Faculty of Engineering and Information Technology
University of Technology, Sydney

Learning from Heterogeneous Data by Bayesian Networks

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Yin Song

April 2014

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

_____

To My Parents, Parents-in-law and Lu

Acknowledgments

First and foremost, I would like to express the deepest appreciation to my supervisor, Professor Longbing Cao, for his professional guidance, persistent help and continuous support throughout my Ph.D study and research. Most importantly, I learned from him many philosophical principles and disciplines that are not only beneficial for academic research, but also apply to many situations in life. I feel very lucky to have him as my advisor and I sincerely hope that we will remain mentor-and-student relationship for many years to come.

I would like to thank all the teachers in Advanced Analytics Institute (AAI), without their generous support this dissertation would not have been possible. I would particularly like to thank Dr. Jian Zhang for his patient guidance, scientific advice and insightful discussion.

I would like to thank all the other colleagues at AAI, for their selfless support on my research, my life and the completion of this dissertation.

My research has also benefited tremendously from various collaborations in other academic and industrial institutions. I would particularly like to thank Dr. Morgan Sangeux in the Royal Children Hospital, Melbourne, for his strong support on imparting me the domain knowledge of clinical gait analysis and assistant of collecting the data, when we collaborated to do a pilot project there. I would also like to thank to Dr Yanchang Zhao and Huaifeng Zhang in Department of Human Services, Canberra, for their great help in both work instruction and living guidance.

Finally, I would like to thank my family, my wife, my parents and parents-

in-law. Without their understanding, support and encouragement, completing this dissertation would be impossible.

Yin Song
Oct. 2013 @ UTS

Contents

List of Figures

List of Tables

Abstract

Non-i.i.d. data breaks the traditional assumption that all data points are independent and identically distributed. It is commonly seen in a wide range of application domains, such as transactional data, pattern recognition data, multimedia data, biomedical data and social media data. Two challenges of learning with such data are the existence of strong *coupling relationships* and *mixed structures (heterogeneity)* in the data. This thesis mainly focuses on learning from *heterogeneous* data, which refers to the non-i.i.d. data with mixed structures. To cater for the learning from such heterogeneous data, this thesis presents a number of algorithms based on Bayesian networks (BNs) that provide an effective and efficient method for representation of heterogeneous structures. A wide spectrum of non-i.i.d. data with different heterogeneity is studied. The heterogeneous data investigated in this thesis includes sequential data of unequal lengths, biomedical data mixed with time series and multivariate attributes, and social media data with both user/user friendship networks and user/item preference matrix. Specifically, for modeling a database of sequential behaviors with different lengths, latent Dirichlet hidden Markov models (LDHMMs), are designed to capture the dependent relationships in two levels (i.e., sequence-level and database-level). To learn the parameters of the model, we propose a variational EM-based algorithm. The learned model achieves substantial or comparable improvement over the-state-of-the-art models on predictive tasks, such as predicting unseen sequences and sequence classification. For learning miscellaneous data in clinical gait analysis, whose data consists of both sequential data and

multivariate data, a correlated static-dynamic model (CSDM) is constructed. An EM-based framework is applied to estimate the model parameters and some intuitive knowledge can be extracted from the model as by-products. Then, for learning more complicated social media data that records both the user/user friendship networks and user/item preference (rating) matrix in social media, we propose a joint interest-social model (JISM). We approximate the lower bound of the likelihood of the observed user/user and user/item interaction data and propose an iterative approach to learn the model parameters under the variational EM framework. The learned model is then used to predict unknown ratings and generally outperforms other comparison methods. Besides the above pure BNs-based models, we also propose a hybrid approach in the context of the sequence anomaly detection problem. This is because the estimation of the parameters of pure BNs-based model usually falls into local minimums, which may further generate inaccurate results for the sequence anomaly detection. Thus, we propose a model-based feature extractor combined with a discriminative classifier (i.e., SVM) to overcome the above issue, which is theoretically proved to have better performance in terms of Bayes error. The empirical results also support our theoretical proof. To sum up, this dissertation provides a novel perspective from Bayesian networks to harness the heterogeneity of non-i.i.d. data and offers effective and efficient solutions to learning such heterogeneous data.

Chapter 1

Introduction

In a data-abundant age, we are dealing with huge complex data in almost every aspect of life. The automatic learning from the data is pivot to knowledge discovery from the data with less human effort, which is the main theme of most machine learning and data mining researches. One of the most common assumptions in traditional machine learning and data mining algorithms is that the data points are independent and identically distributed (i.i.d.). This assumption simplifies the analysis process and enables the development of efficient learning or mining algorithms, which is helpful to a wide range of scenarios. The i.i.d. assumption, however, is poor in many real life scenarios. In these cases, data points have certain relationships that cannot be ignored. Breaking the simple i.i.d. assumption is critical to learning from data in many real-life scenarios. In other words, analyzing such kind of non-i.i.d. data should not ignore the dependent relationships between instances. There are two major characteristics of non-i.i.d. data. One is the underlying strong *coupling relationships* between the data sets, objects, attributes and values and the other is the involvement of *mixed structured* data with strong *heterogeneity* (Cao 2013). (Cao, Ou, Yu & Wei 2010, Cao, Ou & Yu 2011, Song & Cao 2012, Song, Cao, Wu, Wei, Ye & Ding 2012) have made some initial attempts on the learning of *coupling relationships* in non-i.i.d. data. Unfortunately, it is often too costly, or even not practical

to capture such complex *coupling relationships*. Thus, this thesis focuses on learning from the non-i.i.d. data with the characteristic of *heterogeneity*. We term this kind of data as *heterogeneous data* in this thesis. This chapter first introduces heterogeneous data and its characteristics, and then discusses the reasons of applying Bayesian Networks to learn from heterogeneous data, followed by the associated research issues. After that, the highlights and organization of this thesis are given.

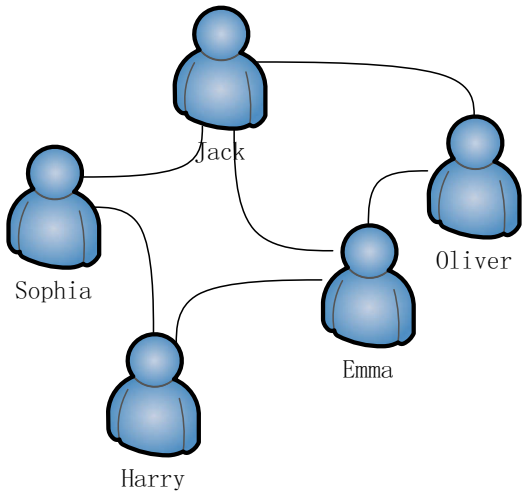
1.1 Heterogeneous Data

The *heterogeneity*/mixed structure existed in heterogeneous data is complex in many real-world scenarios. For example, with the development of social media websites, such as facebook[1] and twitter[2] allows users to share their own interested material (i.e., text, video and audio) with their online friends. The data generated from social media users' behaviors is in different forms and thus heterogeneous.

Here we provide a toy example to illustrate the *mixed structure* of heterogeneous data. In this toy example, as shown in Figure 1.1, there are 5 users (i.e., Sophia, Harry, Emma, Jack and Oliver) and 5 musicians/items (i.e., the Oasis, Eagles, U2, Madonna and Queen). Figure 1.1(a) displays the friendship networks for the users and each edge indicates a friendship relation. For instance, Sophia and Harry are friends. Figure 1.1(b) shows the rating matrix, which reflects the preference of these users. Specifically, each user rates some items on a 5-point integer scale to express the extent of the favor of each item (1, 2, 3, 4 and 5 represent "hate", "don't like", "neutral", "like" and "love", respectively); when the ratings are missing, they are represented with a question mark '?'. For instance, Sophia rates The Oasis band a score of 4, which means she loves the Oasis band; Sophia's rating on the U2 band is missing, which is notated with '?'. Here it is very intuitive that the data

[1] www.facebook.com

[2] www.twitter.com



(a) The Friendship Networks

	The Oasis	Eagles	U2	Madonna	Queen
Sophia	4	3	?	?	5
Harry	3	?	2	?	?
Emma	?	3	?	1	?
Jack	?	?	?	?	?
Oliver	?	?	?	?	?

(b) The Rating Matrix

Figure 1.1: A Toy Example.

shown in this toy example is heterogeneous, since the data is made up of both the user/item preference (rating) matrix and user/user networks. This poses great challenges to model such data since the data structure is mixed and not of the same format.

The non-i.i.d. data with *heterogeneity* is not limited to the above examples, here we briefly list a few application domains where the data is heterogeneous.

- Transactional data: Transactional data is most common in many business applications and usually structured in sequential/temporal order with different lengths, which is heterogeneous for each sequences. Examples can be found in stock market analysis (Fu, Chung, Ng & Luk 2001, Lu, Han & Feng 1998, Tung, Lu, Han & Feng 1999, Tung, Lu, Han & Feng 2003, Plant, Wohlschlager & Zherdin 2009), supermarket transaction analysis (Agrawal & Srikant 1995, Feng, Yu, Lu & Han 2002, Lu, Tseng & Yu 2011), web log mining (Huang & An 2002), and fault detection (Chandola, Banerjee & Kumar 2009).

- Pattern recognition data: The data used for pattern recognition is usually complex. Examples include gesture Recognition (Lee & Kim 1999), trajectory recognition (Gaffney & Smyth 1999), and writer recognition (He, You & Tang 2008), which are usually sequentially and spatially structured in complex and heterogeneous forms.

- Multimedia data: Multimedia data, including audio (Wilpon & Rabiner 1985), image (Ratanamahatana & Keogh 2005) and video (Alon, Sclaroff, Kollios & Pavlovic 2003, Kratz & Nishino 2009, Mahadevan, Li, Bhalodia & Vasconcelos 2010), is inherently structured in squential/temporal and/or spatial order with different lengths.

- Biomedical data: A couple of biomedical problems deal with the analysis of temporal signals such as, functional MRI (fMRI) data (Jansen, White, Mullinger, Liddle, Gowland, Francis, Bowtell & Liddle 2012, Mitchell, Hutchinson, Niculescu, Pereira, Wang, Just & Newman 2004), EEG data (Obermaier, Guger, Neuper & Pfurtscheller 2001) and ECG data (Philips 1993). In most cases, the temporal data is of different lengths and thus heterogeneous.

- Social media data: The social media data is very complex since the involvement of human's behavior and has drawn a lot of attention these days (Liu, Salerno & Young 2008). The social networks (Airoldi, Blei,

Fienberg & Xing 2008) that are the interaction between the users, and the associated users' behaviors in social media (Tang & Liu 2011, Tang & Liu 2009) are naturally heterogeneous.

1.2 Why Learning with Bayesian Networks?

As stated above, the heterogeneous data is complex and has mixed structures. Learning from such heterogeneous data is very challenging and here we list the main challenges as follows:

- Representation of the heterogeneous structure. To break the i.i.d. assumption in analyzing such non-i.i.d. data, we need to utilize proper method to represent mixed/heterogeneous structures that are interested for modeling. This should not be limited to simple heterogeneous structures, such as sequential and temporal structures with different lengths, but also applicable to even more complex structures. To put it in another way, these heterogeneous structures for analysis are diverse and may vary from application to application. Thus, the representation should be flexible and generalized for different types of relationships.

- Uncertainty in the data. The uncertainty can be raised in two aspects. One aspect may be from the data collecting process. Most of the heterogeneous data is collected from real world and easily influenced by noise from human beings and other environmental factors. The other aspect may be caused by the finite size of the data set for learning. Take the speech recognition data as a example, for one word, the utterances we can collect is limited since individuals have various ways to pronounce it. Learning with the above data should consider the generalization of the model on uncollected data.

- Complex Computation. As stated above, the i.i.d. assumption typically simplifies the computation complexity because of omitting the

relationships between data points. The introduction of mixed structures in heterogeneous data brings additional computational cost for learning from the data. For example, the data points are assumed to have sequential relationships, we need to model them with additional efforts.

To overcome the above challenges, in this thesis, we propose several models and algorithms on the basis of *Bayesian networks* (BNs) (Pearl 2000, Heckerman 2008, Dawid 1979). BNs, also known as Bayes networks, belief networks, Bayes(ian) models or probabilistic directed acyclic graphical models, are probabilistic graphical models (PGMs) (Bishop 2006) that represent a set of random variables and their conditional dependencies via directed acyclic graphs (DAGs). Bayesian networks can meet the aforementioned three challenges of learning from heterogeneous data:

- Flexible representation. In BNs, probability distributions are visualized by diagrammatic representations with nodes and arrows. The dependency structure between the variables is indicated by the structure of the diagram, which provides an effective approach for representing heterogeneous relationships. Different structures can be generally modeled as probabilistic dependencies in BNs. Thus, we can simply design new models for different heterogeneous data by utilizing the philosophy of BNs.

- Uncertainty quantification. Since probabilities play a central role in BNs, it is straightforward to quantify the uncertainty in the form of probabilities. In other words, from the perspective of Bayesian interpretation of probabilities (Jaynes 1986), the probability can be a good explanation of the uncertainty that may exist in heterogeneous data.

- Systematic computation. As mentioned above, BNs is the visualized version of probability distributions. Thus, the probability theory, which can be expressed as the sum rule and the product rule (Bayes 1763,

Pearl 2000), is also applicable to BNs. Thus, it is possible to solve complex probabilistic models purely by algebraic manipulation of the above two rules. The graph representation of BNs, however, provides a graphical manipulation (e.g., d-separation (Pearl 2000)) to perform inference in sophisticated models, which could save a lot of unnecessary algebraic calculation.

1.3 Research Goals, Issues and Overview of the Thesis

The main goal of this thesis is to explore a body of principled methods for modeling the heterogeneous data with BNs. Specifically, on the one hand, we will use BNs-based approaches to learn the heterogeneous data in different scenarios with emphasis on different heterogeneous structures. On the other hand, we will apply the learned probabilistic model to perform several predictive tasks, such as behavior prediction and anomaly detection.

The key research issues associated with this thesis, are discussed from the following three perspectives:

- From the data type perspective, we mainly focus on sequential data, miscellaneous Data, and relational data, which are all with heterogeneity. For instance, sequential data is naturally heterogeneous since the sequences are commonly of different lengths; relational data records the user/user and user/item interactions and is typically with mixed structures.

- From the methodology perspective, we investigate both pure BNs-based models and BNs-based hybrid algorithms. For pure BNs-based models, different inference and learning methods need to be designed for data with different heterogeneity. This is because the heterogeneous structures vary from data to data, which requires design of effective and efficient learning algorithms accordingly. In addition, when the parameter

learning of BNs is not accurate enough (Rabiner 1990), combination of BNs and other algorithms should be considered. The principle of combining BNs with other state-of-the-art algorithms, such as support vector machines (SVMs) (Scholkopf & Smola 2002), is valuable to be explored.

- From the application domain perspective, we explore various application domains, ranging from transactional data, pattern recognition data, multimedia data, biomedical data, to social media data.

Before we go deep into these research issues, we make a overview of the thesis.

Chapter 2 first reviews some basic concepts for the BNs, and then lists the related work to the main chapters of this thesis.

In Chapter 3, we study the problem of modeling a database of sequential data. The data is heterogeneous in the sense that the sequences are of different lengths. This kind of data is usually from users' web browsing logs and of different lengths. We develop a hierarchial Bayesian model to characterize such sequential behaviors.

In Chapter 4, we investigate the data modeling problem in clinical gait analysis. The data consists of dynamic data (i.e., sequential data) that records gait characteristics the patients and static data (i.e., multivariate data) that records the physical examination numbers of the patients. The data is miscellaneous and heterogeneous compared to purely sequential data or multivariate data. To jointly model the above data and the correlated relationship among the data points, we develop a probabilistic graphical model for quantifying the correlated relationships between the static and dynamic data.

In Chapter 5, we explore the social recommendation problem whose task is to predict users' preference (ratings) on items by using both the social networks and the preference matrix of the users. The data in this environment is inherently heterogeneous since it involves the data recording both

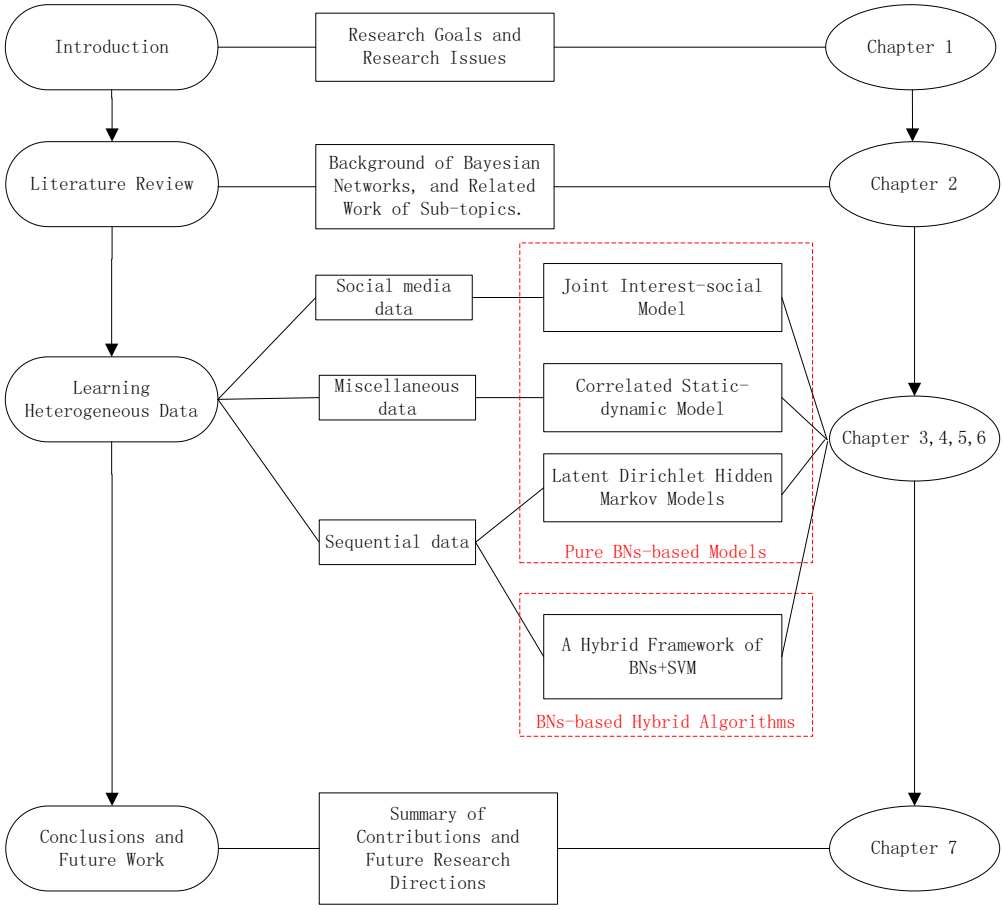


Figure 1.2: Roadmap of the Research Activity

the user/user interactions and the user/item interactions. We design a BNs-based model to jointly model the above data of mixed sources, which provides predictive analytics to unknown ratings.

For some predictive tasks, only modeling the heterogeneous data is not enough to get a satisfactory performance since the parameter estimation usually gets trapped into local minima. Thus, we develop a hybrid framework for identifying abnormal sequences by combining BN-based models and discriminative classifiers in Chapter 6.

Table 1.1 summarizes the research issues we are trying to address in the main body of this thesis. In essence, this thesis provides the concept of heterogeneous data, as well as well-founded and efficient BNs-based algorithms

Table 1.1: Research Issues in Each Chapter.

Research Issues	Detailed Research Issues	Chapter 3	Chapter 4	Chapter 5	Chapter 6
Data Type	Sequential Data	√	√	√	
	Relational Data				√
	Miscellaneous Data	√			
Application Domain	Transactional data		√		
	Pattern Recognition data				√
	Multimedia Data				√
	Biomedical Data	√			√
	Social Media Data			√	
Methodology	Bayesian Networks Only	√	√	√	
	Hybrid with Other methods				√

to model the heterogeneous data with applications on many domains. In other words, this thesis offers a novel and feasible solution for learning with heterogeneous data from the perspective of Bayesian networks.

Finally, we conclude and point out some possible future directions in Chapter 7. For the readers to get a more clear picture of the thesis, Figure 1.2 shows the logic structure of this thesis.

Chapter 2

Background and Related Work

In this chapter, we focus on the preliminaries of Bayesian networks as needed for modeling the heterogeneous data, the related literature for each topic we will explore in the following chapters and discussions of limitations to current existing work.

2.1 Background

Below we first briefly review the concept of Bayesian Networks, then give a toy example to illustrate the problems of latent variable inference and parameter learning, and finally describe a commonly used BNs-based model, hidden Markov Models (HMMs). In essence, this chapter is intended to give readers a context for the use of Bayesian Networks as well as a insight into their general applicability and usefulness.

2.1.1 Bayesian Networks

Bayesian networks (BNs) (Pearl 2000) are directed Probabilistic Graphical Models (PGMs). PGMs are diagrammatic representations of probability distributions and offer several useful properties, such as visualization and inference (Bishop 2006). PGMs consist of nodes (also called vertices) connected by links (also known as edges or arcs). In PGMs, each node represents a

random variable (or group of random variables), and the links express probabilistic relationships between these variables. The joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of the variables, which is captured by the graph. If the links of the PGMS have a particular directionality indicated by arrows, the PGMS become BNs. As a directed version of PGMs, BNS are useful for expressing causal relationships between random variables, which is very intuitive for explaining the generation of the data. In this thesis, we mainly focus on Bayesian networks. The other type of PGMS is undirected graphical models, also known as Markov random fields (Kindermann & Snell 1980, Rue & Held 2005), whose links do not carry arrows and have no directional significance.

The two tasks associated with BNs are *inference* of latent variables and *learning* of parameters[1]:

- *Inference*: probabilistic inference is the process of computing the posterior distribution of some variables given other variables (usually observed). Exact inference methods include the belief propagation (i.e., sum-product) algorithm (Braunstein, Mzard & Zecchina 2005) and its variants. When the exact inference is impractical, some approximate inference methods, such as variation methods and loopy belief propagation, and Monte Carlo methods, such as the Markov chain Monte Carlo algorithm (Andrieu, De Freitas, Doucet & Jordan 2003), are exploited.

- *Learning*: parameters learning is the process of estimating the parameters governing the conditional distributions in BNs. Usually, classical point estimation (i.e., treating parameters as deterministic numbers), such as maximum likelihood estimation (MLE) (Pfanzagl 1994), is applied to parameters learning. A more fully Bayesian approach (Andrieu et al. 2003) to parameters is to treat parameters as unobserved variables

[1]The structure learning (Friedman & Koller 2003) of the graph is can be problematic since the number of different graph structures grows exponentially with the number of nodes, which falls out the scopes of this thesis.

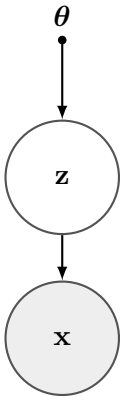


Figure 2.1: A Toy Example of Bayesian Networks

and to compute a full posterior distribution of the variables, which can be computational expensive. When there are latent or hidden variables in BNs, direct maximization of the likelihood with respect to parameters is often impractical since the involvement of them. A classical approach to this problem is the expectation-maximization (EM) algorithm (Baum, Petrie, Soules & Weiss 1970) which alternates computing the posterior distribution of the latent or hidden variables given the observed variables (and deterministic parameters), with maximizing the complete likelihood with respect to parameters given the previously inferred posterior of the latent or hidden variables.

To further illustrate the inference and learning task, we will consider a toy model in the following to clear these concepts.

2.1.2 Inference and Learning

Here we consider a toy example of BNs. In this toy model, we denote the observed data as the variable $\mathbf{x}$, unobserved data (i.e., the latent or hidden variables) as $\mathbf{z}$ and deterministic parameters as $\boldsymbol{\theta}$. All variable are continuous valued. Figure 2.1 defines a very simple generative model, in which $\mathbf{x}$ is generated by $\mathbf{z}$ and $\mathbf{z}$ is governed by $\boldsymbol{\theta}$.

