathematical Statistics*, 28(4):1036–1040, 1957.
- [14] Doru Constantinescu, Patrik Carlsson, and Adrian Popescu. One-way transit time measurements. Technical Report 6, Blekinge Institute of Technology, Department of Telecommunication Systems, 2004.
- [15] Neil E Day. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474, 1969.
- [16] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [17] Melynda Eden. A survey of performance modeling and analysis issues in resource management across x86-based hypervisors in enterprise cloud computing deployments. <http://www.cse.wustl.edu/~jain/cse567-11/ftp/hypervsr/>, 2011.
- [18] Phillipa Gill, Zongpeng Li, Anirban Mahanti, Jingxiang Luo, and Carey Williamson. Network information flow in network of queues. In *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MAS-COTS 2008. IEEE International Symposium on*, pages 1–8. IEEE, 2008.
- [19] Tarsaim L Goyal and Carl M Harris. Maximum-likelihood estimates for queues with state-dependent service. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 65–80, 1972.
- [20] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.
- [21] Arne Henningsen and Ott Toomet. maxlik: A package for maximum likelihood estimation in r. *Computational Statistics*, 26(3):443–458, 2011.

- [22] Margareta Holgersson and Ulf Jorner. Decomposition of a mixture into normal components: a review. *International Journal of Bio-Medical Computing*, 9(5):367–392, 1978.
- [23] Mortaza Jamshidian and Robert I Jennrich. Acceleration of the em algorithm by using quasi-newton methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):569–587, 1997.
- [24] Banks Jerry. *Discrete-event system simulation, fifth edition*. Pearson, 2009.
- [25] K Senthamarai Kannan and A Jabarali. Parameter estimation of single server queue with working vacations. *Research & Reviews: Journal of Statistics (Special Issue on Recent Statistical Methodologies and Applications)*, 2:94–98, 2014.
- [26] James F Kurose and Keith W Ross. *Computer networking: a top-down approach*. Addison-Wesley, 2007.
- [27] Kenneth Lange. A quasi-newton acceleration of the em algorithm. *Statistica sinica*, pages 1–18, 1995.
- [28] Norman Matloff. *The art of R programming: A tour of statistical software design*. No Starch Press, 2011.
- [29] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.
- [30] SP Mukherjee and Shovan Chowdhury. Maximum likelihood and bayes estimation in m/m/1 queue. 2006.
- [31] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.
- [32] Terence Orchard, Max A Woodbury, et al. A missing information principle: theory and applications. In *Proceedings of the 6th Berkeley Symposium on mathematical statistics and probability*, volume 1, pages 697–715. University of California Press Berkeley, CA, 1972.
- [33] Stefano Puntoni. Consumers don't understand the relationship between time and speed. <https://hbr.org/2015/11/consumers-dont-understand-the-relationship-between-time-and-speed>, 2015.

- [34] C Radhakrishna Rao. *Linear statistical inference and its applications*, volume 22. John Wiley & Sons, 2009.
- [35] Thomas G Robertazzi. *Computer networks and systems: queueing theory and performance evaluation*. Springer Science & Business Media, 2012.
- [36] Cisco Support. Cisco unified wireless network solution: Videostream deployment guide. <http://www.cisco.com/c/en/us/support/docs/wireless/5500-series-wireless-controllers/112889-cuwns-vidstrm-guide-00.html>, 2013.
- [37] Marco Taboga. *Lectures on probability theory and mathematical statistics*. CreateSpace Independent Pub., 2012.
- [38] Martin A Tanner. *Tools for statistical inference*, volume 3. Springer, 1991.
- [39] D Thiruvaiyaru and IV Basava. Maximum likelihood estimation for queueing networks. *Stochastic Processes and Statistical Inference (Eds., BLS Prakasa Rao and BR Bhat), (1996)*, pages 132–149, 1996.
- [40] Dharma Thiruvaiyaru, Ishwar V Basawa, and U Narayan Bhat. Estimation for a class of simple queueing networks. *Queueing Systems*, 9(3):301–312, 1991.
- [41] D Michael Titterington. *Statistical analysis of finite mixture distributions*. Wiley, 1985.
- [42] Ott Toomet, Arne Henningsen, and Maintainer Ott Toomet. Package maxlik. 2015.
- [43] Vega-BI. Nfv - network functions virtualization. <http://vega-bi.blogspot.ca/2014/08/nfv-network-functions-virtualization.html>, 2014.
- [44] Joanna Wengrzik. *Parameter Estimation for Mixture Models Given Grouped Data*. Bremen, Universität Bremen, Diss., 2012, 2012.
- [45] Wikipedia. Cluster analysis — wikipedia, the free encyclopedia, 2016. [Online; accessed 20-March-2016].
- [46] Wikipedia. Newton’s method — wikipedia, the free encyclopedia, 2016. [Online; accessed 4-June-2016].