For the above toy model, the corresponding inference and learning task is specified as following:

- *Probabilistic inference of* $\mathbf{z}$: for a particular parametric setting $\boldsymbol{\theta}$, we are interested to infer the posterior distribution of the latent or hidden variables $\mathbf{z}$ given the observed data $\mathbf{x}$ and the parameters $\boldsymbol{\theta}$, which is

$$p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) \tag{2.1}$$

- *Learning of* $\boldsymbol{\theta}$*:* for a MLE of the parameters $\boldsymbol{\theta}$ is expressed as following:

$$\begin{aligned} \hat{\boldsymbol{\theta}} =& \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, f(\boldsymbol{\theta}) \\ =& \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{x}|\boldsymbol{\theta}) \\ =& \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} \end{aligned} \tag{2.2}$$

In real-world problems, no matter how complex the models are, the associated inference and learning problems can always be simplified as the above forms. In the next section, we will review a widely-used Bayesian networks model to make further illustrations.

2.1.3 Hidden Markov Models: An Example of BNs

Here we examine a very common used model for capturing the non-i.i.d. sequential relationships between data instances, hidden Markov models (HMMs) (Rabiner 1990, Ghahramani 1998). They are popular models since their expressive power of real-world behavioral modeling and relatively low computational complexity. HMMs and their variants are not only successful in speech recognition , but also found success in a wide range of fields, from bioinformatics (Baldi & Brunak 2001, Rezek, Gibbs & Roberts 2002, Rezek & Roberts n.d., Zhong & Ghosh 2001, Zhong & Ghosh 2002) to video analysis (Alon et al. 2003, Wang & Singh 2003, Brand, Oliver & Pentland n.d., Velivelli, Huang & Hauptmann 2006, Natarajan & Nevatia 2007, Ding & Fan 2008).

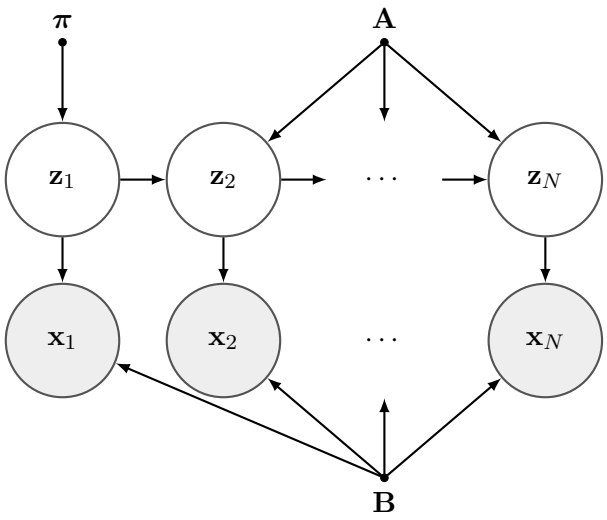


Figure 2.2: The Graphical Model of HMMs

A first-order HMMs can be explained as a simplest form of the dynamic Bayesian networks (Murphy 2002). In HMMs, as shown in Figure 2.3, the state at time t $(1 \leq t \leq N)$ is denoted as $\mathbf{z}_t$ and is hidden (unobserved). $\mathbf{z}_t$ $(1 \leq i \leq N)$ is a K dimensional vector whose elements z_{tk} $(1 \leq k \leq K)$ are 0 or 1 and sum to 1[2] and $z_{tk} = 1$ indicates the hidden state is under state k. $\mathbf{z}_t$ $(1 \leq t \leq t)$ are controlled by the initial state probability distribution $\boldsymbol{\pi}$ whose element $\pi_k = p(z_{1k} = 1)$ $(1 \leq k \leq K)$, and the hidden state transition matrix is $\mathbf{A}$, whose element $a_{ij} = p(z_{t+1,j}|z_{ti}), 1 \leq i, j \leq K$ is the probability for the transition from state i to j. The distribution of observations $\mathbf{x}_t$ $(1 \leq t \leq N)$ is dependent on the states $\mathbf{z}_t$ and parameterized by $\mathbf{B}$. Here we suppose the $\mathbf{x}_t$ is $1 - of - V$ vector and then the element b_{ij} $(1 \leq i \leq K$ and $1 \leq j \leq V)$ of $\mathbf{B}$ equals to $p(x_{tj}|z_{ti})$. Thus, essentially, HMMs consist of a Markov chain of hidden states and the corresponding output of the observations dependent on the hidden states.

[2]Known as $1 - of - K$ vector in (Bishop 2006).

Inference in HMMs

Here we introduce the commonly-used forward-backward algorithm (Rabiner 1990) for inference of the hidden variables $\mathbf{z}_t$ $(1 \leq t \leq N)$ given the observed data $\mathbf{x}_t$ $(1 \leq t \leq N)$ and a set of deterministic parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$.

First, we define some notations for some auxiliary variables (i.e., the forward variables $\alpha(\mathbf{z}_t)$ and the backward variables $\beta(\mathbf{z}_t)$) as following:

$$\alpha(\mathbf{z}_t) = p(\mathbf{x}_1, \cdots, \mathbf{x}_t, \mathbf{z}_t|\boldsymbol{\theta}) \tag{2.3}$$

$$\beta(\mathbf{z}_t) = p(\mathbf{x}_{t+1}, \cdots, \mathbf{x}_N|\mathbf{z}_t, \theta) \tag{2.4}$$

The calculation of $\alpha(\mathbf{z}_t)$ is as follows:

1. Initialization:
$$\alpha(\mathbf{z}_{1k}) = \prod_{v=1}^{V}(\pi_k b_{kx_{1v}})^{z_{1k}} \tag{2.5}$$

2. Induction:
$$\alpha(\mathbf{z}_{tj}) = \prod_{i=1}^{K}\prod_{v=1}^{V} b_{kx_{tv}} \sum_{\mathbf{z}_{t-1}} \alpha(\mathbf{z}_{t-1}) a_{z_{n-1,i},z_{nj}} \tag{2.6}$$

3. Termination:
$$p(\mathbf{x}_{1:N}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha(\mathbf{z}_{Nk}) \tag{2.7}$$

Similarly, we can obtain the calculation of $\beta(\mathbf{z}_t)$ is as follows:

1. Initialization:
$$\beta(\mathbf{z}_{Nk}) = 1 \tag{2.8}$$

2. Induction:
$$\beta(\mathbf{z}_{tj}) = \sum_{j=1}^{K}\prod_{v=1}^{V} \beta(\mathbf{z}_{t+1,j}) b_{j,x_{t+1,v}} a_{z_{ti},z_{t+1,j}} \tag{2.9}$$

By using the above results, the marginal posteriors of $\mathbf{z}_{1:N}$, which is the aim of inference can be expressed as following:

$$\begin{aligned}\gamma(\mathbf{z}_t) =& p(\mathbf{z}_t|\mathbf{x}_{1:N}, \boldsymbol{\theta}) \\ =& \frac{\alpha(\mathbf{z}_t)\beta(\mathbf{z}_t)}{p(\mathbf{x}_{1:N}|\boldsymbol{\theta})}\end{aligned} \tag{2.10}$$

$$\xi(\mathbf{z}_{t-1}, \mathbf{z}_t) = p(\mathbf{z}_{t-1}, \mathbf{z}_t|\mathbf{x}_{1:N}, \boldsymbol{\theta}) = \frac{\alpha(\mathbf{z}_{t-1})p(\mathbf{x}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_{t-1})\beta(\mathbf{z}_t)}{p(\mathbf{x}_{1:N}|\boldsymbol{\theta})} \tag{2.11}$$

Parameter Learning in HMMs

As mentioned before, since involvement of the hidden variables becomes more difficult, and the model is learnt by the well-known EM algorithm, which alternates the following processes:

- E-step: estimating the posterior distribution over hidden variables $\mathbf{z}_{1:N}$ for a particular setting of the parameters $\boldsymbol{\theta}$. This is the inference process we have already reviewed above.
- M-step: re-estimating the best-fit parameters $\boldsymbol{\theta}$ given the inferred posterior distribution over the hidden variables $\mathbf{z}_{1:N}$. The updating formulas are listed in the following:

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{k=1}^{K}\gamma(z_{1k})} \tag{2.12}$$

$$a_{jk} = \frac{\sum_{t=2}^{N}\xi(z_{t-1,j}, z_{tk})}{\sum_{l=1}^{K}\sum_{t=2}^{N}\xi(z_{t-1,j}, z_{nl})} \tag{2.13}$$

$$b_{kv} = \frac{\sum_{t=1}^{N}\gamma(z_{tk})x_{tv}}{\sum_{n=1}^{N}\gamma(z_{tk})} \tag{2.14}$$

The above processes iterates until the criterion of convergence satisfies and the updated parameters are output as the learned parameters of the HMMs. In the rest of the thesis, the EM algorithm and its variants are generally used as the parameter learning algorithm. The process is similar to

the above algorithm used for HMMs, but with proper modification since the likelihood function $f(\boldsymbol{\theta})$ is not always analytical and we need to approximate it with a lower bounder for further computation.

2.2 Related Work

In this section, we briefly review the related work to the main chapters, which investigate learning of different heterogeneous data. Our aim is to show the state-of-the-art related research for each topic.

2.2.1 Sequential Data Modeling

Here we will review existing models that can model sequential behaviors with different lengths in the following[3]. We use conventional notation to represent the graphical model (Bishop 2006). In Figure 4.4, each node represents a random variable (or group of random variables). The directed links express probabilistic causal relationships between these variables. For multiple variables that are of the same kind, we draw a single representative node and then surround this with a plate, labeled with a number indicating that there are many such kinds of nodes. Finally, we denote observed variables by shading the corresponding nodes and the observed variables are shown as shaded nodes.

Hidden Markov Models

Hidden Markov Models (HMMs) drop the difference of parameters between the sequences and assume all the sequences share the same parameters $\boldsymbol{\pi}$, $\mathbf{A}$ and $\mathbf{B}$, as shown in their graphical representation in Figure 2.3.

[3]Please refer to Section 3.2 for the meaning of the notations and we do not repeat them here for conciseness.

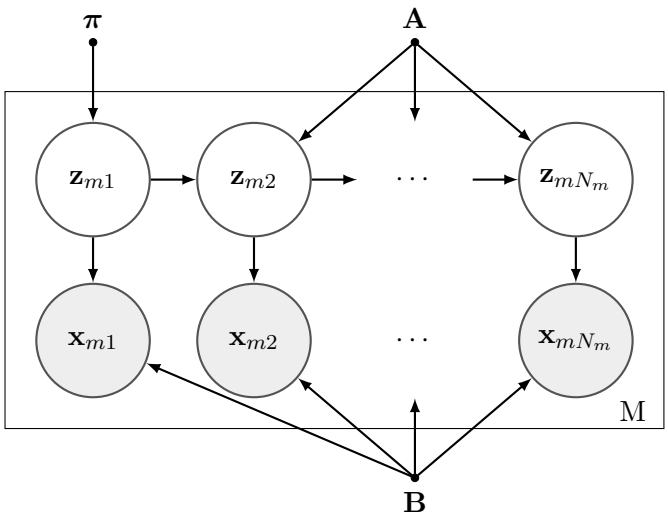


Figure 2.3: The Graphical Model of HMMs.

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) (Blei, Ng & Jordan 2003) is a generative probabilistic model for a set of discrete data, which can be used for modeling sequential behaviors considered in this paper. Its graphical representation is shown in Figure 2.4. It can be seen from the figure that LDA simply ignores the dynamics in the hidden state space.

Variational Bayesian HMMs (VBHMMs)

The graphical model of VBHMMs proposed by Beal and Mckay (Beal 2003, MacKay 1997) is shown in Figure 2.5. VBHMMs assume all the sequences share the same parameters $\boldsymbol{\pi}$, $\mathbf{A}$ and $\mathbf{B}$. The hyper-parameters of VBHMMs are assumed to be known and no algorithm is provided for learning them.

A Hidden Markov Model Variant (HMMV)

As shown in Figure 2.6, the HMMV (Blasiak & Rangwala 2011) assumes sequences share the same parameters $\boldsymbol{\pi}$ and $\mathbf{B}$.

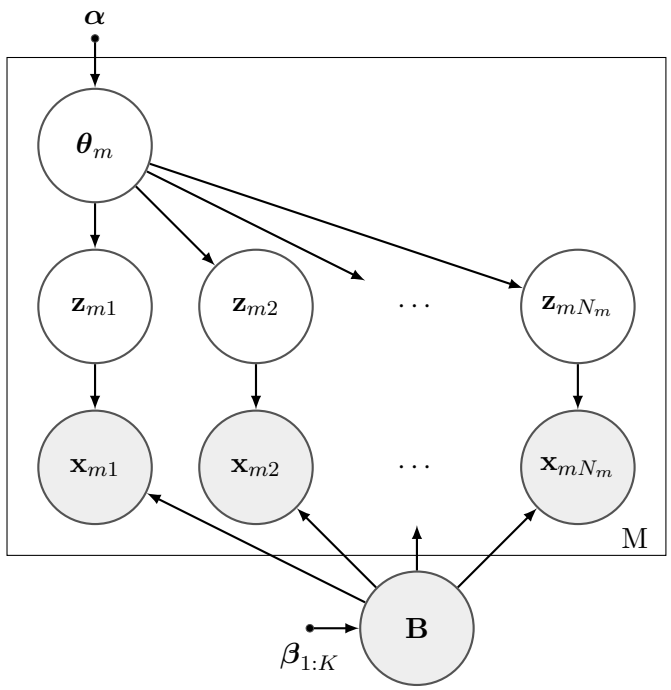


Figure 2.4: The Graphical Model of LDA.

2.2.2 Clinical Gait Data

Recent research in clinical gait analysis (CGA) (Chau 2001, Desloovere, Molenaers, Feys, Huenaerts, Callewaert & Walle 2006, Zhang, Zhang & Begg 2009, Sagawa, Watelain, De Coulon, Kaelin & Armand 2012) have made initial attempts at the automatic discovery of correlated relationships in clinical gait data. They apply machine learning methods, such as multiple linear regression (Desloovere et al. 2006) and fuzzy decision trees (Sagawa et al. 2012), to the data.

Probabilistic models related to modeling gait curves exist. Examples include hidden Markov models (HMMs) (Rabiner 1990) and conditional random fields (CRFs) (Lafferty, McCallum & Pereira 2001).

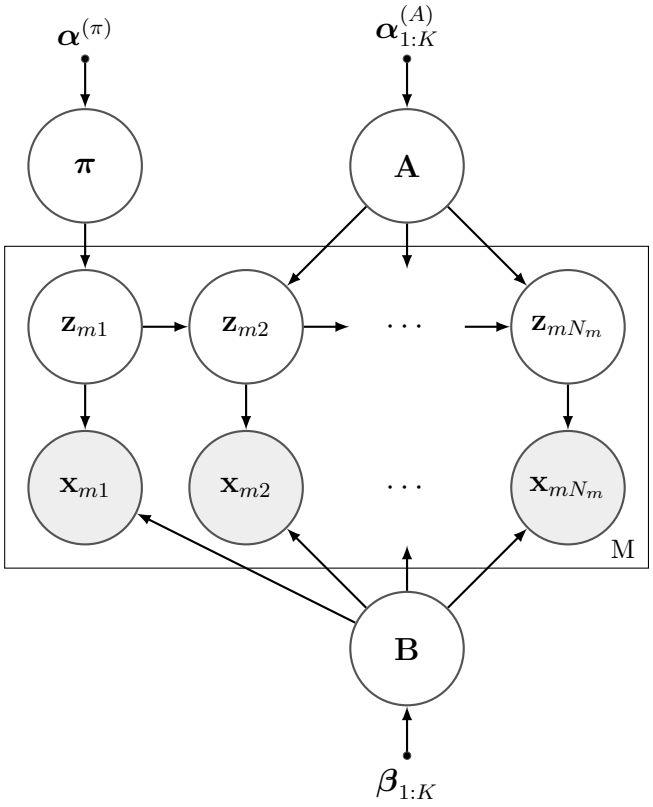


Figure 2.5: The Graphical Model of VBHMMs.

2.2.3 Social Recommendation Data

Here we present two main types of modeling methods related to social recommendation data: one is latent factors models that only use the user/item preference/rating matrix; the other is social recommendation algorithms that utilize both user/user friendship networks and the corresponding user/item preference/rating matrix.

Latent Factor Models for Recommendation

Latent factor models (Mnih & Salakhutdinov 2007, Salakhutdinov & Mnih 2008, Koren, Bell & Volinsky 2009, Agarwal & Chen 2009) are one of the most successful CF-based recommendation approaches when only the rating

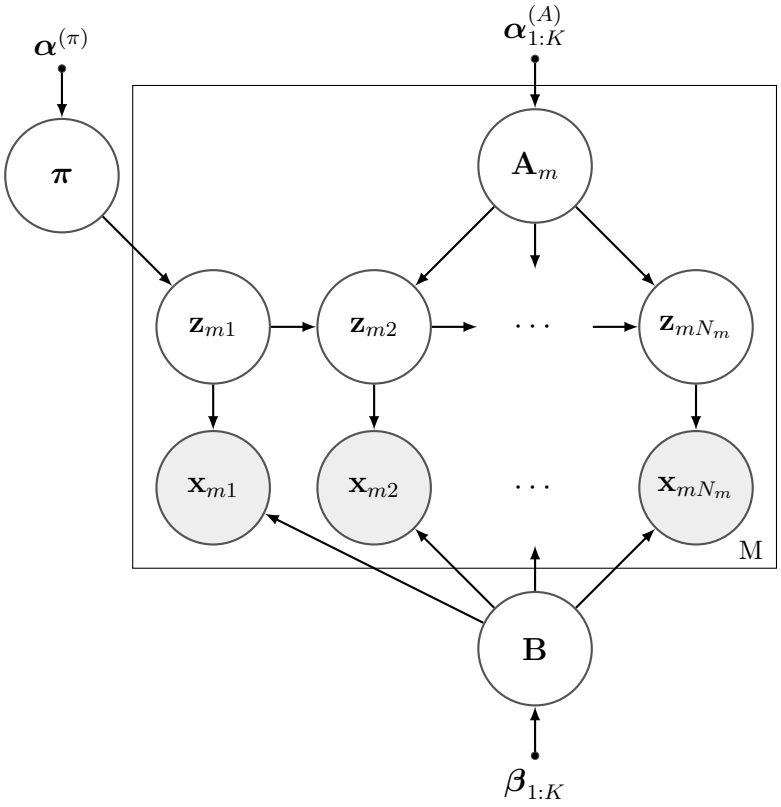


Figure 2.6: The Graphical Model of HMMV.

matrix is available. They are reported by many researchers to have better recommendation performance than other methods, such as the neighborhood methods (Koren et al. 2009). The main philosophy of latent factor models is to uncover latent factors that explain observed ratings. Matrix factorization (Koren et al. 2009) is a representative latent factor model. In matrix factorization, we represent users and items in a homogeneous latent low-dimensional K dimensional space. Specifically, user i is represented by a latent vector $\mathbf{u}_i$ and item j by a latent vector $\mathbf{v}_j$. The prediction of user i's rating on item j is the inner product between their latent representations (i.e., $\hat{r}_{ij} = \mathbf{u}_i^T \mathbf{v}_j$) To compute the latent representations of the users and items given an observed matrix of ratings, a popular approach is to minimize

the regularized squared error loss with respect to $\mathbf{U}$ and $\mathbf{V}$:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i,j} \sigma_{ij}(r_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 + \lambda_u ||\mathbf{u}_i||^2 + \lambda_v ||\mathbf{v}_j||^2 \tag{2.15}$$

where $\mathbf{u}_i$ is the i the column vector of $\mathbf{U}$ and $\mathbf{v}_j$ is the j the column vector of $\mathbf{V}$ ($1 \leq i \leq N$, $1 \leq j \leq M$), σ_{ij} is the indicator function that is equal to 1 if user i rated item j and equal to 0 otherwise, λ_u and λ_v are regularization parameters. Equation 2.15 is usually approximately solved by gradient based approaches (Koren et al. 2009).

An probabilistic generalization of matrix factorization is probabilistic matrix factorization (PMF) (Mnih & Salakhutdinov 2007, Salakhutdinov & Mnih 2008, Shan & Banerjee 2010), which generally assumes the following generative process:

1. For each user i, draw user latent vector $\mathbf{u}_i \sim \mathcal{N}(0, \lambda_u^{-1}\mathbf{I}_K)$, where $\mathbf{I}_K$ is a K-dimensional identity matrix.

2. For each item j, draw item latent vector $\mathbf{v}_i \sim \mathcal{N}(0, \lambda_v^{-1}\mathbf{I}_K)$.

3. For each user-item pair (i, j), draw the rating $r_{ij} \sim \mathcal{N}(\mathbf{u}_i^T \mathbf{v}_j, c_{ij}^{-1})$, where c_{ij} is the precision parameter for r_{ij}.

Gibbs sampling (Geman & Geman 1984) and variational Bayes (Lim & Teh 2007) are two main methods to compute the approximate posterior distribution of the latent representations of the users and items, given an observed matrix of ratings. These inferred posteriors are then used for predictions.

Social Recommendation

Social recommendation is a type of recommendation system in a socialized environment, which recommends items by using both *preference matrix* and friendship networks. In terms of the methods of utilizing the friendship networks, this field of work can be divided into three categories:

- (Guy, Zwerdling, Carmel, Ronen, Uziel, Yogev & Ofek-Koifman 2009, Liu & Lee 2010) generally integrate nearest neighbor algorithms with

simple heuristics on the friendship networks to make item recommendations.

- As an extension of matrix factorization, (Ma, Zhou, Liu, Lyu & King 2011) uses friendship networks as regularization terms adding to the objective function of matrix factorization (i.e., Equation 2.15). Through this way, it constrains the matrix factorization by the available friendship information and reports improved performance compared to purely matrix factorization methods.

- (Yang, Long, Smola, Sadagopan, Zheng & Zha 2011, Ma, Yang, Lyu & King 2008) proposed similar unified latent factor models, SocRec and FIP, for jointly modeling the generating of *rating matrix* and *friendship networks*. Users and items share the same latent space and both the *rating* matrix and *friendship* networks are determined by the latent factors. These models, however, did not provide a clear recipe for estimating the model parameters.

2.2.4 Sequence Anomaly Detection

Anomaly detection has traditionally been an important part of behavior analysis, whose aim is to find abnormal patterns in data that do not conform to expected (normal) behavior (Chandola, Banerjee & Kumar 2009). Most of the traditional anomaly detection techniques focus on static behavioral record or transaction data (Barnett & Lewis 1994). For the purpose of detecting these abnormal sequential behaviors, we should consider the dynamic (heterogeneous) characteristics of sequential data, which is different to anomaly detection in static data.

Several techniques (Budalakoti, Srivastava & Otey 2009, Chandola, Banerjee & Kumar 2009) have been proposed to solve the problem of detecting abnormal sequences. Most of these techniques only consider some of the issues above and can be categorized into two types. One type is to degrade the problem to point (static) anomaly detection. Some techniques in this

category treat a sequence as a vector of attributes assuming that the sequences are of equal length (Blender, Fraedrich & Lunkeit 1997) and then point anomaly detection techniques are applied. This is problematic when the lengths of sequences are not equal. To avoid this problem, different similarity (or distance)-based (Budalakoti, Srivastava, Akella & Turkov 2006) anomaly detection techniques have been proposed. However, the above approaches depend strongly on the definition of similarity (or distance) measure, which could be problematic when the data is very dynamic. For example, the behaviors of ECG signals are changing from time to time, following a stochastic nature. Thus, defining a proper and robust distance measure between sequences in this setting is difficult. To avoid this, another type of sequence anomaly detection techniques tries to model the sequences and thus is model-based. The model-based methods use statistical models to capture the dynamic (heterogeneous) characteristics of the sequences. Representative models, such as Hidden Markov Models (HMMs) (Warrender, Forrest & Pearlmutter 1999), Finite State Automatons (FSAs) (Sekar, Bendre, Dhurjati & Bollineni 2001) and coupled HMMs (CHMMs) (Cao et al. 2010) have been studied in different application domains (e.g., operating system call data, network protocol data and financial data).

2.3 Discussions

In this section, we briefly discuss the limitations of current research according to topics, respectively.

For modeling sequential data with different lengths, current related models have their own restrictions for comprehensively modeling such data. HMMs assume each sequence shares a same set of parameters, and this may overlook the heterogeneous dynamics among the sequences. The LDA model does not consider the temporal-dependent relationships between hidden states. VBHMMs assume sequences share a same set of parameters $\boldsymbol{\pi}_m$, $\mathbf{A}_m$ and $\mathbf{B}_m$ that characterize their dynamics; similarly, the HMMV model as-

sumes sequences share the same set of parameters $\boldsymbol{\pi}_m$ and $\mathbf{B}_m$. However, in order to capture individual characteristics of sequences in a more comprehensive way, it may be better to treat these parameters individually for each sequence. Another limitation of the VBHMMs and the HMMV to note is the assumption of known hyper-parameters.

For clinical gait analysis, previous researchers usually preprocessed the gait data and discarded the dynamic characteristics of that data, which fails to explore the correlated relationship between static data and dynamic curves. To the best of our knowledge, there is few related work to comprehensively exploring this correlated relationship. The existing models focus on modeling dynamic curves, they cannot be applied directly to jointly model the static and dynamic data considering their correlated relationships.

For social recommendation data modeling, one main disadvantage of (probabilistic) matrix factorization (i.e., latent factor models) is that it only uses information from the user/item matrix. This makes it cannot be generalized to predict ratings of users who never give ratings before. The first two categories of social recommendation methods, as mentioned in Section 2.2.3, are usually based on heuristic algorithms and cannot guarantee its performance in general. The third category has limitations that can be summarized in the following two aspects: (1) it represents the users by a homogeneous latent factor space, which may be lack of flexibility to explain the generation of heterogeneous data consisting of user/user networks and the associated user/item matrix. (2) it usually requires tedious work of tuning the parameters, which is not suitable for large-scale applications.

For sequence anomaly detection, the underlying assumption of current widely-used model-based algorithms is that normal sequences conform to the distribution of the model for normal sequences (e.g., BNs-based model) while abnormal ones do not. Although the model-based approaches are reasonable to some extent, we find that directly modeling the normal data has limited discriminative power in identifying abnormal sequences because the estimation of the model parameters may fall into local minimums and

abnormal sequences are highly similar to normal ones. This in turn could result in the degradation of the anomaly detection performance.

This limitations discussed above partly motivate the main chapters of this thesis.

Chapter 3

The Latent Dirichlet Hidden Markov Model

A database of sequential behaviors (sequences) has heterogeneous data structures since sequences may have different lengths. To learn such data, this chapter proposes a generative model, the latent Dirichlet hidden Markov models (LDHMMs). LDHMMs posit that each sequence is generated by an underlying Markov chain process, which are controlled by the corresponding parameters (i.e., the initial state vector, transition matrix and the emission matrix). These *sequence-level* latent parameters for each sequence are modelled as latent Dirichlet random variables and parameterized by a set of deterministic *database-level* hyper-parameters. Through this way, we expect to model the sequence in two levels: the database level by deterministic hyper-parameters and the sequence-level by latent parameters. To learn the deterministic hyper-parameters and approximate posteriors of parameters in LDHMMs, we propose an iterative algorithm under the variational EM framework, which consists of E and M steps. We examine two different schemes, the fully-factorized and partially-factorized forms, for the framework, based on different assumptions. We present empirical results of behavior modeling and sequence classification on three real-world data sets, and compare them to other related models. The experimental results prove that

the proposed LDHMMs produce better generalization performance in terms of log-likelihood and deliver competitive results on the sequence classification problem.

3.1 Introduction

In this chapter we explore the problem of characterizing a database of sequential behaviors (i.e., sequences). An example of such sequential behaviors is the web browsing behaviors of Internet users. Table 3.1 shows some user-browsing data excerpted from the web server logs of msnbc.com. Each row of the table is an ordered list of discrete symbols, each of which represents one behavior made by a user. Here the behavior is described by the categories of web pages requested by the user. For example, User 1 first browses a 'frontpage' page, then visits a 'news' page, followed by visiting 'travel' page and other pages denoted by dots. This form typical sequential behaviors for a user, and other individuals have similar sequential behaviors. For the last decade, many efforts have been made to characterize the above sequential behaviors for further analysis.

Significant progress has been made on behaviour modelling in the field of Sequence Pattern Mining (SPM). Pattern is an expression describing a subset of the data (Piateski & Frawley 1991). Sequential pattern mining discovers frequently occurring behaviors or subsequences as patterns, which was first introduced by Agrawal and Srikant (Agrawal & Srikant 1995). Several algorithms, such as Generalized Sequential Patterns (GSP) (Agrawal & Srikant 1995), SPADE (Zaki 2001) and PrefixSpan (Han, Pei, Mortazavi-Asl, Pinto, Chen, Dayal & Hsu 2001, Han, Pei, Yin & Mao 2004), have been proposed to mine sequential patterns efficiently. Generally speaking, SPM techniques aim at discovering comprehensible sequential patterns in data, which is *descriptive* (Novak, Lavrac & Webb 2009). And there is a lack of well-founded theories to apply the discovered patterns for further data analysis tasks, such as sequence classification and behavior modeling.

In the statistical and machine learning community, researchers try to characterize the sequential behaviors using probabilistic models. The probabilistic models not only describe the generative process of the sequential behaviors but also have a *predictive* property which is helpful for further analytic tasks, such as prediction of future behaviors. One representative model widely used is the hidden Markov models (HMMs) (Rabiner 1990). Usually, each sequence is modelled as an observation generated by an HMM model. In other words, the dynamics of each sequence is represented by a list of deterministic parameters (i.e., initialization prior vector and transition matrix), and there is no generative probabilistic model for these numbers. This leads to several problems when we directly extend HMMs to modeling a database of sequences: (1) the number of parameters for the HMMs grows linearly with the number of sequences, which leads to a serious problem of over-fitting, and (2) it is not clear how to assign probability to a sequence outside of the training set. Although (Rabiner 1990) suggests a strategy for modeling multiple sequences, it simply ignores the difference on parameters between sequences and assumes all the dynamics of sequences can be characterized by one set of deterministic parameters. This could alleviate the problem of over-fitting to some extent, but may overlook the individual characteristics for individual sequences at the same time, which may further deteriorate the accuracy of behavior modeling.

The goal of this chapter is to characterize a database of sequential behaviors preserving the essential statistical relationships for each individual sequence and the whole database, while avoiding the problem of over-fitting. To achieve this goal, we propose a generative model that has both sequence-level and database-level variables to comprehensively and effectively modeling behavioral sequences.

The chapter is organized as follows: Section 3.2 formalizes the problem studied in this chapter, followed by the proposed approach described in Section 3.3. Then, experimental results on several data mining tasks on 3 real-world data sets are reported in Section 4.4. Finally, Section 3.5 summarizes

Table 3.1: An Example of Sequential Behaviors

User	Sequential Behaviors			
1	frontpage	news	travel	...
2	news	news	news	...
3	frontpage	news	frontpage	...
4	frontpage	news	news	...
5	news	weather	weather	...
6	news	health	health	...
7	frontpage	sports	sports	...

this chapter.

3.2 Problem Statement

Here we use the terms, such as 'behaviors', 'sequences' and 'database' to describe a database of sequential behaviors. This is helpful for understanding the probabilistic model derived on the data. It is important to note that the model proposed in this chapter is also applicable to other sequential data that has the similar data forms. In this paper, vectors are denoted by lower case bold Roman or Greek letters and all vectors are assumed to be column vectors except for special explanations. Uppercase bold Roman letters denote matrices while letters in other cases are assumed to be scalar.

- A database $\mathscr{D}$ is a collection of M sequences denoted by $\mathscr{D} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_M\}$.
- A sequence $\mathbf{X}_m$ $(1 \leq m \leq M)$ is an ordered list of N_m behaviors denoted by $\mathbf{X}_m = (\mathbf{x}_{m1}, \mathbf{x}_{m2}, \cdots, \mathbf{x}_{mN_m})$, where $\mathbf{x}_{mn}$ $(1 \leq n \leq N_m)$ is the n^{th} behavior in the sequence $\mathbf{X}_m$. The behaviors in the sequence are ordered by increasing time when behaviors are made.

- A behavior $\mathbf{x}_{mn}$ $(1 \leq m \leq M,\ 1 \leq n \leq N_m)$ is the basic unit of sequential behaviors, defined to be a 1-of-V vector $\mathbf{x}_{mn}$ such that $x_{mnv} = 1$ and $x_{mnu} = 0$ (for all $u \neq v$), which represents an item v from a vocabulary indexed by $\{1, 2, \cdots, V\}$. Each index represents one type of behaviors, such as browsing a 'travel' web page as shown in Table 3.1.

Given a database D of sequential behaviors, the problem of characterizing behaviors is to derive a probabilistic model which preserves the statistical relationships in the sequences and tends to assign high likelihood (given the model) to "similar" sequences.

3.3 The Proposed Model

3.3.1 The Graphical Model

The basic idea of the Latent Dirichlet Hidden Markov Models (LDHMMs) is that the dynamics of each sequence $\mathbf{X}_m$ is assumed to be reflected through a hidden Markov chain $\mathbf{Z}_m = (\mathbf{z}_{m1}, \mathbf{z}_{m2}, \cdots, \mathbf{z}_{mN_m})$[1] parameterized by the corresponding initial prior vector $\boldsymbol{\pi}_m$, transition matrix $\mathbf{A}_m$ and a state-dependent emission matrix $\mathbf{B}_m$. Then $\boldsymbol{\pi}_m$, $\mathbf{A}_m$ ($\mathbf{a}_{mi}$ $(1 \leq i \leq K)$ is the ith row vector of $\mathbf{A}_m$) and $\mathbf{B}_m$ ($\mathbf{b}_{mi}$ $(1 \leq i \leq K)$ is the ith row vector of $\mathbf{B}_m$) can be seen as a lower dimension representation of the dynamics of the sequence. The distribution of these parameters of all sequences are then further governed by database-level Dirichlet hyper-parameters, i.e., $\boldsymbol{\alpha}^{(\pi)}$, $\boldsymbol{\alpha}^{(A)}_{1:K}$ and $\boldsymbol{\beta}_{1:K}$, where $\boldsymbol{\alpha}^{(A)}_{1:K}$ is a matrix whose ith row vector is $\boldsymbol{\alpha}^{(A)}_i$ and $\boldsymbol{\beta}_{1:K}$ is a matrix whose ith row vector is $\boldsymbol{\beta}_i$. To be more specific, for a database of sequential behaviors $\mathscr{D}$, the generative process is as follows[2]:

1. Generate hyper-parameters $\boldsymbol{\alpha}^{(\pi)}$, $\boldsymbol{\alpha}^{(A)}_{1:K}$ and $\boldsymbol{\beta}_{1:K}$.

[1] $\mathbf{z}_{mn}$ $(1 \leq n \leq N_m)$ can be represented by a 1-of-K vector (similar to the form of a behavior $\mathbf{x}_{mn}$) and has K possible hidden states, where K is the number of possible hidden states and is usually set empirically.

[2] Please refer to Appendix A.1 for details of Dirichlet (Dir) and Multinomial distributions.

2. For each sequence index m,

   1) Generate $\boldsymbol{\pi}_m \sim Dir(\boldsymbol{\alpha}^{(\pi)})$, $\mathbf{a}_{mi} \sim Dir(\boldsymbol{\alpha}_i^{(A)})$ and $\mathbf{b}_{mi} \sim Dir(\boldsymbol{\beta}_i)$

   2) For the first time stamp in the sequence $\mathbf{X}_m$:

      (a) Generate an initial hidden state $\mathbf{z}_{m1} \sim Multinomial(\pi)$.

      (b) Generate a behavior from $p(\mathbf{x}_{m1}|\mathbf{z}_{m1}, \mathbf{B}_m)$, a multinomial probability conditioned on the hidden state $\mathbf{z}_{m1}$ and $\mathbf{B}_m$.

   3) For each of other time stamps in the sequence $\mathbf{X}_m$ $(1 \leq n \leq N_m)$:

      (a) Generate a hidden state $\mathbf{z}_{mn}$ from $p(\mathbf{z}_{mn}|\mathbf{z}_{m,n-1}, \mathbf{A_m})$.

      (b) Generate a behavior from $p(\mathbf{x}_{mn}|\mathbf{z}_{mn}, \mathbf{B}_m)$.

Accordingly, the graphical model of LDHMMs is shown in Figure 3.1. As per the graph states itself, there are three levels of modeling in LDHMMs. The hyper-parameters $\boldsymbol{\alpha}^{(\pi)}$, $\boldsymbol{\alpha}_{1:K}^{(A)}$ and $\boldsymbol{\beta}_{1:K}$ are database-level variables, assumed to be sampled once in the process of generating a database. The variables $\boldsymbol{\pi}_m$, $\mathbf{A}_m$ and $\mathbf{B}_m$ $(1 \leq m \leq M)$ are sequence-level variables, denoted as $\boldsymbol{\theta}_m = \{\boldsymbol{\pi}_m, \mathbf{A}_m, \mathbf{B}_m\}$ sampled once per sequence. Finally, the variables $\mathbf{z}_{mn}$ and $\mathbf{x}_{mn}$ are behavior-level variables sampled once for each behavior in each sequence.

3.3.2 Learning the Model

In this section, our goal is to learn the deterministic hyper-parameters of the LDHMMs given a database D, by maximizing the likelihood function $\log p(\mathscr{D}; \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K})$ as following:

$$\sum_{\mathbf{Z}_m, m} \int_{\boldsymbol{\theta}_m} \log p(\mathbf{X}_m, \mathbf{Z}_m, \boldsymbol{\theta}_m; \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K}) \tag{3.1}$$

Direct optimization of the above equation is very difficult since the involvement of latent variables, thus we turn to optimize its lower bound

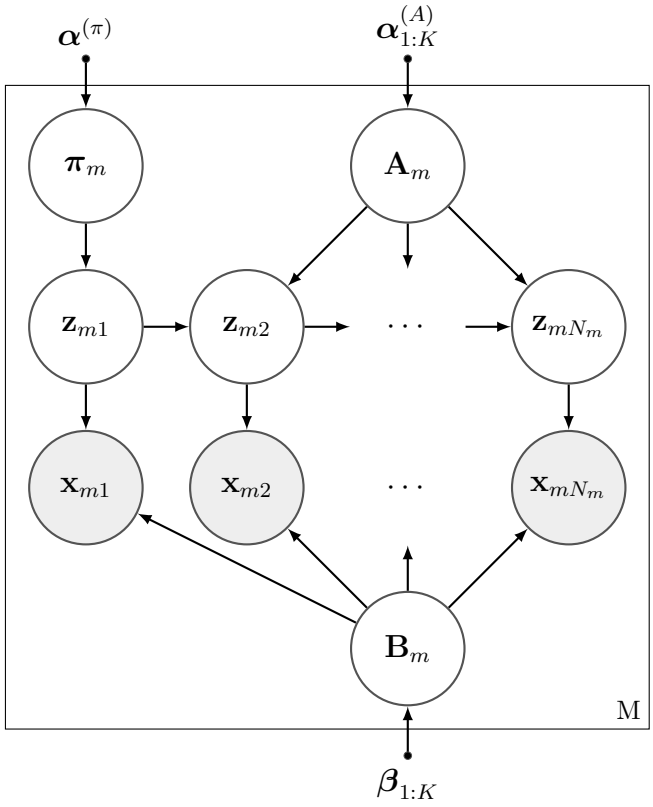


Figure 3.1: The Graphical Model for the LDHMMs.

$L(q, \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:M}, \boldsymbol{\beta}_{1:K})$ given by the Jensen's inequality as follows (Bishop 2006):

$$L(q, \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:M}, \boldsymbol{\beta}_{1:K}) = \sum_{m=1}^{M} [E_q[\log p(\boldsymbol{\theta}_m, \mathbf{Z}_m, \mathbf{X}_m)] - E_q[\log q(\boldsymbol{\theta}_m, \mathbf{Z}_m)]] \tag{3.2}$$

where q is assumed to be a variational distribution function approximate to the posterior distribution of latent variables $\boldsymbol{\theta}_m$ given $\mathbf{X}_m, \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:K}, \boldsymbol{\beta}_{1:K}$ and can be decomposed as $q_1 q_2 \cdots q_M$. Specifically, $q_m = q_m(\boldsymbol{\theta}_m, \mathbf{Z}_m)$ is a variational distribution function approximate to $p(\theta_m, \mathbf{Z}_m | \mathbf{X}_m; \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:K}, \boldsymbol{\beta}_{1:K})$ for the sequence $\mathbf{X}_m$.

Then the lower bound of the likelihood function becomes a function of q, $\boldsymbol{\alpha}^{(\pi)}$, $\boldsymbol{\alpha}^{(A)}_{1:K}$ and $\boldsymbol{\beta}_{1:K}$. To obtain the optimal $\boldsymbol{\alpha}^{(\pi)*}$, $\boldsymbol{\alpha}^{(A)*}_{1:K}$ and $\boldsymbol{\beta}^{*}_{1:K}$ is still

difficult since the involvement of q. Thus, we propose a variational EM-based algorithm for learning the hyper-parameters of the LDHMMs and the algorithm is summarized in Algorithm 3.1. Since the algorithm is under the EM framework, it is guaranteed to increase likelihood after each iteration (Bishop 2006) and thus converges. To be more specific, the variational EM algorithm is a two-stage iterative optimization technique which iterates the E-step (i.e., optimization with respect to q) and M-step (optimization with respect to the hyper-parameters) from lines 1 to 7. For each iteration, the E-step (lines 1-4) fixes the hyper-parameters and optimize the L with respect to q_m for each sequence; while the M-step (line 5) fixes the q and optimizes the L with respect to the hyper-parameters. Through this manner, the optimal hyper-parameters $\boldsymbol{\alpha}^{(\pi)*}$, $\boldsymbol{\alpha}_{1:K}^{(A)*}$ and $\boldsymbol{\beta}_{1:K}^{*}$ are obtained when the iterations are terminated in line 7. It is also important to note that the approximate posteriors of sentence-level parameters (i.e., q) are learned as by-products in E steps.

The following two sections will discuss the details of the procedure E-step and M-step in Algorithm 3.1 and gives out two different implementations.

3.3.3 The E step: Variational Inference of Latent Variables

In this section, we provide the details of the E step, which is to estimate q_m for $(1 \leq m \leq M)$ given the observed sequence $\mathbf{X}_m$ and fixed hyper-parameters $\boldsymbol{\alpha}_{1:K}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K}$ and this process is usually termed as variational inference (Bishop 2006, Ghahramani & Beal 2000, Jaakkola & Jordan 1997, Jordan, Ghahramani, Jaakkola & Saul 1999).

Here we consider two different implementations of variational inference based on different decompositions of q_m:

- A fully-factorized (FF) form.
- A partially-factorized (PF) form.

Algorithm 3.1: The Learning Algorithm for LDHMMs.

Input : An initial setting for the hyper-parameters $\boldsymbol{\alpha}^{(\pi)}$, $\boldsymbol{\alpha}_{1:K}^{(A)}$, $\boldsymbol{\beta}_{1:K}$

Output: Learned hyper-parameters $\boldsymbol{\alpha}^{(\pi)*}$, $\boldsymbol{\alpha}_{1:K}^{(A)*}$, $\boldsymbol{\beta}_{1:K}^{*}$

while *the convergence criterion is not satisfied* **do**
  `// E-step`
  foreach *sequence* $\mathbf{X}_m$ **do**
    `// optimize` L `with respect to` q_m
    $q_m \leftarrow$ `Estep(`$\boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K}, \mathbf{X}_m$`)` ;
  end
  `// M-step`
  `// optimizing` L `with respect to` $\boldsymbol{\alpha}^{(\pi)}$, $\boldsymbol{\alpha}_{1:K}^{(A)}$, $\boldsymbol{\beta}_{1:K}$
  $\boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K} \leftarrow$ `Mstep(`q, $\boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K}$`)` ;
end
$\boldsymbol{\alpha}^{(\pi)*}, \boldsymbol{\alpha}_{1:K}^{(A)*}, \boldsymbol{\beta}_{1:K}^{*} \leftarrow \boldsymbol{\alpha}_{1:K}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K}$;

As shown in Figure 3.2(a), the FF form assumes:

$$q_m(\theta_m) = q_m(\boldsymbol{\pi}_m) q_m(\mathbf{A}_m) q_m(\mathbf{B}_m) q_m(\mathbf{z}_{m1}) q_m(\mathbf{z}_{m2}) \cdots q_m(\mathbf{z}_{mN_m}) \tag{3.3}$$

This is inspired by the standard mean-field approximation in (Jaakkola & Jordan 1997, Jordan et al. 1999).

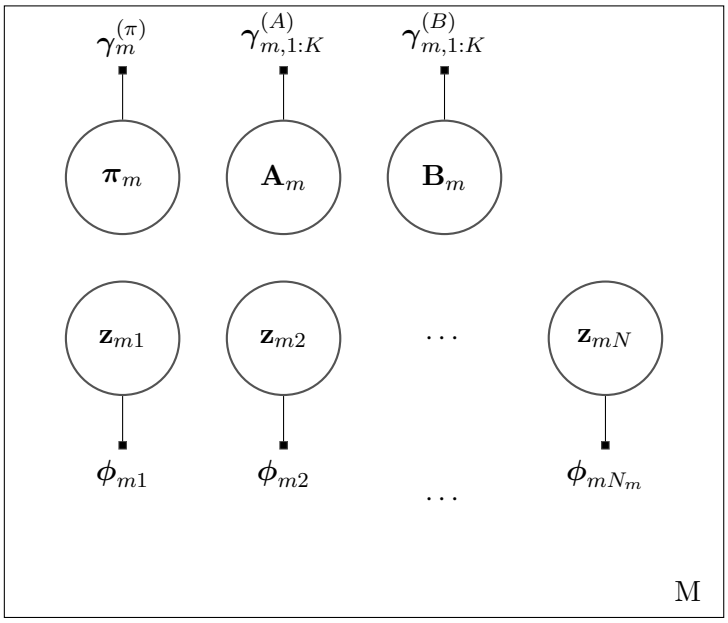
As shown in Figure 3.2(b), the PF form assumes:

$$q_m(\theta_m) = q_m(\boldsymbol{\pi}_m) q_m(\mathbf{A}_m) q_m(\mathbf{B}_m) q_m(\mathbf{Z}_m) \tag{3.4}$$

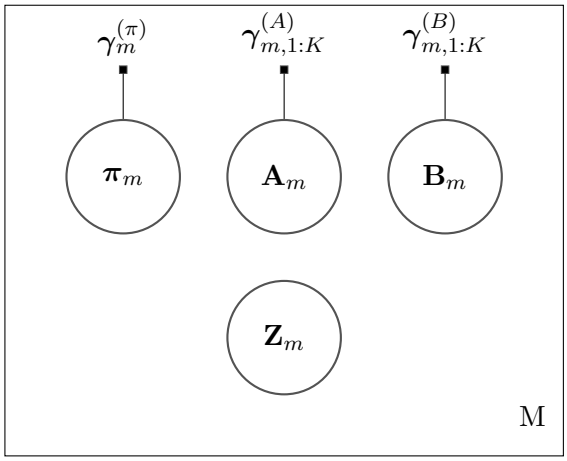
and no further assumption has been made on $q_m(\mathbf{Z}_m)$. This is inspired by the manners proposed in (MacKay 1997, Ghahramani 1997, Beal 2003, Ghahramani & Hinton 2000), which preserves the conditional dependency between the variables of $q_m(\mathbf{Z}_m)$.

E Step: the FF Form

The variational inference process of the FF form yields the following iterations. Please refer to Appendix A.2.1 for the details of the derivation of the



(a)



(b)

Figure 3.2: The graphical models of variational distributions: (a) the FF form, (b) the PF form.

updating formulas.

Fixed $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, Update $\boldsymbol{\phi}_{m,1:N_m}$ $(1 \leq m \leq M)$ ϕ_{m1i} is updated by:

$$\begin{aligned}\phi_{m1i} = exp[&\textstyle\sum_{j=1}^{V}(x_{m1j}(\Psi(\gamma_{mij}^{(B)}) - \Psi(\sum_{v=1}^{V}\gamma_{miv}^{(B)})))\\ &+(\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\textstyle\sum_{j=1}^{K}\gamma_{mj}^{(\pi)}))\\ &+\textstyle\sum_{k=1}^{K}\phi_{m2k}(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K}\gamma_{mij}^{(A)}))]\end{aligned} \tag{3.5}$$

ϕ_{mni} $(2 \leq n \leq N_m - 1)$ is updated by:

$$\begin{aligned}\phi_{mni} = exp[&\textstyle\sum_{j=1}^{V}(x_{mnj}(\Psi(\gamma_{mij}^{(B)}) - \Psi(\sum_{v=1}^{V}\gamma_{miv}^{(B)})))\\ &+(\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\textstyle\sum_{j=1}^{K}\gamma_{mj}^{(\pi)}))\\ &+\textstyle\sum_{i=1}^{K}\phi_{m,n-1,i}(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K}\gamma_{mij}^{(A)}))]\\ &+\textstyle\sum_{k=1}^{K}\phi_{m,n+1,k}(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K}\gamma_{mij}^{(A)}))]\end{aligned} \tag{3.6}$$

ϕ_{mN_mi} is updated by:

$$\begin{aligned}\phi_{mN_mi} = exp[&\textstyle\sum_{j=1}^{V}(x_{mN_mj}(\Psi(\gamma_{mij}^{(B)})\\ &-\Psi(\textstyle\sum_{v=1}^{V}\gamma_{miv}^{(B)}))) + (\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\textstyle\sum_{j=1}^{K}\gamma_{mj}^{(\pi)}))\\ &+\textstyle\sum_{i=1}^{K}\phi_{m,N_{m-1},i}(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K}\gamma_{mij}^{(A)}))]\end{aligned} \tag{3.7}$$

Fixed $\boldsymbol{\phi}_{m,1:N_m}$, $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, $\boldsymbol{\gamma}_{m,1:K}^{(B)}$, Update $\boldsymbol{\gamma}_m^{(\pi)}$

$$\gamma_{mi}^{(\pi)} = \alpha_i^{(\pi)} + \phi_{m1i} \tag{3.8}$$

Fixed $\boldsymbol{\phi}_{m,1:N_m}$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$, Update $\boldsymbol{\gamma}_{m,1:K}^{(A)}$

$$\gamma_{mik}^{(A)} = \alpha_{ik}^{(A)} + \sum_{n=2}^{N}\phi_{m,n-1,i}\phi_{m,n,k} \tag{3.9}$$

Fixed $\boldsymbol{\phi}_{m,1:N_m}$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, Update $\boldsymbol{\gamma}_{m,1:K}^{(B)}$

$$\gamma_{mij}^{(B)} = \alpha_{ij}^{(A)} + \sum_{n=1}^{N}\phi_{mni}x_{mnj} \tag{3.10}$$

where γ is Gamma function and Ψ is the first derivative of the $\log\gamma$ function. For simplicity, the E step for the FF form can be summarized in Procedure Estep.

Procedure Estep($\boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:K}, \boldsymbol{\beta}_{1:K}, \mathbf{X}_m$)

input : A set of the parameters $\boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:K}, \boldsymbol{\beta}_{1:K}$, a sequence $\mathbf{X}_m$

output: The variational distribution q_m

Initialize $\gamma^{(\pi)}_{mi}$ for all i ;
Initialize $\gamma^{(A)}_{mik}$ for all i and k ;
Initialize $\gamma^{(B)}_{mij}$ for all i and j ;
repeat
    Update ϕ_{mni} according to Equation 3.5 to 3.7 for all n and i;
    Update $\gamma^{(\pi)}_{mi}$ according to Equation 3.8 for all i;
    Update $\gamma^{(A)}_{mik}$ according to Equation 3.9 for all i and k;
    Update $\gamma^{(B)}_{mij}$ according to Equation 3.10 for all i and j;
until *convergence*;
$q_m \leftarrow q_m(Z_m)$, $q_m(\boldsymbol{\pi}_m)$, $q_m(\mathbf{A}_m)$, $q_m(\mathbf{B}_m)$;

E Step: the PF Form

The variational inference process of the PF form yields the following iterations. Please refer to Appendix A.2.2 for the details of the derivation of the updating formulas.

Fixed $\boldsymbol{\gamma}^{(\pi)}_m$ and $\boldsymbol{\gamma}^{(B)}_{m,1:K}$ and $\boldsymbol{\gamma}^{(A)}_{m,1:K}$, Update $q_m(\mathbf{Z}_m)$ The relevant (i.e., used in the M step) marginal posteriors of the distribution $q_m(\mathbf{Z}_m)$, i.e., $q_m(\mathbf{z}_{mn}) = \boldsymbol{\gamma}_{mn}$ $(1 \leq n \leq N_m)$ and $q_m(\mathbf{z}_{mn}, \mathbf{z}_{m,n+1}) = \boldsymbol{\xi}_{m,n,n+1}$ $(1 \leq n \leq N_m - 1)$ using the forward-backward algorithm (Rabiner 1990) and the details are described in Appendix A.2.2.

Fixed $q_m(\mathbf{Z}_m)$, $\boldsymbol{\gamma}^{(A)}_{m,1:K}$, $\boldsymbol{\gamma}^{(B)}_{m,1:K}$, Update $\boldsymbol{\gamma}^{(\pi)}_m$

$$\gamma^{(\pi)}_{mi} = \alpha^{(\pi)}_i + q_m(z_{m1i}) \tag{3.11}$$

where $1 \leq m \leq M$ and $1 \leq i \leq K$.

Fixed $q_m(\mathbf{Z}_m)$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$, Update $\boldsymbol{\gamma}_{m,1:K}^{(A)}$

$$\gamma_{mik}^{(A)} = \alpha_{ik}^{(A)} + \sum_{n=2}^{N} q(z_{m,n-1,i}, z_{mnk}) \tag{3.12}$$

Fixed $q_m(\mathbf{Z}_m)$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, Update $\boldsymbol{\gamma}_{m,1:K}^{(B)}$

$$\gamma_{mij}^{(B)} = \alpha_{ij}^{(B)} + \sum_{n=1}^{N} x_{mnj} q_m(z_{mni}) \tag{3.13}$$

where $1 \leq m \leq M$, $1 \leq i \leq K$ and $1 \leq j \leq V$.

The E-step can be also summarized as a procedure similar to Procedure Estep by replacing the corresponding updating formulas. We omit it here for conciseness.

Discussion of computational complexity

The computational complexity for the E-step of approximately inferring the posterior distribution of $\boldsymbol{\pi}_m$, $\boldsymbol{B}_{m,1:K}$ and $\boldsymbol{A}_{m,1:K}$ $(1 \leq m \leq M)$ given the hyper-parameters and the observed behaviors are similar for both the PF and FF forms. Specifically, the computational complexity for inferring the approximate posteriors of $\boldsymbol{\pi}_m$, $\boldsymbol{B}_{m,1:K}$ and $\boldsymbol{A}_{m,1:K}$ $(1 \leq m \leq M)$ are the same for the two forms, which are proportional to $O(MT_EK)$ and $O(MT_EK^2N)$, respectively, where K is the number of hidden states, T_E is the iteration number of E-step, N is the maximum length of all sequences. However, the computational cost for approximate inference of the posterior of $\mathbf{Z}_m$ $(1 \leq m \leq M)$ is slightly different for the two forms. The computational complexity for the PF form is proportional to $O(K^2N)$ while its counterpart of the FF form is proportional to $O(KN)$. Thus, the overall computational complexity for the PF and FF form are $O(MT_E(K+3K^2N))$ and $O(MT_E(K+KN+2K^2N))$, respectively. It is clear that two forms have comparable computational cost and the FF form is slightly faster.

3.3.4 The M Step: Estimation of Hyper-parameters

In this section, we provide the details of the M-step, which is to estimate hyper-parameters $\boldsymbol{\alpha}_{1:K}^{(\pi)}, \boldsymbol{\alpha}_{1:K}^{(A)}, \boldsymbol{\beta}_{1:K}$ given the observed sequence $\mathbf{X}_m$ and fixed variational variables q_m for $(1 \leq m \leq M)$. In particular, it maximizes the lower bound of the log-likelihood L with respect to respective hyper-parameters as follows:

The FF Form

Update $\boldsymbol{\alpha}^{(\pi)}$ Maximizing $\boldsymbol{\alpha}^{(\pi)}$ can be solved by iterative linear-time Newton-Raphson algorithm (Blei et al. 2003, Minka 2000). Define the following variables:

$$g_i = M(\Psi(\sum_{j=1}^{K} \alpha_j^{(\pi)}) - \Psi(\alpha_i^{(\pi)})) + \sum_{m=1}^{M}(\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\sum_{j=1}^{K} \gamma_{mj}^{(\pi)})) \tag{3.14}$$

$$h_i = -M\Psi'(\alpha_i^{(\pi)}) \tag{3.15}$$

$$w = M\Psi(\sum_{j=1}^{K} \alpha_j^{(\pi)}) \tag{3.16}$$

$$c = \frac{\sum_{j=1}^{K} g_j/h_j}{w^{-1} + \sum_{j=1}^{K} h_j^{-1}} \tag{3.17}$$

The updating equation is given by:

$$\alpha_i^{(\pi)*} = \alpha_i^{(\pi)} - \eta\frac{g_i - c}{h_i} \tag{3.18}$$

Procedure Newton-Raphson summarizes the above algorithm, which is an iterative process of updating the value of $\boldsymbol{\alpha}^{(\pi)}$. To be more specific, at the beginning of each iteration, the variables g, h, w, c are calculated by Equation 3.14-3.17 and η to be 1 in lines 2 and 3. Then line 4 updates $\boldsymbol{\alpha}^{(\pi)*}$ by Equation 3.18 and line 5 judges if the updated $\boldsymbol{\alpha}^{(\pi)*}$ falls into the feasible region. If so, it reduces η by a factor of 0.5 in line 6 and updates $\boldsymbol{\alpha}^{(\pi)*}$ in line 7 until it becomes valid. In line 8, update $\boldsymbol{\alpha}^{(\pi)}$ as $\boldsymbol{\alpha}^{(\pi)*}$ for the next iteration.

Procedure Newton-Raphson($\gamma^{(\pi)}_{1:M,1:K}, \boldsymbol{\alpha}^{(\pi)}, T_M$)

input : $\gamma^{(\pi)}_{1:M,1:K}$, $\boldsymbol{\alpha}^{(\pi)}$, Number of iterations T_M
output: Updated $\alpha^{(\pi)*}_i$

for *iter* $\leftarrow 1$ **to** T_M **do**
  Update g_i, h_i, w, c for all i according to Equation 3.14-3.17 ;
  $\eta \leftarrow 1$;
  Update $\alpha^{(\pi)*}_i$ according to Equation 3.18;
  while *Any* $\alpha^{(\pi)*}_i < 0$ **do**
    $\eta \leftarrow 0.5\eta$;
    Update $\alpha^{(\pi)*}_i$ according to Equation 3.18;
  end
  $\alpha^{(\pi)}_i \leftarrow \alpha^{(\pi)*}_i$;
end

Procedure Mstep($q, \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:K}, \boldsymbol{\beta}_{1:K}$)

input : $q, \boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:K}, \boldsymbol{\beta}_{1:K}$
output: A set of the parameters $\boldsymbol{\alpha}^{(\pi)}, \boldsymbol{\alpha}^{(A)}_{1:K}, \boldsymbol{\beta}_{1:K}$

Call Procedure Newton-Raphson to update $\boldsymbol{\alpha}^{(\pi)}$;
Call Procedure Newton-Raphson to update $\boldsymbol{\alpha}^{(A)}_{1:K}$;
Call Procedure Newton-Raphson to update $\boldsymbol{\beta}_{1:K}$;

Update $\boldsymbol{\alpha}^{(A)}_{1:K}$ Similarly, the estimation of $\boldsymbol{\alpha}^{(A)}_i$ $(1 \leq i \leq K)$ can be solved by the Procedure Newton-Raphson with changes on Equation 3.14-3.18 (i.e., replace $\boldsymbol{\alpha}^{(\pi)}$ by $\boldsymbol{\alpha}^{(A)}_i$ and $\boldsymbol{\gamma}^{(\pi)}$ by $\boldsymbol{\gamma}^{(A)}_i$).

Update $\boldsymbol{\beta}_{1:K}$ Similarly, the estimation of $\boldsymbol{\beta}_i$ $(1 \leq i \leq K)$ can be done by the Procedure Newton-Raphson with changes on Equation 3.14-3.18 (i.e., replace $\boldsymbol{\alpha}^{(\pi)}$ by $\boldsymbol{\beta}_i$ and $\boldsymbol{\gamma}^{(\pi)}$ by $\boldsymbol{\gamma}^{(B)}_i$).

The M-step can be summarized in Procedure Mstep.

The PF Form

The process is the same as the above process.

3.4 Empirical Study

In this section, we apply the proposed LDHMMs in several data mining tasks, such as sequential behavior modeling and sequence classification. To be more specific, firstly, we use two public-available data sets from web-browsing logs to study the data mining tasks. Secondly, we adopt a public-available biological sequence data set to study the problem of sequence classification. All algorithms were implemented in matlab[3] and performed on a 2.9GHz 20MB L3 Cache Intel Xeon E5-2690 (8 Cores) cluster node with 32GB 1600MHz ECC DDR3-RAM (Quad Channel), running on a Red Hat Enterprise Linux 6.2 (64bit) operating system.

3.4.1 Sequential Behavior Modeling

Data Sets

The Entree Data Set This data set[4] records users' interactions with the Entree Chicago restaurant recommendation system from September, 1996 to April, 1999. The sequential behaviors of each user are his/her interactions with the system, i.e. their navigation operations. The characters L-T encode 8 navigation operations as shown in Table 3.2. We use a subset of 422 sequences whose lengths vary from 20 to 59.

The MSNBC Data Set This data set[5] describes the page visits of users who visited msnbc.com on September 28, 1999. Visits are recorded at

[3]The code is publicly available on https://sites.google.com/site/yinsong1986/codes.

[4]Available at http://archive.ics.uci.edu/ml/datasets/Entree+Chicago+Recommendation+Data

[5]Available at http://archive.ics.uci.edu/ml/datasets/MSNBC.com+Anonymous+Web+Data.

Table 3.2: The Codebook of Navigation Operations

Code	Navigation Operations
L	browse from one restaurant to another in a list
M	search for a similar but cheaper restaurant
N	search for a similar but nicer one
P	search for a similar but more traditional one
Q	search for a similar but more creative one
R	search for a similar but more lively one
S	search for a similar but quieter one
T	search for a similar but different cuisine one

the level of URL category (The 17 categories are 'frontpage', 'news', 'tech', 'local', 'opinion', 'on-air', 'misc', 'weather', 'health', 'living', 'business', 'sports', 'summary', 'bbs' (bulletin board service), 'travel', 'msn-news', and 'msn-sports'.) and are recorded in a temporal order. Each sequence in the data set corresponds to page viewing behaviors of a user during that twenty-four hour period. Each behavior recorded in the sequence corresponds to the category of the user's requesting page. We use a subset of 31071 sequences whose lengths vary from 20 to 100.

Evaluation Metrics

We learned the LDHMMs of the proposed two different learning forms (denoted as LDHMMs-ff and LDHMMs-pf) and related models (i.e., HMMs, LDA, VBHMMs and HMMV), on the above two data sets to compare the generalization performance of these models. Our goal is to achieve high likelihood on a held-out test set. Thus, we computed the log-likelihood of a held-out test set to evaluate the models given the learned deterministic hyper-parameters/parameters. In particular, for LDHMMs, we first learned their deterministic database-level hyper-parameters according to Algorith-

m 3.1 using the training data; then approximately inferred the sequence-level parameters of the testing data by applying Procedure Estep with the learned hyper-parameters; and finally computed the log-likelihood of the test data as Equation A.3 using the learned hyper-parameters and inferred parameters. For other models, we used similar processes adjusting to their learning and inference algorithms. A higher log-likelihood indicates better generalization performance. We performed 10-fold cross validation on the above two data sets. Specifically, we split the data into 10 folds. Each time we held out 1 fold of the data for testing and trained the models on the remained 9-folds, and this process was repeated for 10 times. We report the averaged results of the 10-fold cross validation in the following.

Comparison of Log-likelihood on the Test Data Set

The results for different number of hidden states K on the Entree data set is shown in Figure 3.3. As seen from the chart, the LDHMMs-pf consistently performs the best and LDHMMs-ff generally has the second best performance (only slightly worse than HMMs sometimes). Similar trend can be observed in Figure 3.4. Both LDHMMs-ff and LDHMMs-pf perform better than the other models while LDHMMs-pf has a slightly better performance. This is because the PF form may have a more accurate approximation in these data sets. In summary, the proposed LDHMMs has a better generalization performance compared to other models. To further validate the statistical significance of our experiments, we also perform the paired t-test (2-tail) between LDHMMs-pf, LDHMMs-ff and other models over the perplexity of the experimental results. The p-level of t-tests is always smaller than 0.01, which proves the improvements of LDHMMs over other models are statistically significant.

Comparison of Computational time for the two forms

Since the computational complexity of related models are much lower than the proposed model due to their simpler structures, here we focus on the

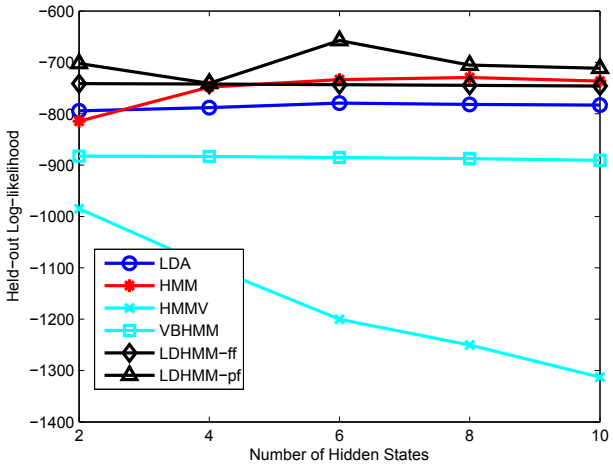


Figure 3.3: Log-likelihood Results on the Entree Data Set for the Models.

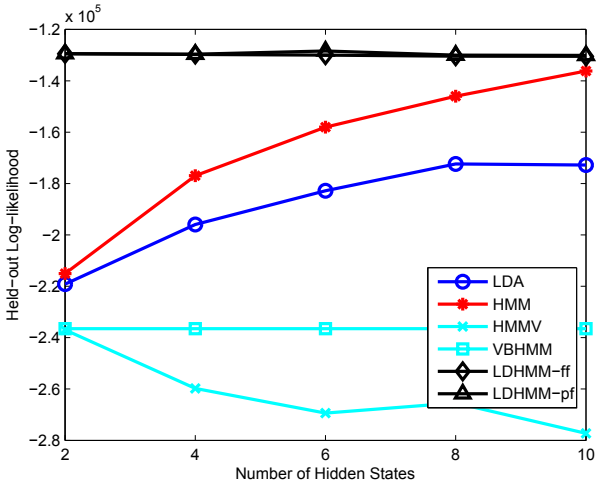
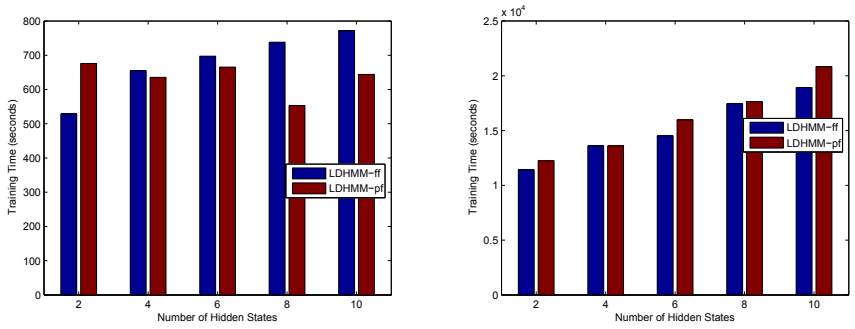


Figure 3.4: Log-likelihood Results on the MSNBC Data Set for the Models.

comparison of the proposed two forms. Figure 3.5 shows the comparison of training time on the two used data sets. Qualitatively speaking, the two approaches have similar computational time. But sometimes, the PF form is faster the FF form, which seems to be contradict to our theoretical analysis in Section Estep. However, in practice, the stopping criterion used in the



(a) (b)

Figure 3.5: Comparison of Training Time for the LDHMMs on (a) Entree Data Set, (b) MSNBC Data Set.

EM algorithm which may cause the iteration to stop earlier. Since the PF form may converge faster than the FF form does, it may need less numbers of E and M steps. Thus, it may converge faster than the FF form in those cases.

Visualization of LDHMMs

It is also important to obtain an intuitive understanding of the complex model learned. LDHMMs have database-level hyper-parameters, (i.e., $\boldsymbol{\alpha}^{(\pi)}$, $\boldsymbol{\alpha}^{(A)}_{1:K}$ and $\boldsymbol{\beta}_{1:K}$), which can be seen as database-level characteristics of the sequences; sequence-level variational parameters (i.e., $\boldsymbol{\gamma}^{(\pi)}_{m}$, $\boldsymbol{\gamma}^{(B)}_{m,1:K}$ and $\boldsymbol{\gamma}^{(A)}_{m,1:K}$), which can be seen as sequence-level characteristics of each individual sequence. To visualize LDHMMs, we plot Hinton Diagrams for these parameters, each of which is represented by a square whose size is associated with the magnitude. Figure 3.6 shows a sample visualization from the Entree data set when $K = 6$. The left diagrams represent the database-level hyper-parameters $\boldsymbol{\alpha}^{(A)}_{1:K}$, $\boldsymbol{\beta}_{1:K}$ and $\boldsymbol{\alpha}^{(\pi)}$ from the top to the bottom; the right diagrams represent the sequence-level variational parameters, $\boldsymbol{\gamma}^{(B)}_{m,1:K}$, $\boldsymbol{\gamma}^{(A)}_{m,1:K}$

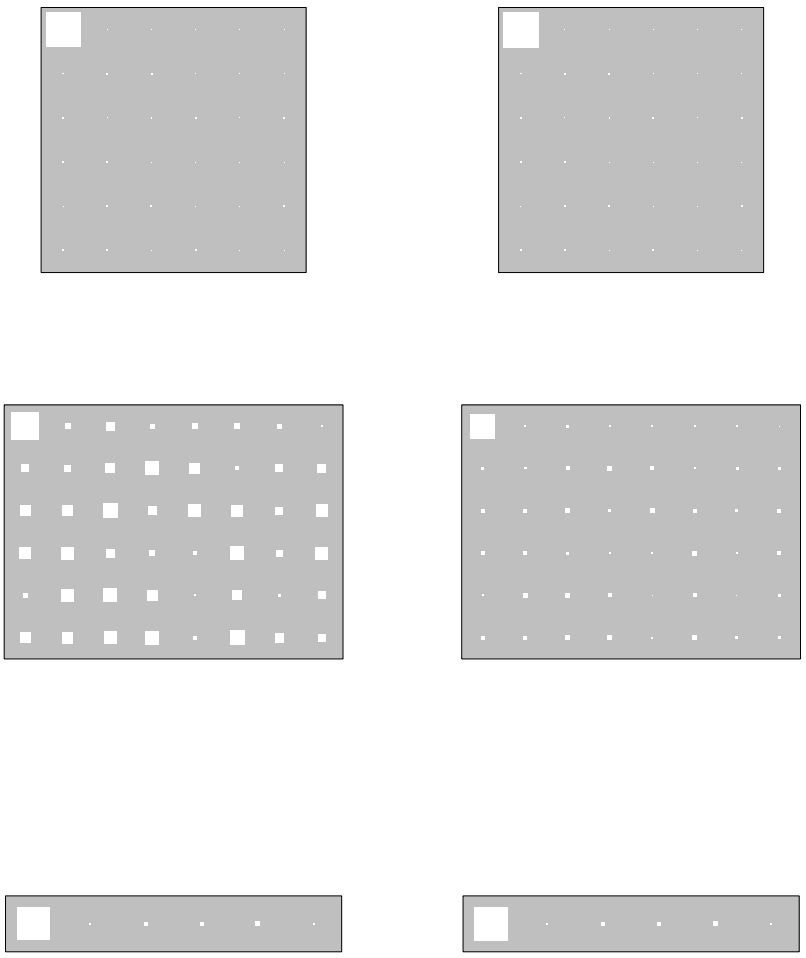


Figure 3.6: The Hinton Diagrams for (a) Database Level Parameters, (b) Sequence Level Variational Parameters.

and $\boldsymbol{\gamma}_m^{(\pi)}$ from the top to the bottom for a sample sequence from the data set. It is clear from the picture that the individual sequence displays slightly different characteristics from the whole database. Thus, it is important to model sequence-level characteristics individually.

3.4.2 Sequence Classification

Data Sets

This data set[6] consists of 3 classes of DNA sequences. One class is made up of 767 sequences belonging to the exon/intron boundaries, referred to as the EI class; another class of 768 sequences belongs to the intron/exon boundaries, referred to as the IE class; the third class of 1655 sequences does not belong to either of the above classes, referred to as the N class.

Evaluation Metrics

We conducted 3 binary classification experiments (i.e., EI vs IE, EI vs N and IE vs N) using 10-fold cross validation. For each class c, we learned a separate model $p(\mathbf{X}_{1:M_c}|c)$ of the sequences in that class, where M_c is the number of training sequences of class c. An unseen sequence was classified by picking $\arg\max_c p(\mathbf{X}|c)p(c)$. To eliminate the influence of $p(c)$, we varied its value and obtained the corresponding area under ROC curve (AUC) (Fawcett 2006), which is widely used for classification performance comparison.

Comparison of AUC on the Test Data Set

Table 3.3 reports the averaged results on the 10-fold validation and the best results for each number of hidden states are in bold. Surprisingly, our proposed LDHMMs do not significantly dominate other models. An possible explanation is that the generative models are not optimized for classification and thus more accurate modeling does not result in the significant improvement of classification performance. This problem may be alleviated by combining the model with a discriminative classifier. However, LDHMMs have very competitive performance compared to the best models in all cases. In addition, To further validate the statistical significance of our experiments,

[6]Available at http://archive.ics.uci.edu/ml/datasets/Molecular+Biology+%28Splice-junction+Gene+Sequences%29

we also perform the paired t-test (2-tail) between LDHMMs and other models on the experimental results. All the t-test results are less than 0.01, which proves the differences of LDHMMs versus other models are statistically significant.

3.5 Summary

Statistical modeling of sequential data has been studied for many years in machine learning and data mining. In this chapter, we propose LDHMMs to comprehensively characterize/model a database of sequential behaviors. Rather than assuming all the sequences share the same parameters as in traditional models, such as HMMs and VBHMMs, we explicitly assign sequence-level parameters to each sequence and database-level hyper-parameters to the whole database. The experimental results show that our model outperforms the other state-of-the-art models in predicting unseen sequential behaviors from web browsing logs and is competitive in classifying the unseen biological sequences. To sum up, the strength of LDHMMs is their comprehensively modeling of many sequences while the weakness of LDHMMs are their relatively high computational cost.

Table 3.3: The Experimental Results of the Real Data Sets

Dataset	Q	**LDA**	**HMM**	**HMMV**	**VBHMM**	**LDHMM-ff**	**LDHMM-nf**
EI vs IE	2	0.829 ± 0.032	$\mathbf{0.838 \pm 0.037}$	0.829 ± 0.034	0.829 ± 0.034	0.828 ± 0.035	0.834 ± 0.031
	3	$\mathbf{0.829 \pm 0.032}$	0.827 ± 0.046	$\mathbf{0.829 \pm 0.034}$	$\mathbf{0.829 \pm 0.034}$	0.828 ± 0.035	0.791 ± 0.044
	4	$\mathbf{0.829 \pm 0.032}$	0.81 ± 0.048	$\mathbf{0.829 \pm 0.034}$	$\mathbf{0.829 \pm 0.034}$	0.828 ± 0.035	0.803 ± 0.048
EI vs N	2	0.67 ± 0.029	0.679 ± 0.038	0.67 ± 0.029	0.67 ± 0.029	0.677 ± 0.03	$\mathbf{0.689 \pm 0.03}$
	3	0.667 ± 0.026	0.649 ± 0.033	0.669 ± 0.029	0.67 ± 0.029	$\mathbf{0.677 \pm 0.03}$	0.671 ± 0.021
	4	0.671 ± 0.025	0.659 ± 0.036	0.67 ± 0.029	0.67 ± 0.029	0.677 ± 0.03	$\mathbf{0.678 \pm 0.03}$
IE vs N	2	0.724 ± 0.036	0.739 ± 0.029	0.724 ± 0.036	0.724 ± 0.036	0.734 ± 0.033	$\mathbf{0.743 \pm 0.028}$
	3	0.725 ± 0.034	0.66 ± 0.024	0.724 ± 0.036	0.724 ± 0.036	$\mathbf{0.734 \pm 0.033}$	0.733 ± 0.032
	4	0.729 ± 0.033	0.721 ± 0.023	0.723 ± 0.036	0.724 ± 0.036	$\mathbf{0.734 \pm 0.033}$	0.725 ± 0.039

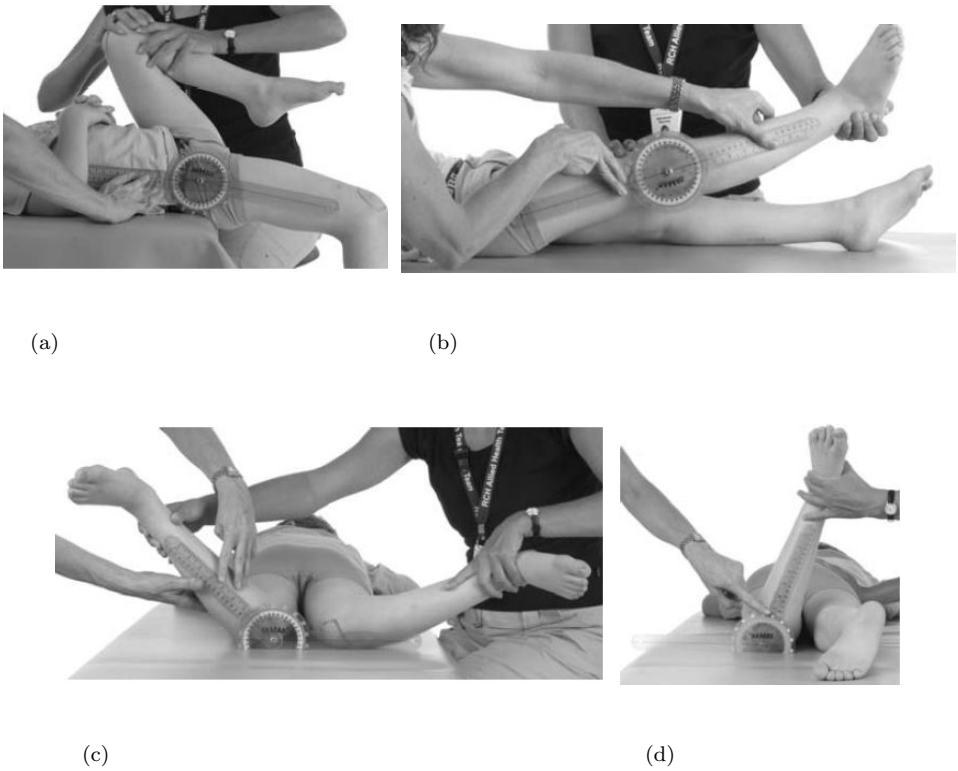
Chapter 4

The Correlated Static-dynamic Model

In clinical gait analysis (CGA), gait experts try to use patients' physical examination results, known as *static* data, to interpret the dynamic characteristics in an abnormal gait, known as *dynamic* data. From the data perspective, the above data has mixed structures and is thus heterogeneous. This chapter proposes a new probabilistic correlated static-dynamic model (CSDM) to model this kind of mixed structured data, which may be helpful for facilitating the automation of the gait analysis process and forming a relatively objective diagnosis. We propose an EM-based algorithm to learn the parameters of the CSDM. One of the main advantages of the CSDM is its ability to provide intuitive knowledge. For example, the CSDM can describe what kinds of static data will lead to what kinds of hidden gait patterns in the form of a decision tree, which helps us to infer dynamic characteristics based on static data. Our initial experiments indicate that the CSDM is promising for discovering the correlated relationship between physical examination (static) and gait (dynamic) data.

4.1 Introduction

'Gait' is a person's manner of walking. Patients may have an abnormal gait due to a range of physical impairment or brain damage. Clinical gait analysis (CGA) is a technique for identifying the underlying impairments that affect a patient's gait pattern. The CGA is critical for treatment planning. This process is carried out by gait analysis experts, mainly based on their experience which may lead to subjective diagnoses. The past 20 years have witnessed a burgeoning interest in clinical gait analysis for children with cerebral palsy (CP). The aim of clinical gait analysis is to determine a patient's impairments to plan manageable treatment. Usually, two types of data are used in clinical gait analysis: *static* data, which is the physical examination data that is measured when the patient is not walking, such as the shape of the femur and the strength of the abductor muscles. Figure 4.2 shows some examples of the process of the physical examination, which produces the static data shown in Table 4.1. From this excerpted data set, we can see that there are many attributes for the static data. The other type of data is *dynamic* data, which records dynamic characteristics that evolve during a gait trial. Usually, as shown in Figure 4.2, 3D Gait Analysis systems are applied to capture the movement and forces through individual joints, such as the hip, knee and ankle, when patients are walking or running. To achieve this, a set of reflective markers are placed on the interested joints of the patient and are tracked by the system. Those movement and forces in the interested joints can usually be displayed in curves. Figure 4.3 shows gait curve examples for one subject. Gait curves are recorded from multiple dimensions (i.e., from different parts of the body), such as the pelvis and hips. Since each subject has multiple trials, there are multiple curves for each dimension. In addition, each dimension has both the left and right side of the body. Thus, the total number of curves for each dimension is the number of trials multiplied by two. We use the red line to denote the dynamic of the left side and the blue line to denote the counterpart of the right side. Figure 1(a)-(d) show 4 different dimensions of the dynamics. Each curve in each dimension represents



(a) (b)

(c) (d)

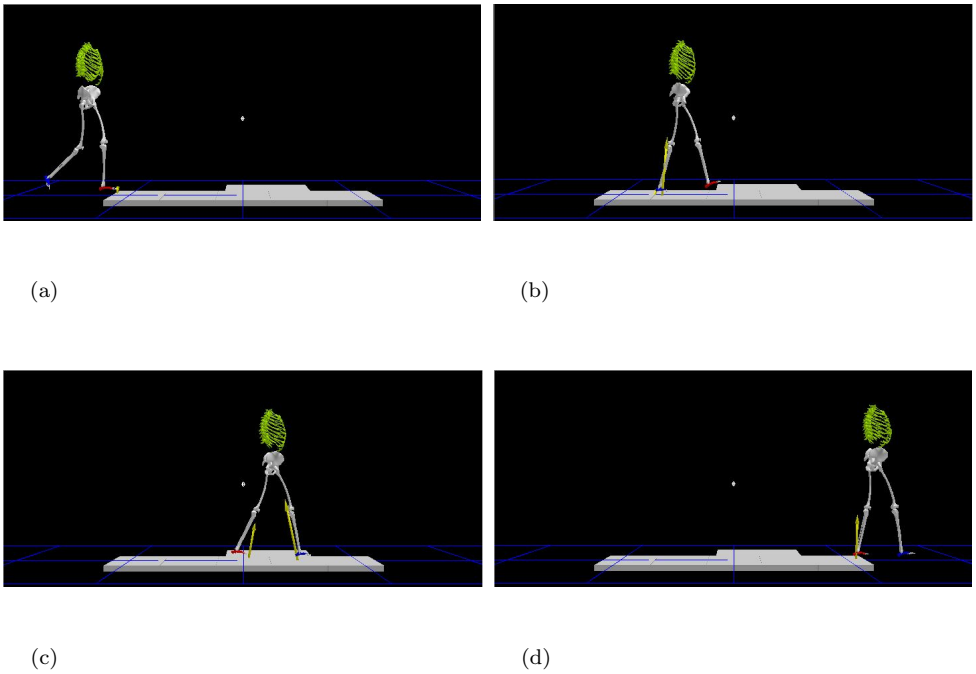
Figure 4.1: The Physical Examination Process: (a) Psoas - Thomas Test; (b) FFD Knee - Knee Extension; (c) Hip - Internal Rotation; (d) Hip - External Rotation.

the corresponding dynamics of one trial for the left or right part. The grey shaded area termed as *normal* describes the dynamic curve obtained from healthy people with a range of +/- 1 standard deviations for each observation point. From the example data shown above, we can see that describing the relationship between the *static* and *dynamic* data in the clinical gait data is not intuitive.

In practice, static data is used to explain abnormal features in dynamic data. In other words, gait analysis experts try to discover hidden relationships between *static* and *dynamic* variables for further clinical diagnosis. This process has been conducted empirically by clinical experts and thus is

Table 4.1: An Excerpt Data Set from the Static Data

Subject	Internal_Rotation_r	Internal_Rotation_l	Anteversion_r	···	Knee_Flexors_l
1	58	63	25	···	3+
2	60	71	15	···	4
3	53	52	29	···	3
⋮	⋮	⋮	⋮	⋮	⋮

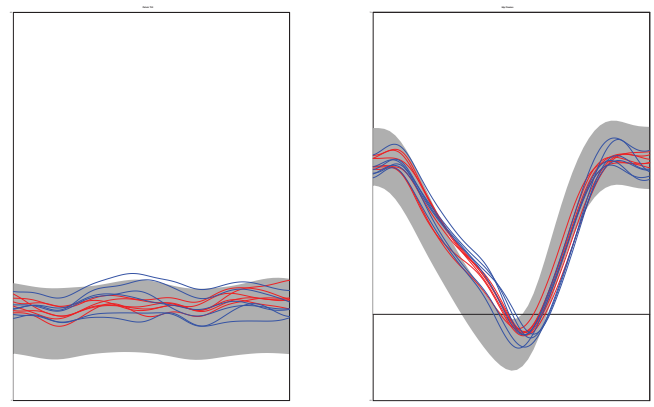


(a) (b) (c) (d)

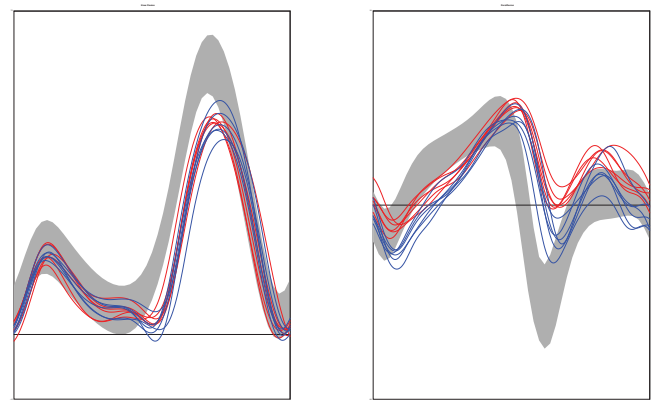
Figure 4.2: The 3D Gait Analysis System and chart (a) - (d) is in temporal order.

qualitative. In this chapter, we make an initial exploration to discover the *quantitative* correlated relationships between the *static* data and *dynamic* curves.

The rest of the chapter is organize as following: Section 4.2 presents the problem formalization. Then, Section 4.3 proposes a probabilistic graphical model to simulate the data generating process and gives an EM-based recipe for learning the model given the training data. Experimental results on both synthetic and real-world data sets are reported in Section 4.4 and Section 4.5



(a) (b)



(c) (d)

Figure 4.3: Example Gait Curves for One Patient with 6 Trials: (a) The Pelvic Tilt Dimension; (b) The Hip Flexion Dimension; (c) The Knee Flexion Dimension; (d) The Dorsiflexion Dimension.

summarizes this chapter.

4.2 Problem Statement

The following terms are defined:

- A *static profile* is a collection of static physical examination features of one subject denoted by $\mathbf{y} = (y_1, y_2, \cdots, y_L)$, where the subscript i $(1 \leq i \leq L)$ denotes the i^{th} attribute of the physical examination features, e.g., the Internal_Rotation_r attribute in Table 4.1.

- A *gait profile* is a collection of M gait trials made by one subject denoted by $\mathbf{X}_{1:M} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_M\}$.

- A *gait trial* (cycle) is multivariate time series denoted by $\mathbf{X}_m = (\mathbf{x}_{m1}, \mathbf{x}_{m2}, \cdots, \mathbf{x}_{mN})$, where $\mathbf{x}_{mj}$ $(1 \leq m \leq M$ and $1 \leq j \leq N)$ is the j^{th} vector observation of the time series and $\mathbf{x}_{mj} = \begin{bmatrix} x_{m1j} & x_{m2j} & \cdots & x_{mDj} \end{bmatrix}^T$ (D is the number of the dimensions for dynamic data and N is the length of the time series). For example, one dimension of the multivariate time series $(x_{mj1}, x_{mj2}, \cdots, x_{mjN})$ $(1 \leq j \leq D)$ can be plotted as one curve in Figure 4.3(a) and represents the dynamics of that dimension for one trial. $\mathbf{X}_m$ can be seen as a collection of such curves in different dimensions.

Our goal was to develop a probabilistic model $p(\mathbf{X}_{1:M}, \mathbf{y})$ that considers the correlated relationships between the *static profile* (i.e., static data) and the corresponding *gait profile* (i.e., dynamic data). In other words, we aim to produce a probabilistic model that assigns high probability to 'similar' data.

4.3 Proposed Model

4.3.1 Motivation

The basic idea is to construct the data generating process based on the domain knowledge gained by gait experts and model the process. Specifically, *static profile* $\mathbf{y}$ of a subject determines the generation of that subject's potential gait pattern. We denote this hidden gait pattern as a latent variable $\mathbf{h}$, a vector whose elements h_g $(1 \leq g \leq G)$[1] are 0 or 1 and sum to 1, where G is the number of hidden gait patterns. The generation of the corresponding *gait profile* $\mathbf{X}_{1:M}$ is then determined by this latent variable $\mathbf{h}$. In other words, the gait pattern is characterized by a distribution on the gait data. Due to the high dimensionality of $p(\mathbf{X}_{1:M}|\mathbf{h})$, the generating process of it is not intuitive. Thus, we need to consider the corresponding physical process. According to (Perry & Davids 1992), a gait trial can usually be divided into a number of phases[2] and each vector observation $\mathbf{x}_{mj}$ belongs to a certain state indicating its phase stage. These states are usually not labeled and we thus introduce latent variables $\mathbf{z}_{mj}$ $(1 \leq m \leq M,\ 1 \leq j \leq N_m)$ for each vector observation $\mathbf{x}_{mj}$ in each gait trial $\mathbf{X}_m$. We thus have two advantages: firstly, $p(\mathbf{X}_{1:M}|\mathbf{h})$ can be decomposed into a set of conditional probability distributions (CPDs) whose forms are intuitive to obtain; secondly, the dynamic process of the gait trials are captured by utilizing the domain knowledge (Bishop 2006).

4.3.2 The Correlated Static-Dynamic Model

We propose a novel correlated static-dynamic model (CSDM), which models the above conjectured data generating process. As mentioned before, existing models (e.g., HMMs and CRFs), cannot be directly used here. This is because HMMs only model the dynamic data $p(\mathbf{X}_m)$ and CRFs only model the relationship between $\mathbf{X}_m$ and $\mathbf{z}_m$, i.e., $p(\mathbf{z}_m|\mathbf{X}_m)$ $(1 \leq m \leq M)$, which is different to our goal of jointly modeling the *static* and *gait* profiles $p(\mathbf{X}_{1:M}, \mathbf{y})$.

[1] $h_g = 1$ denotes the g^{th} hidden gait pattern.

[2] Please refer to Appendix B.1 for the detailed description of the phases.

The graphical model for the CSDM is shown in Figure 4.4 (subscript m is omitted for convenience). We use conventional notation to represent the graphical model (Bishop 2006). In Figure 4.4, each node represents a random variable (or group of random variables). For instance, a *static profile* is represented as a node $\mathbf{y}$. The directed links express probabilistic causal relationships between these variables. For example, the arrow from the *static profile* $\mathbf{y}$ to the hidden gait pattern variable $\mathbf{h}$ indicates their causal relationships. For multiple variables that are of the same kind, we draw a single representative node and then surround this with a plate, labeled with a number indicating that there are many such kinds of nodes. An example can be found in Figure 4.4 in which M trials $\mathbf{Z}_{1:M}, \mathbf{X}_{1:M}$ are indicated by a plate label with M. Finally, we denote observed variables by shading the corresponding nodes and the observed *static profile* $\mathbf{y}$ is shown as shaded node in Figure 4.4. To further illustrate the domain knowledge-driven data generating process in Figure 4.4, the generative process for a *static profile* $\mathbf{y}$ to generate a *gait profile* $\mathbf{X}_{1:M}$ is described as follows:

1. Generate the static profile $\mathbf{y}$ by $p(\mathbf{y})$
2. Generate the latent gait pattern $\mathbf{h}$ by $p(\mathbf{h}|\mathbf{y})$
3. For each of the M trials
   (a) Generate the initial phase state $\mathbf{z}_{m1}$ from $p(\mathbf{z}_{m1}|\mathbf{h})$
   (b) Generate the corresponding gait observation $\mathbf{x}_{m1}$ by $p(\mathbf{x}_{m1}|\mathbf{z}_{m1}, \mathbf{h})$
   (c) For each of the gait observations $\mathbf{x}_{mn}$ $(2 \leq n \leq N)$
      i. Generate the phase state $\mathbf{z}_{mn}$ from $p(\mathbf{z}_{mn}|\mathbf{z}_{m,n-1}, \mathbf{h})$
      ii. Generate the the corresponding gait observation $\mathbf{x}_{mn}$ from $p(\mathbf{x}_{mn}|\mathbf{z}_{mn}, \mathbf{h})$

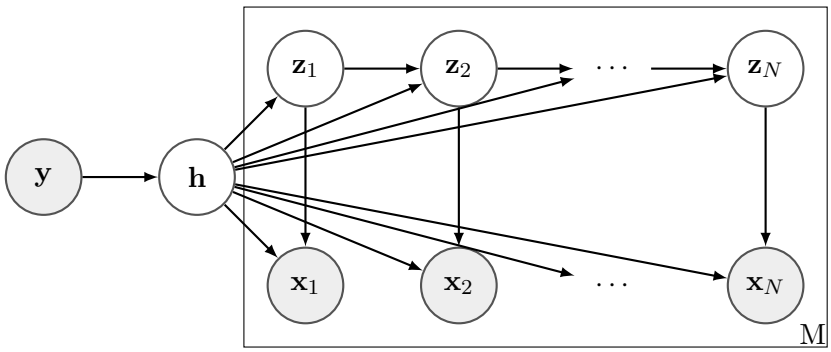


Figure 4.4: The Graphical Model of the CSDM

4.3.3 The Parameters of the CSDM

The parameters (i.e., the variables after the semicolon of each CPD) governing the CPDs of the CSDM are listed in the following[3]:

$$p(\mathbf{h}|\mathbf{y};\mathbf{d}) = \prod_{g=1}^{G} d_g(\mathbf{y})^{h_g} \tag{4.1}$$

where d_g $(1 \leq g \leq G)$ is a set of mapping functions $(\mathbf{y} \rightarrow d_g(\mathbf{y}) \equiv p(h_g = 1|\mathbf{y}))$ and h_g, d_g are the g^{th} element of $\mathbf{h}$, $\mathbf{d}$, respectively. Since the input $\mathbf{y}$ consists of discrete and continuous values, it is not intuitive to assume the format of the functions. Thus, here we use the form of a probability estimation tree (PET) (Provost & Domingos 2003) to represent the CPD $p(\mathbf{h}|\mathbf{y};\mathbf{d})$. To be more specific, the parameters governing the CPD is similar to the form "if $\mathbf{y}$ in some value ranges, then the probability of $h_g = 1$ is $d_g(\mathbf{y})$".

$$p(\mathbf{z}_{m1}|h;\boldsymbol{\pi}) = \prod_{g=1}^{G}\prod_{k=1}^{K} \pi_{gk}^{h_g,\ z_{m1k}} \tag{4.2}$$

where $\boldsymbol{\pi}$ is a matrix of probabilities with elements $\pi_{gk} \equiv p(z_{m1k} = 1|h_g = 1)$.

[3]We assume $p(\mathbf{y}) = const$ and the const is normalized and determined empirically from the data for convenience. Thus, we do not put it as a parameter.

$$p(\mathbf{z}_{mn}|\mathbf{z}_{m,n-1}, h; \mathbf{A}) = \prod_{g=1}^{G}\prod_{k=1}^{K}\prod_{j=1}^{K} a_{gjk}^{h_g,\ z_{m,n-1,j},\ z_{mnk}} \tag{4.3}$$

where $\mathbf{A}$ is a matrix of probabilities with elements $a_{gjk} \equiv p(z_{mnk} = 1|z_{m,n-1,j} = 1, h_g = 1)$.

$$p(\mathbf{x}_{ml}|\mathbf{z}_{ml}, h; \mathbf{\Phi}) = \prod_{g=1}^{G}\prod_{k=1}^{K} p(\mathbf{x}_{ml}|\boldsymbol{\phi}_{gk})^{h_g, z_{mlk}} \tag{4.4}$$

where $\mathbf{\Phi}$ is a matrix with elements $\boldsymbol{\phi}_{gk}$. For efficiency, in this chapter, we assume that $p(\mathbf{x}_{ml}; \boldsymbol{\phi}_{gk}) = \mathcal{N}(\mathbf{x}_{ml}; \boldsymbol{\mu}_{gk}, \boldsymbol{\sigma}_{gk})$, which is Gaussian distribution, and thus $\boldsymbol{\phi}_{gk} = (\boldsymbol{\mu}_{gk}, \boldsymbol{\sigma}_{gk})$.

Thus, the CSDM can be represented by the parameters $\boldsymbol{\theta} = \{\mathbf{d}, \boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\sigma}\}$.

4.3.4 Learning the CSDM

In this section we present the algorithm for learning the parameters of the CSDM, given a collection of *gait profiles* $\mathbf{X}_{s,1:M}$ and corresponding *static profiles* $\mathbf{y}_s$ $(1 \leq s \leq S)$ for different subjects. We assume each pair of gait and static profiles are independent of every others since they are from different subjects and share the same set of model parameters. Our goal is to find parameters $\boldsymbol{\theta}$ that maximize the log likelihood of the observed data $\mathbf{X}_{1:S,1:M}, \mathbf{y}_{1:S}$[4].

$$L(\boldsymbol{\theta}) = \sum_{s=1}^{S} \log p(\mathbf{X}_{s,1:M}|\mathbf{y}_s; \boldsymbol{\theta}) \tag{4.5}$$

Directly optimizing the above function with respect to $\boldsymbol{\theta}$ is very difficult because of the involvement of latent variables (Bishop 2006). We adopted an expectation-maximization (EM)-based algorithm (Dempster, Laird & Rubin 1977) to learn the parameters, yielding the iterative method presented in Algorithm 4.1. First, the parameters $\boldsymbol{\theta}^{old}$ need to be initialized. Then in the E step, $p(\mathbf{z}_{s,1:M}, \mathbf{h}_s|\mathbf{X}_{s,1:M}, \mathbf{y}_s, \boldsymbol{\theta}^{old})$ $(1 \leq s \leq S)$ is inferred given the

[4]We add the subscript s for representing the s^{th} profile in the rest of the paper.

Algorithm 4.1: The Learning Algorithm for the Proposed CSDM.

Input : An initial setting for the parameters $\boldsymbol{\theta}^{old}$

Output: Learned parameters $\boldsymbol{\theta}^{new}$

```
while the convergence criterion is not satisfied do
    Estep();
    θ^new = Mstep();
end
```

parameters $\boldsymbol{\theta}^{old}$ and will be used in M step. The M step then obtains the new parameters $\boldsymbol{\theta}^{new}$ that maximize the $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ function with respect to $\boldsymbol{\theta}$ as follows:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{s,\mathbf{h},\mathbf{z}} p(\mathbf{z}_{s,1:M}, \mathbf{h}_s | \mathbf{X}_{s,1:M}, \mathbf{y}_s; \theta^{old}) \log p(\mathbf{h}_s, \mathbf{z}_{s,1:M}, \mathbf{X}_{s,1:M}, \mathbf{y}_s; \boldsymbol{\theta}) \tag{4.6}$$

The E and M steps iterate until the convergence criterion is satisfied. In this manner, $L(\boldsymbol{\theta})$ is guaranteed to increase after each interaction.

Challenges of the Learning Algorithms

The challenges of the above algorithm is in the calculation of the E step and the M step. A standard forward-backward inference algorithm (Rabiner 1990) cannot be directly used here for the E step because of the introduction of latent variables $\mathbf{h}_s$ $(1 \leq s \leq S)$. We provided a modified forward-backward inference algorithm in Algorithm 4.2 considering the involvement of $\mathbf{h}_s$ $(1 \leq s \leq S)$. In calculating the M step, it was difficult to find an analytic solution for $\mathbf{d}(\cdot)$. We utilized a heuristic algorithm to solve it in Procedure estimatePET. The details of the implementation for E and M steps are discussed in the following.

The E step

Here we provide the detailed process of inferring the posterior distribution of the latent variables $\mathbf{h}_{1:S}, \mathbf{z}_{1:S,1:M}$ given the parameters of the model $\boldsymbol{\theta}^{old}$. Actually, we only infer some marginal posteriors instead of the joint posterior $p(\mathbf{z}_{s,1:M}, \mathbf{h}_s|\mathbf{X}_{s,1:M}, \mathbf{y}_s, \boldsymbol{\theta}^{old})$. This is because only these marginal posteriors will be used in the following M-step. We define the following notations for these marginal posteriors $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ and auxiliary variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ ($1 \leq s \leq S, 1 \leq m \leq M, 1 \leq n \leq N, 2 \leq n^{'} \leq N, 1 \leq j \leq K, 1 \leq k \leq K, 1 \leq g \leq G$):

$$\alpha_{sgmnk} = p(\mathbf{x}_{sm1}, \cdots, \mathbf{x}_{smn}, z_{smnk}|h_{sg}; \theta^{old}) \tag{4.7}$$

$$\beta_{sgmnk} = p(\mathbf{x}_{s,m,n+1}, \cdots, \mathbf{x}_{smN}|z_{smnk}, h_{sg}; \theta^{old}) \tag{4.8}$$

$$\gamma_{sgmnk} = p(z_{smnk}, h_{sg}|\mathbf{X}_{sm}, \mathbf{y}_s; \theta^{old}) \tag{4.9}$$

$$\xi_{s,g,m,n^{'}-1,j,n^{'},k} = p(z_{s,m,n^{'}-1,j}, z_{smn^{'}k}|h_{sg}, \mathbf{X}_{sm}, \mathbf{y}_s; \theta^{old}) \tag{4.10}$$

The inference algorithm is presented in Algorithm 4.2. Specifically, line 1 calls Procedure forward to calculate the forward variables $\boldsymbol{\alpha}$, while line 2 calls Procedure backward to calculate the backward variables $\boldsymbol{\beta}$. Then line3-15 calculate the value of each element of the posteriors $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ and the h_s^* ($1 \leq s$) on the basis of the $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\theta}^{old}$. These posteriors will be used in the M-step for updating the parameters.

The M step

Here we provide the detailed process for M step. Basically, it updates the parameters by maximizing the $Q(\theta, \theta^{old})$ with respect to them. If substituting the distributions with inferred marginal posteriors in the Q function, we can

Procedure forward

input : A set of the parameters $\boldsymbol{\theta}$

output: The variables $\boldsymbol{\alpha}$

// Initialization;

$\alpha_{sgm1k} = \pi_{gk}\mathcal{N}(\mathbf{x}_{sm1}; \boldsymbol{\mu}_{gk}, \boldsymbol{\sigma}_{gk})$ for all s, g, m and k;

for *s=1* **to** S **do** // Induction
  for *g=1* **to** G **do**
    for *m=1* **to** M **do**
      for *n=1* **to** *N-1* **do**
        for *k=1* **to** K **do**
          $\alpha_{s,g,m,n+1,k} = \sum_{j=1}^{K} \alpha_{sgmnj} a_{gjk} \mathcal{N}(\mathbf{x}_{s,m,n+1}; \boldsymbol{\mu}_{gk}, \boldsymbol{\sigma}_{gk})$;
        end
      end
    end
  end
end

obtain

$$
\begin{aligned}
Q(\theta, \theta^{old}) \;=\; & \sum_{s,\mathbf{h},\mathbf{z}_{s,1:M}} p(\mathbf{z}_{s,1:M}, \mathbf{h} | \mathbf{X}_{s,1:M}, \mathbf{y}_s; \boldsymbol{\theta}^{old}) \sum_{g=1}^{G} h_{sg} \log d_g(\mathbf{y}) \\
& + \sum_{s,g,m,k} \gamma_{sgm1k} \log \pi_{gk} \\
& + \sum_{s,g,m,j,k} \sum_{n=2}^{N} \xi_{s,g,m,n-1,j,n,k} \log a_{gjk} \\
& + \sum_{s,g,m,n,k} \gamma_{sgmnk} \log \mathcal{N}(\mathbf{x}_{smn}; \boldsymbol{\mu}_{gk}, \boldsymbol{\sigma}_{gk}) \qquad (4.11)
\end{aligned}
$$

Then the update formula for parameters $\mathbf{d}, \boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\sigma}$ can be obtained by maximizing the Q with respect to them, respectively:

- Updating of $\mathbf{d}$: Maximizing Q with respect to $\mathbf{d}$ is equivalent to maximizing the first item of Equation 4.11. However, $\mathbf{y}$ is a mixture of

Procedure backward

input : A set of the parameters $\boldsymbol{\theta}$

output: The variables $\boldsymbol{\beta}$

// Initialization;

$\beta_{sgmNk} = 1$ for all s, g, m and k;

for *s=1* **to** S **do** // Induction
  for *g=1* **to** G **do**
    for *m=1* **to** M **do**
      for *n=N-1* **to** *1* **do**
        for *j=1* **to** K **do**
          $\beta_{sgmnk} = \sum_{j=1}^{K} a_{gjk} \mathcal{N}(\mathbf{x}_{s,m,n+1}; \boldsymbol{\mu}_{gk}, \boldsymbol{\sigma}_{gk}) \beta_{s,g,m,n+1,j}$;
        end
      end
    end
  end
end

discrete and continuous values and it is impractical to find an analytic solution to $\mathbf{d}$. Here we consider a heuristic solution through the formation of probability estimation trees (PETs), which is a decision tree (Olshen & Stone 1984) with a Laplace estimation (Provost & Fawcett 2001) of the probability on class memberships (Provost & Domingos 2003). The heuristic algorithm for estimating the PET is described in Procedure estimatePET.

- Updating of $\boldsymbol{\pi}$, $\mathbf{A}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$: Maximize Q with respect to $\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\sigma}$ is easily achieved using appropriate Lagrange multipliers, respectively. The results are as follows:

$$\pi_{gk} = \frac{\sum_{s,m,g} \gamma_{sgm1k}}{\sum_{s,m,k,g} \gamma_{sgm1k}} \tag{4.12}$$

Algorithm 4.2: Estep()

input : An initial setting for the parameters $\boldsymbol{\theta}^{old}$

output: Inferred posterior distributions $\boldsymbol{\gamma}$, $\boldsymbol{\xi}$ and h_s^* $(1 \leq s \leq S)$

`/* Calculation of` $\boldsymbol{\alpha}$`,` $\boldsymbol{\beta}$ `*/`

Call Procedure forward using $\boldsymbol{\theta}^{old}$ as input;

Call Procedure backward using $\boldsymbol{\theta}^{old}$ as input;

`/* Calculation of` $\boldsymbol{\gamma}$`,` $\boldsymbol{\xi}$ `and` h_s^* $(1 \leq s \leq S)$ `*/`

for *s=1* **to** S **do**

  for *g=1* **to** G **do**

    for *m=1* **to** M **do**

      $p(\mathbf{X}_{sm}|h_{sg};\boldsymbol{\theta}^{old}) = \sum_{k=1}^{K} \alpha_{sgmNk}$;

      for *n=1* **to** N **do**

        $\gamma_{sgmnk} = \frac{\alpha_{sgmnk}\beta_{sgmnk}}{p(\mathbf{X}_{sm}|h_{sg};\boldsymbol{\theta}^{old})}$;

        $\xi_{s,g,m,n-1,j,n,k} = \frac{\alpha_{s,g,m,n-1,k}\mathcal{N}(\mathbf{x}_{smn'};\boldsymbol{\mu}_{gk},\boldsymbol{\sigma}_{gk})a_{gjk}\beta_{sgmnk}}{p(\mathbf{X}_{sm}|h_{sg};\boldsymbol{\theta}^{old})}$ $(n > 2)$;

      end

    end

  end

  $p(h_{sg}|\mathbf{y}_s;\boldsymbol{\theta}^{old}) = \prod_{m=1}^{M} p(\mathbf{X}_{sm}|h_{sg};\boldsymbol{\theta}^{old})$

  $p(h_{sg}|\mathbf{X}_{s,1:M},\mathbf{y}_s;\boldsymbol{\theta}^{old}) = \frac{p(h_{sg}|\mathbf{y}_s;\boldsymbol{\theta}^{old})p(h_{sg}|\mathbf{y}_s;\boldsymbol{\theta}^{old})}{\sum_{g=1}^{G} p(h_{sg}|\mathbf{y}_s;\boldsymbol{\theta}^{old})p(h_{sg}|\mathbf{y}_s;\boldsymbol{\theta}^{old})}$;

  $h_s^* = \arg\max_g p(h_{sg}|\mathbf{X}_{s,1:M},\mathbf{y}_s;\boldsymbol{\theta}^{old})$;

end

$$a_{gjk} = \frac{\sum_{s,m,n,g} \xi_{s,g,m,n-1,j,n,k}}{\sum_{s,m,l,n,g} \xi_{s,g,m,n-1,j,n,l}} \tag{4.13}$$

$$\boldsymbol{\mu}_{gk} = \frac{\sum_{s,m,g,n} \gamma_{sgmnk}\mathbf{x}_{smn}}{\sum_{s,m,n,g} \gamma_{sgmnk}} \tag{4.14}$$

Procedure estimatePET

input : The data tuple $(\mathbf{y}_s, h_s^*)$ $(1 \leq s \leq S)$

output: The learned PET $\mathbf{d}$

```
while stopping rule is not satisfactory do
    Examine all possible binary splits on every attribute of y_s
    (1 ≤ s ≤ S);
    Select a split with best optimization criterion;
    Impose the split on the PET d;
    Repeat recursively for the two child nodes;
end
for node in the PET d(·) do
    Do Laplace correction on each node;
end
```

Algorithm 4.3: Mstep()

input : Inferred posterior distributions $\boldsymbol{\gamma}$, $\boldsymbol{\xi}$ and h_s^* $(1 \leq s \leq S)$

output: The updated parameters θ^{new}

1 Call Procedure estimatePET to update $\mathbf{d}(\cdot)$;

2 Update $\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}_{gk}, \boldsymbol{\sigma}_{gk}$ according to Equation 4.12-4.15;

$$\boldsymbol{\sigma}_{gk} = \frac{\sum\limits_{s,m,g,n} \gamma_{sgmnk}(\mathbf{x}_{smn} - \mu_{gk})(\mathbf{x}_{smn} - \mu_{gk})^T}{\sum\limits_{s,m,n,g} \gamma_{sgmnk}} \tag{4.15}$$

Algorithm 4.3 summarizes the whole process of the M step.

4.4 Empirical Study

The aim of this study is to test:

- The feasibility of the learning algorithm for the CSDM. Since we have

proposed an iterative (i.e., EM-based) learning method, it is pivotal to show its convergence on the gait data set.

- The predictability of the CSDM. The aim of the CSDM is to discover the correlated relationship between the static and dynamic data. Thus, it is interesting to validate its predictive power on other data falling outside the scope of the training data set.

- The usability of the CSDM. Because the CSDM is designed to be used by gait experts, we need to demonstrate intuitive knowledge extracted by the CSDM.

4.4.1 Experimental Settings

We sampled the synthetic data from the true parameters listed in Table 4.2. We varied the s_0 for different sample sizes (e.g., $s_0 = 100, 500, 1500$) to represent relatively small, medium and large data sets. The real-world data set we used was provided by the gait lab at the Royal Children's Hospital, Melbourne[5]. We have collected a subset of static and dynamic data for 99 patients. The static data subset consisted of 8 attributes summarized in Table 4.3. There were at most 6 gait trials for each subject and each gait trial had 101 vector observations. In principle, curves for both left and right sides may be included. However, for simplicity and consistency, we only used the right side curves of the hip rotation dimension for analysis in this pilot study. In addition, for a given patient, all records for that patient were either in the test set or in the training set. By doing so, it was expected to avoid the information leak from the training data set to the test data set.

[5]http://www.rch.org.au/gait/

Table 4.2: The Parameters for the Synthetic Data

d	if $-50 \leq y < -25$, $p(h_1 = 1\|y) = 1$, if $-25 \leq y < 0$, $p(h_2 = 1\|y) = 1$, if $0 \leq y < 25$, $p(h_1 = 1\|y) = 1$, if $25 \leq y < 50$, $p(h_2 = 1\|y) = 1$.
$\boldsymbol{\pi}$	$\pi_{1,1:2} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$ $\pi_{2,1:2} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$
A	$a_{1,1:2,1:2} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$ $a_{2,1:2,1:2} = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$
$\boldsymbol{\mu}$	$\mu_{1,1:2,1} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$ $\mu_{2,1:2,1} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$
$\boldsymbol{\sigma}$	$\sigma_{1,1:2,1} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ $\sigma_{2,1:2,1} = \begin{bmatrix} 1 & 1 \end{bmatrix}$

Table 4.3: Description of the Static Data

Name of Attributes	Data Type	Value Range
internalrotation_r (ir_r)	continuous	23 to 90
internalrotation_l (ir_l)	continuous	20 to 94
externalrotation_r (er_r)	continuous	-5 to 57
externalrotation_l (er_l)	continuous	-26 to 51
anteversion_r (a_r)	continuous	10 to 50
anteversion_l (a_l)	continuous	4 to 45
hipabductors_r (h_r)	discrete	-1 to 5
hipabductors_l (h_l)	discrete	-1 to 5

4.4.2 Experimental Results

Convergence of the Learning Process

For each iteration, we calculate the averaged log-likelihood as

$$\frac{1}{S}\sum_{s=1}^{S}\sum_{m=1}^{M}\log p(\mathbf{X}_{sm}, \mathbf{y}_s; \boldsymbol{\theta}^{old}) \tag{4.16}$$

where $\boldsymbol{\theta}^{old}$ is the parameters updated from last iteration. Figure 4.5(a) shows the CSDM against the iteration numbers for different sample sizes of the synthetic data and Figure 4.5(b) shows the results of the averaged log-likelihoods for CSDMs using different numbers (represented as G) of hidden gait patterns. As expected, the averaged log-likelihood is not monotonic all the time, since part of the learning process uses a heuristic algorithm. However, the best averaged log-likelihoods are usually achieved after at most 5 iterations, which proves the convergence of the proposed learning algorithm. It can be seen from Figure 4.5(a), a larger sample size will lead to a higher log-likelihood for the learning algorithm. For the real-world data set, $G = 4$[6] shows the fastest convergence rate of the three settings for CSDMs.

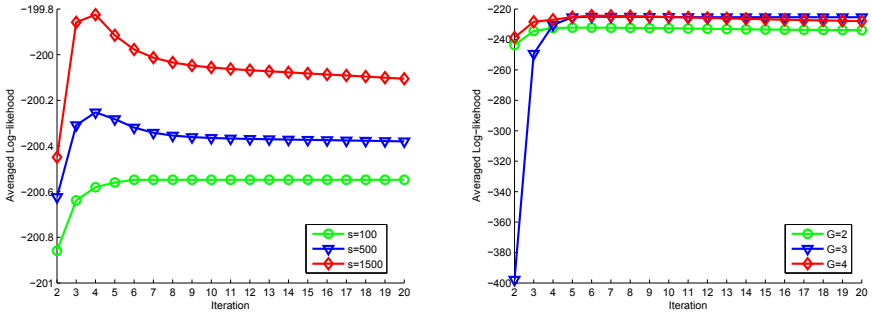
Predictive Performance

We measured the CSDM predictive accuracy in terms of how well the future gait profile can be predicted given the static profile and learned parameters. Since the final prediction is a set of complex variables, we measure the predictive log-likelihood $\sum_{s'=1}^{S'}\log p(\mathbf{X}_{s',1:M}|\mathbf{y}_{s'};\boldsymbol{\theta})$ in the testing data with S' static and gait profiles, where $\boldsymbol{\theta}$ is learned from the training data. Then, the following can be obtained by using Bayes rule:

$$\log p(\mathbf{X}_{s',1:M}|\mathbf{y}_{s'};\boldsymbol{\theta}) = \log(\sum_{g} p(h_{s'g}|\mathbf{y}_{s'};\boldsymbol{\theta})p(\mathbf{X}_{s',1:M}|h_{s'g};\boldsymbol{\theta})) \tag{4.17}$$

where $p(h_{s'g}|\mathbf{y}_{s'};\boldsymbol{\theta})$ and $p(\mathbf{X}_{s',1:M}|h_{s'g};\boldsymbol{\theta})$ can be calculated by using the line 13 and 14 of Algorithm 4.2 (i.e., E step).

[6]The number of G is suggested by gait experts not exceeding 4.



(a) The Synthetic Data Sets

(b) The Real-world Data Set

Figure 4.5: Log-likelihood for the CSDM against the iteration numbers for different numbers of hidden gait pattern G.

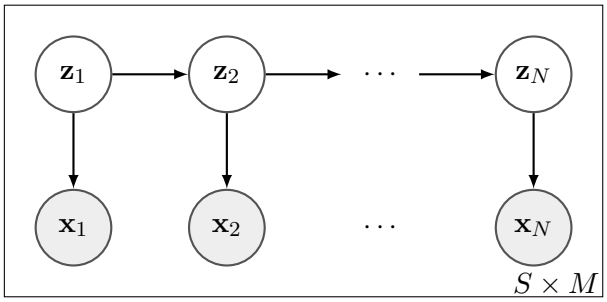


Figure 4.6: The Graphical Model for the Baseline Algorithm.

Without loss of generality, we proposed a baseline algorithm which ignored the static data for modeling and prediction to compare with our proposed method. The baseline model is a standard HMM with multiple observation sequences, whose graphical model is depicted in Figure 4.6. It assumes all the gait trials are independently generated from an HMM. Using the standard algorithm provided in (Baum et al. 1970, Rabiner 1990), we can learn the parameters of the baseline model, denoted as $\boldsymbol{\theta}_0$ from the training data. Accordingly, the predictive averaged log-likelihood for new gait trials can be calculated as $\sum_{s'=1}^{S'} \log p(\mathbf{X}_{s',1:M}; \boldsymbol{\theta}_0)$.

Table 4.4: The Comparison of the Log-likelihoods

	$s_0 = 100$	$s_0 = 500$	$s_0 = 1500$
CSDM	**-8016**	**-40090**	**-120310**
Baseline	-8025	-40132	-120420

(a) The Synthetic Data

	$G = 2$	$G = 3$	$G = 4$
CSDM	**-1310**	**-1388**	**-1299**
Baseline	-1426	-1502	-1426

(b) The Real-world Data

We compare the CSDM with the alternating baseline scheme, an HMM with multiple sequences. We report on averages over 10 times 5-fold cross validations for the synthetic and real-world data, respectively. As shown in Table 4.5(a), all the CSDMs outperformed the baseline algorithm significantly. This may be because the proposed CSDM captures the correlated relationships existing in the data rather than ignoring them. Similarly, it can be observed from Table 4.5(b) that all the CSDMs achieved higher log-likelihoods than their counterparts of the baseline model. This proves the predictive power of our proposed CSDM on real-world data.

Extracting Knowledge from the CSDM

In this section, we provide an illustrative example of extracting intuitive knowledge from a CSDM on the gait data. Our real-world data are described in Section 4.4.1. We used the EM algorithm described in Section 4.3.4 to find the model parameters for a 4-hidden-gait-pattern CSDM as suggested by gait experts. Given the learned CSDM, we can extract the intuitive knowledge

from the data set to answer the following questions:

- What kinds of static data will lead to what kinds of hidden gait patterns?
- What does the gait look like for each hidden gait pattern?

The first question is actually asking what is $p(\mathbf{h}|\mathbf{y};\boldsymbol{\theta})$ (and subscript s is omitted since all s share the same parameters). Figure 4.7 shows an answer to the first question in the form of a decision tree representation. This tree[7] decides hidden gait patterns based on the 8 features of the static data (e.g., ir_r, er_r and a_r) used in the data set. To decide the hidden gait patterns based on the static data, start at the top node, represented by a triangle ($\triangle$). The first decision is whether ir_r is smaller than 57. If so, follow the left branch, and see that the tree classifies the data as gait pattern 2. If, however, anteversion exceeds 57, then follow the right branch to the lower-right triangle node. Here the tree asks whether er_r is is smaller than 21.5. If so, then follow the right branch to see the question of next node until the tree classifies the data as ones of the gait patterns. For other nodes, the gait patterns can be decided in similar manners.

The second question is actually asking $\arg\max_g p(h_{sg}|\mathbf{X}_{s,1:M}, \mathbf{y}_s; \boldsymbol{\theta})$ ($1 \leq s \leq S$). In other words, we need to infer which gait trials belong to the corresponding hidden gait patterns in the corpus. We use line 14 described in Algorithm 4.2 to obtain the hidden gait pattern names of the gait trials. We can then plot representative gaits for each hidden gait pattern to answer the second question above, as shown in Figures 4.8(a)-4.8(d). Figure 4.8(d) shows a collection of gaits for the hidden gait pattern 4. We can see that most of them fall into the normal area, which may indicate that these gaits are good. Figure 4.8(b) shows a collection of gaits for the hidden gait pattern 2 and most of them are a little below the normal area, indicating that these gaits are not as good. By contrast, most of the gaits in Figure 4.8(a) representing

[7]For simplicity, we only display the gait pattern with the highest probability. The tree shown in Figure 4.7 is partial and the fully tree is available at Appendix B.2.

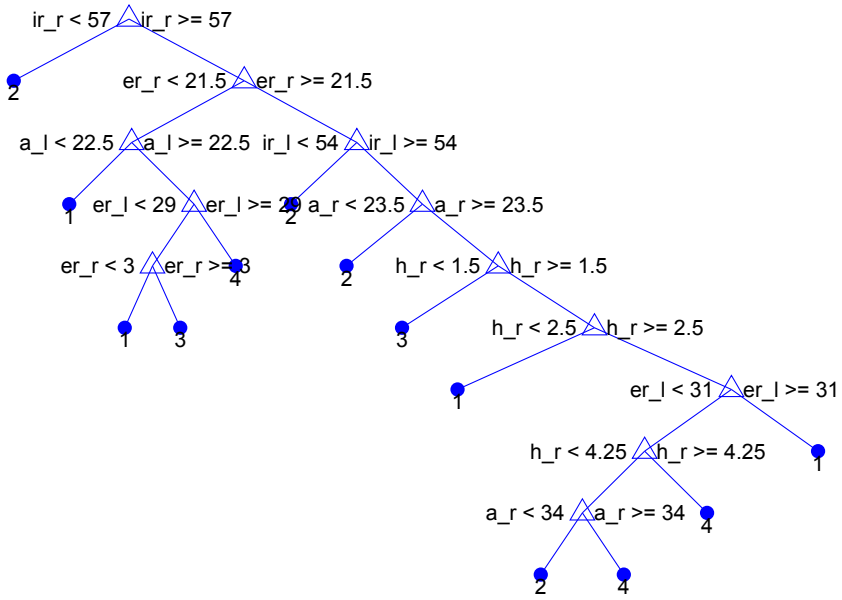


Figure 4.7: The Decision Tree to Predict Gait Patterns Given the Static Data

hidden gait pattern 1 fall outside the normal area and are abnormal gaits. Figure 4.8(c) shows that the representative gaits for hidden gait pattern 3 are slightly above the normal area, which indicates these gaits are only slightly abnormal. Most subjects displaying pattern 1 and some subjects displaying pattern 3 would be susceptible to have surgery. By extracting the different paths that lead to those two patterns from the decision tree in Figure 4.7, we can infer what combinations of static data may have clinical implications.

4.5 Summary

This chapter presents a new probabilistic graphical model (i.e., CSDM) for quantitatively discovering the correlated relationship between static physical examination data and dynamic gait data in clinical gait analysis. To learn the parameters of the CSDM on a training data set, we proposed an EM-

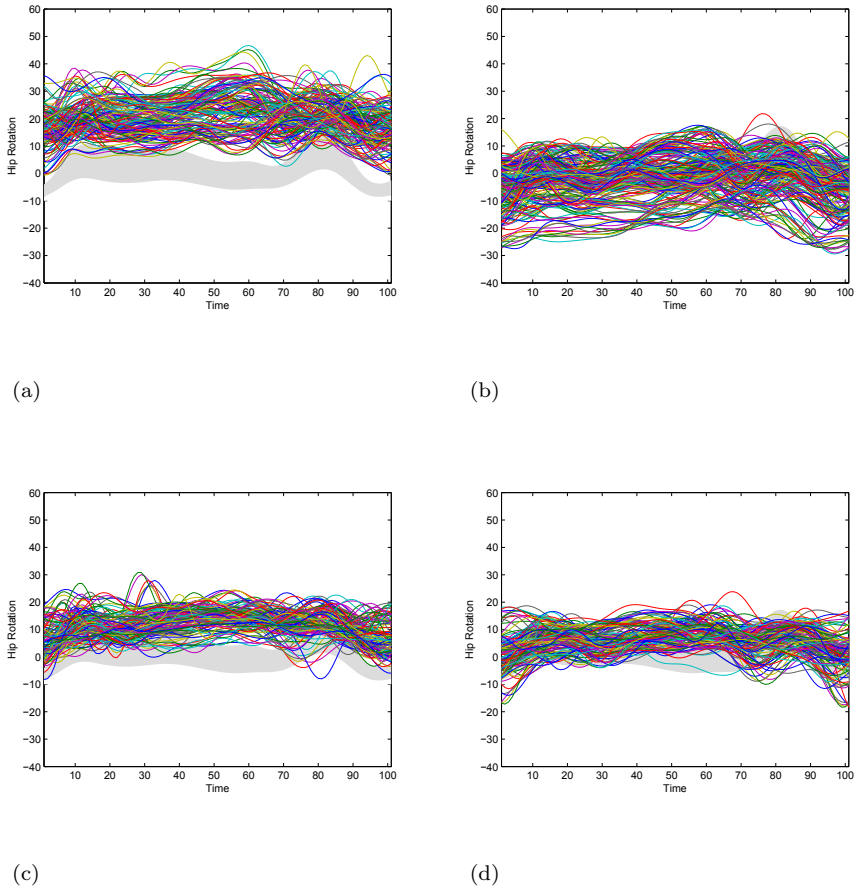


Figure 4.8: Representative Gaits for Gait Pattern 1-4.

based algorithm. One of the main advantages of the CSDM is its ability to provide intuitive knowledge. For example, the CSDM informs us what kinds of static data will lead to what kinds of hidden gait patterns and what the gaits look like for each hidden gait pattern. The experiments on both synthetic and real-world data (excerpted from patient records at the Royal Children's Hospital, Melbourne) showed promising results in terms of learning convergence, predictive performance and knowledge discovery.

Chapter 5

The Joint Interest-social Model

With the development of web 2.0 sites, the user/user friendship networks have become available as well as the user/item preference matrix. The above data is inherently heterogeneous since it consists of two different data sources. In order to jointly model the user/user networks and the user/item matrix, this chapter proposes a probabilistic model, the joint interest-social model (JISM). To be more specific, we represent each user with a latent *interest* variable and a latent *social* variable, and each item with a latent *interest* variable. The interactions are then determined by these latent variables. Variational Bayes is employed to infer these latent variables and Variational EM is applied to estimate the parameters of the model. We then use the learned model to predict the missing ratings. The experimental results on real-world data sets shows the proposed model outperforms other state-of-the-art methods.

5.1 Introduction

As a subclass of information filtering system, recommendation systems typically provide each user with the *ratings* or *preference* that the user would give to a list of items (such as music, books, or movies). Research on such systems has been increasingly popular in both commercial and research community

for the last decade. In academic communities, they have attracted many researchers ranging from information retrieval (Herlocker, Konstan, Borchers & Riedl 1999, Sarwar, Karypis, Konstan & Riedl 2001, Hofmann 2003), machine learning (Mnih & Salakhutdinov 2007, Salakhutdinov & Mnih 2008, Rennie & Srebro 2005) to data mining (Koren et al. 2009, Koren 2010). Meanwhile, E-commerce websites, such as the DVD rental provider Netflix[1] and the online book retailer Amazon[2] have deployed their own recommendation systems (Shani & Gunawardana 2011). Collaborative filtering (CF) is the traditional approach to item recommendation and usually have better performance than other methods (Sarwar et al. 2001, Su & Khoshgoftaar 2009), such as neighborhood methods (Herlocker et al. 1999, Koren et al. 2009). CF techniques typically are based on *latent factor models* (Mnih & Salakhutdinov 2007, Salakhutdinov & Mnih 2008, Koren et al. 2009) and use the user-item rating matrix (i.e., *preference matrix*) to predict the missing values. Specifically, items are recommended to a user based on other users with similar patterns of selected items[3].

Recently, web 2.0 web sites, such as Douban[4], Last.fm[5] and Fixster[6], allow users to create their own friendship networks online as well as share the ratings/preferences on items. This provides a new data source for analysis and how to use the additional social networks data for recommendation systems raises a new challenge. As mentioned before, popular CF-based recommendation systems only utilize the *preference matrix* for rating/preference prediction and ignore the social interactions among users. These social relations, however, may reveal users' preference as well and potentially improve the accuracy of the item recommendation.

The item recommendation issue in the aforementioned socialized environ-

[1] www.Netflix.com

[2] www.amazon.com

[3] Please note that CF does not use the content of the items and utilizing the content of items falls out the scope of this chapter.

[4] www.douban.com

[5] www.last.fm

[6] www.flixster.com

ment, which terms as *social recommendation* (Ma et al. 2008, Xin, King, Deng & Lyu 2009, Ma et al. 2011), has drawn a few attempts (Ma et al. 2008, Yang et al. 2011). Almost of all of them are based on the *Homophily* assumption (McPherson, Smith-Lovin & Cook 2001) that any pair of friends share the same interest. Specifically, they use homogeneous *interest* latent factor space to represent users and items. The user/item ratings (i.e., the elements of the *preference matrix*) are determined by the *interest* latent factors of the corresponding user/item pairs; the user/user interactions (i.e., the edges of the *friendship networks*) are similarly determined by the *interest* latent factors of the corresponding user/user pairs.

However, as it is very intuitive to see the latent factor space of users are heterogeneous. This is because there are not only *interest* latent factors but also additional latent factors, such as *social* latent factors (e.g., working in the same company), influence the user/user and user/item interactions. For example, the user/item ratings may have social-bias, i.e., some community may favor some type of items than others, which indicates these ratings are not only determined by the *interest* latent factors. The situation is similar in the user/user interactions. For instance, two users have friendship may not only because of they have social connections, such as going to the same school, but also because of the same interest thanks to the online web service. Thus, on the basis of the above understanding, we can conclude that the user/user and user/item interactions are not independent and can be connected by the underlying latent factors and these interactions need to be jointly modeled for predicting missing ratings.

With the above thoughts in mind, we develop a probabilistic graphical model for jointly modeling the *rating matrix* and *friendship networks* in an unified model, which can be further used to predict missing ratings (i.e., recommendation). Different to most of current research that assumes one homogeneous *interest* latent factor space, our model represents users with heterogeneous latent factors, i.e., *interest* and *social* factors. These latent factors are governed by some parameters and determine the generation of user/item

and user/user interactions. We develop a variational EM (Bishop 2006) algorithm to learn parameters and approximately infer the posterior distribution of the latent factors given the parameters. The inferred posteriors of latent factors are then used to predict missing ratings.

We tested our algorithm on three real-world data sets, Lastfm, Douban and Flixster, which are crawled from three popular web 2.0 web sites. From the perspective of data size, these three data sets represent relatively small, medium and large sample sizes. Our proposed model generally has better performance on all of the data sets when withholding some of each user's ratings for prediction. Another advantage of our algorithm is its ability to predict a user's ratings with only his/her friendship networks, which traditional CF-based methods fail to provide. The experimental results prove the feasibility of our algorithm on predicting one user's ratings without any ratings for training in a socialized environment.

5.2 Problem Statement

5.2.1 An Illustrative Example

Here we recall the toy example described in Figure 1.1. The problem we study here is to predict the missing ratings (those represented by '?') of the user/item matrix with additional *friendship* networks. In terms of available ratings, the problem can be divided into two types of tasks: *in-matrix* prediction and *out-of-matrix prediction*. In-matrix prediction refers to the task of making rating prediction for those users that at least have one rating on items. As shown in Figure 1.1(b), predicting the missing ratings of Sophia, Harry and Emma is the task of *in-matrix* prediction. This is the task that traditional CF-based techniques focus on. Out-of-matrix prediction refers to the task of making rating prediction for those users have no available ratings. This is one type of cold start problems (Shani & Gunawardana 2011). As shown in Figure 1.1(b), predicting the missing ratings of Jack and Oliver is the task of *out-of-matrix* prediction. Traditional CF-based algorithms cannot

make generalized predictions for the task of *out-of-matrix* prediction. This is due to the usage of the user/item rating matrix alone. The preference of the users who have no rating cannot be inferred. However, the utilization of the additional *friendship* networks sheds new light on the task of the *out-of-matrix* prediction, which is one of advantages of our proposed model compared to the other CF-based algorithms.

5.2.2 Problem Formalization

The social recommendation problem we study can be formally described below:

Definition 1 *Suppose there are N users and M items, given a rating matrix $\mathbf{R}$ whose element r_{ij} is user i's rating on item j, friendship networks $\mathbf{E}$ whose element $e_{ii'}$ indicates the friendship of user i and user $i^{'}$, and given some known elements of $\mathbf{R}$, how to infer the values of the remaining elements of $\mathbf{R}$? This task can be further divided as following:*

- *In-matrix prediction: For user i and at least one element of $\mathbf{r}_{i,1:M}$ is known, predicting the other missing values of user i.*
- *Out-of-matrix prediction: For user i and no element of $\mathbf{r}_{i,1:M}$ is known, predicting all the missing values of user i.*

5.3 The Joint Interest-social Model (JISM)

Our model represents users with heterogeneous latent factors, i.e., *interest* and *social* factors, and items with *interest* latent factor. These latent factors are conditionally i.i.d (Blei et al. 2003) and governed by the same parameters of the whole data set. The *interest* and *social* latent factors of users, the *interest* latent factors of items and the social bias parameter jointly determine the generation of user/item ratings. Similarly, the *interest* and *social* latent factors of users and link formation parameters control the generating of user/user interactions. Formally, the generative process is as following:

1. Generate parameters $\mathcal{P}$.
2. For each user i $(1 \leq i \leq N)$,
   (a) Generate $\mathbf{u}_i \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$.
   (b) Generate $\mathbf{w}_i \sim \mathcal{N}(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$
3. For each item j, generate $\mathbf{v}_j \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ $(1 \leq j \leq M)$.
4. For each non-missing entry (i, j) in $\mathbf{R}$, generate $r_{ij} \sim \mathcal{N}(\mathbf{u}_i^T \mathbf{v}_j + \mathbf{w}_i^T \mathbf{B} \mathbf{v}_j, \tau)$ $(1 \leq i \leq N \text{ and } 1 \leq j \leq M)$.
5. For each non-missing entry $(i, i^{'})$ in $\mathbf{E}$, generate $e_{ii'} \sim \text{Bernoulli}(\sigma(t_1 \mathbf{u}_i^T \mathbf{u}_{i'} + t_2 \mathbf{w}_i^T \mathbf{w}_{i'} + t_3))$ $(1 \leq i, i^{'} \leq N \text{ and } i \neq i^{'})$, where σ is sigmoid function, i.e. $\sigma(x) = 1/(1 + exp(-x))$.

For quick reference, the notations used in the JISM is listed in Table 5.1[7]. For visualization of the proposed model, we plot the graphical model of the JISM in Figure 5.1 and We use conventional notation to represent the graphical model (Bishop 2006). To be more specific, each open circle represents a random variable (or group of random variables) and smaller solid circles denote deterministic parameters, and the directed links express probabilistic causal relationships between these variables. For multiple variables that are of the same kind, we draw a single representative node and then surround this with a plate, labeled with a number indicating that there are many such kinds of nodes. Finally, we denote observed variables by shading the corresponding open circles.

5.4 Learning and Prediction

5.4.1 Variational EM Learning

The goal of learning is to estimate the model parameters $\mathcal{P} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_3, \mathbf{B}, \boldsymbol{\mu}_\tau, \mathbf{t}\}$. Direct maximum likelihood estima-

[7] $\boldsymbol{\xi}$, $\mathbf{E}$ and $\mathbf{R}$ are sparse.

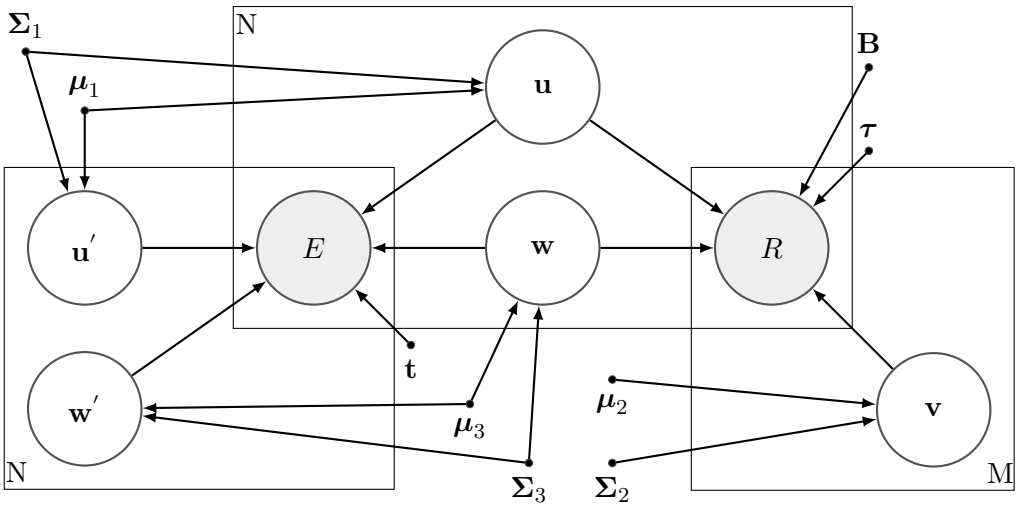


Figure 5.1: The Graphical Model of the JISM.

tion is very difficult since involvement of latent variables. Thus, we apply the variational EM framework (Bishop 2006), which consists of E and M steps and iterates between them, to learn the parameters. Specifically, the E step, known as variational inference (Jordan et al. 1999), fixes the parameters and approximately infer the posteriors of the latent variables; while the M step fixed the approximate posteriors of the latent variables and estimate the parameters. This framework, however, cannot be applied directly here since the JISM is non-conjugate and it is difficult to directly derives an analytic objective function. Thus, in the following, Section 6.3.1 first derives an analytic lower bound of the log-likelihood as the objective function, Section 5.4.1 and Section Estep then describe the implementation of the variational EM algorithm based on the derived objective function.

The Objective Function

Lemma 5.4.1.1 *The Log-likelihood of the observed data is bounded by the following inequality:*

$$\begin{aligned}\log p(R, E|\mathcal{P}) \geq & E_q[\log p(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}, \mathbf{w}_{1:N}, R, E|\mathcal{P})] - E_q[q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}, \mathbf{w}_{1:N}|\mathcal{P}_\prime)] \\ = & L_0\end{aligned} \tag{5.1}$$

where q is the variational approximation to $p(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}, \mathbf{w}_{1:N}|\mathcal{P}, R, E)$, given by

$$\begin{aligned}q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}, \mathbf{w}_{1:N}|\mathcal{P}, R, E) = & \prod_{i=1}^{N} q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, \text{diag}(\boldsymbol{\nu}_{1i}^2))q(\mathbf{w}_i|\boldsymbol{\lambda}_{3i}, \text{diag}(\boldsymbol{\nu}_{3i}^2)) \\ & \prod_{j=1}^{M} q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j}, \text{diag}(\boldsymbol{\nu}_{2j}^2))\end{aligned} \tag{5.2}$$

$\mathcal{P}' = \{\boldsymbol{\lambda}_{1i}, \text{diag}(\boldsymbol{\nu}_{1i}^2), \boldsymbol{\lambda}_{2j}, \text{diag}(\boldsymbol{\nu}_{2j}^2), \boldsymbol{\lambda}_{3i}, \text{diag}(\boldsymbol{\nu}_{3i}^2)\}$ is variational parameters of the Gaussian posterior distributions.

Then, we have

Corollary 5.4.1.1 *The Lower bound L_0 can be expanded as:*

$$\begin{aligned}& E_q[\sum_{i=1}^{N} \log p(\mathbf{u}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)] + E_q[\sum_{j=1}^{M} \log p(\mathbf{v}_j|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)] + E_q[\sum_{i=1}^{N} \log p(\mathbf{w}_i|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)] \\ & + E_q[\sum_{i=1}^{N}\sum_{j=1}^{M} \delta_{ij} \log p(r_{ij}|\mathbf{u}_i^T\mathbf{v}_j + \mathbf{w}_i^T\mathbf{B}\mathbf{v}_j, \boldsymbol{\tau})] \\ & + E_q[\sum_{i=1}^{N}\sum_{i'=1}^{N} \delta_{ii'} \log p(e_{ii'}|\mathbf{u}_i^T\mathbf{u}_{i'} + \mathbf{w}_i^T\mathbf{w}_{i'}, \mathbf{t})] \\ & - E_q[\sum_{i=1}^{N} \log q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, diag(\boldsymbol{\nu}_{1i}^2))] - E_q[\sum_{j=1}^{M} \log q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j}, diag(\boldsymbol{\nu}_{2j}^2))] \\ & - E_q[\sum_{i=1}^{N} \log q(\mathbf{w}_i|\boldsymbol{\lambda}_{3i}, diag(\boldsymbol{\nu}_{3i}^2))]\end{aligned} \tag{5.3}$$

Since we have the following lemmas:

Lemma 5.4.1.2 *The analytic expansion f_4 of the fourth item of L_0, i.e., $E_q[\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\log p(r_{ij}|\mathbf{u}_i^T\mathbf{v}_j+\mathbf{w}_i^T\mathbf{B}\mathbf{v}_j,\tau)]$, is:*

$$\begin{aligned}
f_4 =&(-\frac{K}{2}\ln 2\pi-\frac{1}{2}\ln|\tau^2|)\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\\
&-\frac{1}{2\tau^2}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}[r_{ij}^2-2r_{ij}(\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j})\\
&+tr(diag(\boldsymbol{\nu}_{1i}^2)diag(\boldsymbol{\nu}_{2j}^2))+\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}\\
&+\boldsymbol{\lambda}_{2j}^T diag(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i}\\
&+tr(diag(\boldsymbol{\nu}_{3i}^2)\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T)+\boldsymbol{\lambda}_{3i}^T\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T\boldsymbol{\lambda}_{3i}\\
&+\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T diag(\boldsymbol{\nu}_{3i}^2)\mathbf{B}\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T\boldsymbol{\lambda}_{3i}\\
&+\boldsymbol{\lambda}_{3i}^T\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i}\\
&+\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T\boldsymbol{\lambda}_{3i}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T\boldsymbol{\lambda}_{3i}]
\end{aligned}\tag{5.4}$$

Lemma 5.4.1.3 *The analytic lower bounder f_5 of the fifth item of L_0, i.e., $E_q[\sum_{i=1}^{N}\sum_{i'=1}^{N}\delta_{ii'}\log p(e_{ii'}|\mathbf{u}_i^T\mathbf{u}_{i'}+\mathbf{w}_i^T\mathbf{w}_{i'},\mathbf{t})]$, is:*

$$f_5=\sum_{i=1}^{N}\sum_{i'=i+1}^{N}\delta_{ii'}[\log\sigma(\xi_{ii'})+\frac{f_{51}-\xi_{ii'}}{2}+g(\xi_{ii'})f_{52}-g(\xi_{ii'})\xi_{ii'}^2]\tag{5.5}$$

where $f_{51}=t_1\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}+t_2\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}+t_3$ and $f_{52}=t_1^2(tr(\mathrm{diag}(\boldsymbol{\nu}_{1i}^2)\mathrm{diag}(\boldsymbol{\nu}_{1i'}^2))+\boldsymbol{\lambda}_{1i}^T\mathrm{diag}(\boldsymbol{\nu}_{1i'}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{1i'}^T\mathrm{diag}(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{1i'}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{1i'}^T\boldsymbol{\lambda}_{1i})+t_2^2(tr(\mathrm{diag}(\boldsymbol{\nu}_{3i}^2)\mathrm{diag}(\boldsymbol{\nu}_{3i'}^2))+\boldsymbol{\lambda}_{3i}^T\mathrm{diag}(\boldsymbol{\nu}_{3i'}^2)\boldsymbol{\lambda}_{3i}+\boldsymbol{\lambda}_{3i'}^T\mathrm{diag}(\boldsymbol{\nu}_{3i}^2)\boldsymbol{\lambda}_{3i'}+\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}\boldsymbol{\lambda}_{3i'}^T\boldsymbol{\lambda}_{3i})+t_3^2+2t_1t_2\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}+2t_1t_3\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}+2t_2t_3\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}$.

Then using the above lemmas we can obtain:

Theorem 5.4.1.1 *The analytic lower bound of L_0 is:*

$$L=\sum_{i=1}^{5}f_i-\sum_{j=1}^{3}f_j^{'}\tag{5.6}$$

where

$$\begin{aligned} f_1 =& E_q[\sum_{i=1}^{N} \log p(\mathbf{u}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)] \\ =& -\frac{KN}{2}\ln 2\pi - \frac{N}{2}\ln|\boldsymbol{\Sigma}_1| \\ & -\frac{1}{2}\sum_{i=1}^{N}[tr(\mathrm{diag}(\boldsymbol{\nu}_{1i}^2)\Sigma_1^{-1}) + (\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1)^T\Sigma_1^{-1}(\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1)] \end{aligned} \tag{5.7}$$

$$\begin{aligned} f_2 =& E_q[\sum_{j=1}^{M} \log p(\mathbf{v}_j|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)] \\ =& -\frac{KN}{2}\ln 2\pi - \frac{N}{2}\ln|\boldsymbol{\Sigma}_2| \\ & -\frac{1}{2}\sum_{j=1}^{M}[tr(\mathrm{diag}(\boldsymbol{\nu}_{2j}^2)\Sigma_2^{-1}) + (\boldsymbol{\lambda}_{2j}-\boldsymbol{\mu}_2)^T\Sigma_2^{-1}(\boldsymbol{\lambda}_{2j}-\boldsymbol{\mu}_2)] \end{aligned} \tag{5.8}$$

$$\begin{aligned} f_3 =& E_q[\sum_{i=1}^{N} \log p(\mathbf{w}_i|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)] \\ =& -\frac{KN}{2}\ln 2\pi - \frac{N}{2}\ln|\boldsymbol{\Sigma}_3| \\ & -\frac{1}{2}\sum_{i=1}^{N}[tr(\mathrm{diag}(\boldsymbol{\nu}_{3i}^2)\Sigma_3^{-1}) + (\boldsymbol{\lambda}_{3i}-\boldsymbol{\mu}_3)^T\Sigma_3^{-1}(\boldsymbol{\lambda}_{3i}-\boldsymbol{\mu}_3)] \end{aligned} \tag{5.9}$$

$$\begin{aligned} f_1^{'} =& E_q[\sum_{i=1}^{N} \log q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, \mathrm{diag}(\boldsymbol{\nu}_{1i}^2))] \\ =& -\frac{KN}{2}(\ln 2\pi + 1) - \frac{1}{2}\sum_{i=1}^{N}\log|\mathrm{diag}(\boldsymbol{\nu}_{1i}^2)| \end{aligned} \tag{5.10}$$

$$\begin{aligned} f_2^{'} =& E_q[\sum_{j=1}^{M} \log q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j}, \mathrm{diag}(\boldsymbol{\nu}_{2j}^2))] \\ =& -\frac{KM}{2}(\ln 2\pi + 1) - \frac{1}{2}\sum_{j=1}^{M}\log|\mathrm{diag}(\boldsymbol{\nu}_{2j}^2)| \end{aligned} \tag{5.11}$$

$$
\begin{aligned}
f_3^{'} =& E_q[\sum_{i=1}^{N} \log q(\mathbf{w}_i|\boldsymbol{\lambda}_{3i}, \text{diag}(\boldsymbol{\nu}_{3i}^2))] \\
=& -\frac{KN}{2}(\ln 2\pi + 1) - \frac{1}{2}\sum_{i=1}^{N} \log|\text{diag}(\boldsymbol{\nu}_{3i}^2)|
\end{aligned} \tag{5.12}
$$

f_4 and f_5 are shown in Equation 5.4 and 5.5.

E step: Variational Inference

Fix the parameters $\mathcal{P}$, we can update the variational parameters by optimizing the derived lower bound.

Proposition 5.4.1.1 *To optimize of the lower bound L, the updating formula for* $\boldsymbol{\lambda}_{1i}$, $\boldsymbol{\lambda}_{3i}$, $diag(\boldsymbol{\nu}_{1i}^2)$, $diag(\boldsymbol{\nu}_{3i}^2)$, $\boldsymbol{\xi}$, $\boldsymbol{\lambda}_{2j}$ *and* $diag(\boldsymbol{\nu}_{2j}^2)$ *are as follows:*

$$
\begin{aligned}
\boldsymbol{\lambda}_{1i}^T =& (\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1} + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(r_{ij}\boldsymbol{\lambda}_{2j}^T - \boldsymbol{\lambda}_{3i}^T\mathbf{B}(diag(\boldsymbol{\nu}_{2j}^2) + \boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T)) \\
&+ \sum_{i^{'}=1,\neq i}^{N}\delta_{ii^{'}}((\frac{1}{2}t_1 + 2g(\xi_{ii^{'}})t_1t_3 + 2g(\xi_{ii^{'}})t_1t_2\boldsymbol{\lambda}_{3i^{'}}^T\boldsymbol{\lambda}_{3i})\boldsymbol{\lambda}_{1i^{'}}^T)) \\
&(\boldsymbol{\Sigma}_1^{-1} + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T + diag(\boldsymbol{\nu}_{2j}^2)) \\
&- \sum_{i^{'}=1,\neq i}^{N}\delta_{ii^{'}}(2g(\xi_{ii^{'}})t_1^2(\boldsymbol{\lambda}_{1i^{'}}\boldsymbol{\lambda}_{1i^{'}}^T + diag(\boldsymbol{\nu}_{1i^{'}}^2))))^{-1}
\end{aligned} \tag{5.13}
$$

$$\begin{aligned}\boldsymbol{\lambda}_{3i}^T =&(\boldsymbol{\mu}_3^T\boldsymbol{\Sigma}_3^{-1} + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(r_{ij}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T - \boldsymbol{\lambda}_{1i}^T(diag(\boldsymbol{\nu}_{2j}^2) + \boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T)\mathbf{B}^T)\\ &+ \sum_{i'=1,\neq i}^{N}\delta_{ii'}((\frac{1}{2}t_2 + 2g(\xi_{ii'})t_2t_3 + 2g(\xi_{ii'})t_1t_2\boldsymbol{\lambda}_{1i'}^T\boldsymbol{\lambda}_{1i})\boldsymbol{\lambda}_{3i'}^T))\\ &(\boldsymbol{\Sigma}_3^{-1} + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T + \mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T)\\ &- \sum_{i'=1,\neq i}^{N}\delta_{ii'}(2g(\xi_{ii'})t_2^2(\boldsymbol{\lambda}_{3i'}\boldsymbol{\lambda}_{3i'}^T + diag(\boldsymbol{\nu}_{3i'}^2))))^{-1}\end{aligned} \tag{5.14}$$

$$\begin{aligned}diag(\boldsymbol{\nu}_{1i}^2) =&(\boldsymbol{\Sigma}_1^{-1} + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T + diag(\boldsymbol{\nu}_{2j}^2))\\ &- \sum_{i'=1,\neq i}^{N}\delta_{ii'}(2g(\xi_{ii'})t_1^2(\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{1i'}^T + diag(\boldsymbol{\nu}_{1i'}^2))))^{-1}\end{aligned} \tag{5.15}$$

$$\begin{aligned}diag(\boldsymbol{\nu}_{3i}^2) =&(\boldsymbol{\Sigma}_3^{-1} + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T + \mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T)\\ &- \sum_{i'=1,\neq i}^{N}\delta_{ii'}(2g(\xi_{ii'})t_2^2(\boldsymbol{\lambda}_{3i'}\boldsymbol{\lambda}_{3i'}^T + diag(\boldsymbol{\nu}_{3i'}^2))))^{-1}\end{aligned} \tag{5.16}$$

$$\xi_{ii'} = (f_{52})^{\frac{1}{2}} \tag{5.17}$$

$$\begin{aligned}\boldsymbol{\lambda}_{2j}^T =&(\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}_2^{-1} + \frac{1}{\tau^2}\sum_{i=1}^{N}\delta_{ij}r_{ij}(\boldsymbol{\lambda}_{1i}^T + \boldsymbol{\lambda}_{3i}^T\mathbf{B}))\\ &(\boldsymbol{\Sigma}_2^{-1} + \frac{1}{\tau^2}\sum_{i=1}^{N}\delta_{ij}(\boldsymbol{\lambda}_{1i}\boldsymbol{\lambda}_{1i}^T + diag(\boldsymbol{\nu}_{1i}^2) + \mathbf{B}^T\boldsymbol{\lambda}_{3i}\boldsymbol{\lambda}_{3i}^T\mathbf{B} + \mathbf{B}^T diag(\boldsymbol{\nu}_{3i}^2)\mathbf{B}\\ &+ \mathbf{B}^T\boldsymbol{\lambda}_{3i}\boldsymbol{\lambda}_{1i}^T + \boldsymbol{\lambda}_{1i}\boldsymbol{\lambda}_{3i}^T\mathbf{B}))^{-1}\end{aligned} \tag{5.18}$$

Procedure Estep($\mathcal{P}$, $\mathbf{E}$, $\mathbf{R}$)

input : The parameters $\mathcal{P}$, friendship network $\mathbf{E}$ and preference network $\mathbf{R}$.

output: The variational parameters $\mathcal{P}'$

Initialize $\boldsymbol{\lambda}_{1,i}$, $\boldsymbol{\lambda}_{3,i}$, $\boldsymbol{\nu}_{1,i}$ and $\boldsymbol{\nu}_{3,i}$ for all i;
Initialize $\boldsymbol{\lambda}_{2,j}$ and $\boldsymbol{\nu}_{2,j}$ for j;
Initialize ξ;
repeat
    Update $\boldsymbol{\lambda}_{1,i}$ according to Equation 5.13;
    Update $\boldsymbol{\lambda}_{3,i}$ according to Equation 5.14;
    Update $\boldsymbol{\nu}_{1,i}$ according to Equation 5.15;
    Update $\boldsymbol{\nu}_{3,i}$ according to Equation 5.16;
    Update $\boldsymbol{\xi}$ according to Equation 5.17;
    Update $\boldsymbol{\lambda}_{2,j}$ according to Equation 5.18;
    Update $\boldsymbol{\nu}_{2,j}$ according to Equation 5.19;
until *convergence*;
$\mathcal{P}' \leftarrow \boldsymbol{\lambda}_{1,1:N}$, $\boldsymbol{\lambda}_{3,1:N}$, $\boldsymbol{\nu}_{1,1:N}$, $\boldsymbol{\nu}_{3,1:N}$, $\boldsymbol{\lambda}_{2,1:M}$, $\boldsymbol{\nu}_{2,1:M}$, ξ;

$$
\begin{aligned}
diag(\boldsymbol{\nu}_{2j}^2) =& (\boldsymbol{\Sigma}_2^{-1} + \frac{1}{\tau^2}\sum_{i=1}^{N}\delta_{ij}(\boldsymbol{\lambda}_{1i}\boldsymbol{\lambda}_{1i}^T + diag(\boldsymbol{\nu}_{1i}^2) + \mathbf{B}^T\boldsymbol{\lambda}_{3i}\boldsymbol{\lambda}_{3i}^T\mathbf{B} \\
&+ \mathbf{B}^T diag(\boldsymbol{\nu}_{3i}^2)\mathbf{B} + \mathbf{B}^T\boldsymbol{\lambda}_{3i}\boldsymbol{\lambda}_{1i}^T + \boldsymbol{\lambda}_{1i}\boldsymbol{\lambda}_{3i}^T\mathbf{B}))^{-1}
\end{aligned}
\tag{5.19}
$$

M step: Parameter Estimation

Proposition 5.4.1.2 *To optimize of the lower bound L, the updating formula for $\boldsymbol{\mu}_{1:3}$, $\boldsymbol{\Sigma}_{1:3}$, $\mathbf{B}$, $\mathbf{t}$ and τ^2 is as follows:*

$$
\boldsymbol{\mu}_1 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\lambda}_{1i} \tag{5.20}
$$

$$
\boldsymbol{\mu}_2 = \frac{1}{N}\sum_{j=1}^{M}\boldsymbol{\lambda}_{2j} \tag{5.21}
$$

$$\boldsymbol{\mu}_3 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\lambda}_{3i} \tag{5.22}$$

$$\boldsymbol{\Sigma}_1 = \frac{1}{N}\sum_{i=1}^{N}[diag(\boldsymbol{\nu}_{1i}^2) + (\boldsymbol{\lambda}_{1i} - \boldsymbol{\mu}_1)(\boldsymbol{\lambda}_{1i} - \boldsymbol{\mu}_1)^T] \tag{5.23}$$

$$\boldsymbol{\Sigma}_2 = \frac{1}{M}\sum_{j=1}^{M}[diag(\boldsymbol{\nu}_{2j}^2) + (\boldsymbol{\lambda}_{2j} - \boldsymbol{\mu}_2)(\boldsymbol{\lambda}_{2j} - \boldsymbol{\mu}_2)^T] \tag{5.24}$$

$$\boldsymbol{\Sigma}_3 = \frac{1}{N}\sum_{i=1}^{N}[diag(\boldsymbol{\nu}_{3i}^2) + (\boldsymbol{\lambda}_{3i} - \boldsymbol{\mu}_3)(\boldsymbol{\lambda}_{3i} - \boldsymbol{\mu}_3)^T] \tag{5.25}$$

$$\begin{aligned}\mathbf{B}^T =&(\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}(diag(\boldsymbol{\nu}_{2j}^2) + \boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T))^{-1}\\ &(\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}(\boldsymbol{\lambda}_{2j}r_{ij}\boldsymbol{\lambda}_{3i}^T - ((diag(\boldsymbol{\nu}_{2j}^2) + \boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T)\boldsymbol{\lambda}_{1i}\boldsymbol{\lambda}_{3i}^T)))\\ &(\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}(diag(\boldsymbol{\nu}_{3i}^2) + \boldsymbol{\lambda}_{3i}\boldsymbol{\lambda}_{3i}^T))^{-1}\end{aligned} \tag{5.26}$$

$$\begin{aligned}t_1 =& -(\sum_{i=1}^{N}\sum_{i'=i+1}^{N}\delta_{ii'}g(\xi_{ii'})(2t_2\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'} + 2t_3\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}))\\ &(\sum_{i=1}^{N}\sum_{i'=i+1}^{N}\delta_{ii'}(\frac{1}{2}\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'} + 2g(\xi_{ii'})(tr(diag(\boldsymbol{\nu}_{1i}^2)diag(\boldsymbol{\nu}_{1i'}^2))\\ &+ \boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{1i'}^2)\boldsymbol{\lambda}_{1i} + \boldsymbol{\lambda}_{1i'}^T diag(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{1i'} + \boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{1i'}^T\boldsymbol{\lambda}_{1i})))^{-1}\end{aligned} \tag{5.27}$$

$$\begin{aligned}t_2 =& -(\sum_{i=1}^{N}\sum_{i'=i+1}^{N}\delta_{ii'}g(\xi_{ii'})(2t_1\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'} + 2t_3\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}))\\ &(\sum_{i=1}^{N}\sum_{i'=i+1}^{N}\delta_{ii'}(\frac{1}{2}\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'} + 2g(\xi_{ii'})(tr(diag(\boldsymbol{\nu}_{3i}^2)diag(\boldsymbol{\nu}_{3i'}^2))\\ &+ \boldsymbol{\lambda}_{3i}^T diag(\boldsymbol{\nu}_{3i'}^2)\boldsymbol{\lambda}_{3i} + \boldsymbol{\lambda}_{3i'}^T diag(\boldsymbol{\nu}_{3i}^2)\boldsymbol{\lambda}_{3i'} + \boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}\boldsymbol{\lambda}_{3i'}^T\boldsymbol{\lambda}_{3i})))^{-1}\end{aligned} \tag{5.28}$$

Algorithm 5.1: The Learning Algorithm for the JISM.

Input : An initial setting for the parameters $\mathcal{P}$

Output: Learned parameters $\mathcal{P}^*$

```
while the convergence criterion is not satisfied do
    // E-step
    P' ← Estep(P, E, R);
    // M-step
    Update P according to Equation 5.20-5.30;
end
P* ← P;
```

$$
\begin{aligned}
t_3 = &-(\sum_{i=1}^{N}\sum_{i'=i+1}^{N}\delta_{ii'}\,g(\xi_{ii'})(2t_1\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}+2t_2\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}))\\
&(\sum_{i=1}^{N}\sum_{i'=i+1}^{N}\delta_{ii'}(\frac{1}{2}+2g(\xi_{ii'})))^{-1}
\end{aligned}
\tag{5.29}
$$

$$
\begin{aligned}
\tau^2 =&\frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}[r_{ij}^2-2r_{ij}(\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j})\\
&+tr(diag(\boldsymbol{\nu}_{1i}^2)diag(\boldsymbol{\nu}_{2j}^2))+\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}\\
&+\boldsymbol{\lambda}_{2j}^T diag(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i}\\
&+tr(diag(\boldsymbol{\nu}_{3i}^2)\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T)+\boldsymbol{\lambda}_{3i}^T\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T\boldsymbol{\lambda}_{3i}\\
&+\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T diag(\boldsymbol{\nu}_{3i}^2)\mathbf{B}\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T\boldsymbol{\lambda}_{3i}\\
&+\boldsymbol{\lambda}_{3i}^T\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i}\\
&+\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T\boldsymbol{\lambda}_{3i}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T\boldsymbol{\lambda}_{3i}]
\end{aligned}
\tag{5.30}
$$

Summary of the Learning Algorithm

For clarity, Algorithm 5.1 summarizes the process of learning the JISM.

5.4.2 Preference Prediction

In-matrix Prediction

For prediction of the $(i,j)^{th}$ entry r_{ij} in the test data set, we first calculate the predictive distribution $p(r_{ij}|\mathbf{R},\mathcal{P}^*)$ as following:

$$\begin{aligned}
p(r_{ij}|\mathbf{R},\mathcal{P}^*) &= \int\int\int p(r_{ij},\mathbf{u}_i,\mathbf{v}_j,\mathbf{w}_i|\mathbf{R},\mathcal{P}^*)d\mathbf{u}_i d\mathbf{v}_j d\mathbf{w}_i \\
&= \int\int\int p(\mathbf{u}_i,\mathbf{v}_j,\mathbf{w}_i|\mathbf{R},\mathcal{P}^*)p(r_{ij}|\mathbf{u}_i,\mathbf{v}_j,\mathbf{w}_i,\mathbf{R},\mathcal{P}^*)d\mathbf{u}_i d\mathbf{v}_j d\mathbf{w}_i \\
&= \int\int\int p(\mathbf{u}_i,\mathbf{v}_j,\mathbf{w}_i|\mathbf{R},\mathcal{P}^*)p(r_{ij}|\mathbf{u}_i,\mathbf{v}_j,\mathbf{w}_i)d\mathbf{u}_i d\mathbf{v}_j d\mathbf{w}_i \\
&\approx \int\int\int q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i})q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j})q(\mathbf{w}_i|\boldsymbol{\lambda}_{3i})p(r_{ij}|\mathbf{u}_i,\mathbf{v}_j,\mathbf{w}_i)d\mathbf{u}_i d\mathbf{v}_j d\mathbf{w}_i
\end{aligned}$$

Then we use the expectation of r_{ij} with respect to its posterior $p(r_{ij}|\mathbf{R},\mathcal{P}^*)$ as the prediction:

$$\begin{aligned}
\hat{r}_{ij} &= \int\int\int\int p(r_{ij}|\mathbf{R},\mathcal{P}^*)r_{ij}d\mathbf{u}_i d\mathbf{v}_j d\mathbf{w}_i dr_{ij} \\
&= \int\int\int q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i})q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j})q(\mathbf{w}_i|\boldsymbol{\lambda}_{3i})(\mathbf{u}_i^T\mathbf{v}_j + \mathbf{w}_i^T\mathbf{B}\mathbf{v}_j)d\mathbf{u}_i d\mathbf{v}_j d\mathbf{w}_i \\
&= \boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j} + \boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}
\end{aligned}$$

Out-of-matrix Prediction

The well-known cold start problem on new users (Shani & Gunawardana 2011). The preference prediction process is similar to in-matrix prediction, except that dropping the items with $\boldsymbol{\xi}$ and $\mathbf{t}$ of the updating formulas, Equation 5.13 and 5.17 and no updating $\boldsymbol{\lambda}_{2j}$, $\boldsymbol{\nu}_{2j}$ and $\boldsymbol{\xi}$, for the out-of-matrix friendship prediction.

Summary of the Prediction Algorithm

For clarity, Algorithm 5.2 summarizes the algorithm for missing ratings prediction by using the JISM.

Algorithm 5.2: The Prediction Algorithm for the JISM.

Input : Learned parameters $\mathcal{P}^*$

Output: Predicted Ratings $\mathbf{R}^{'}$

`// Variational Inference`

1 $\mathcal{P}^{'} \leftarrow$ `Estep(`$\mathcal{P}^*$, $\mathbf{E}$, $\mathbf{R}$`)`;

`// Prediction`

2 Predict the elements of $\mathbf{R}^{'}$ according to Equation 5.31;

5.4.3 Discussions

The computational complexity for the E-step is proportional to $O((K+Q+Q^2+K^2)(N_R+N_E))T_E$, where K is the number of hidden states, T_E is the iteration number of E-step, N_E is the number of non-missing values in $\mathbf{E}$ and N_R is the number of non-missing values in $\mathbf{R}$. Similarly, The computational complexity for the E-step is proportional to $O((K+K^2)(N+M)+(Q+Q^2)N$. An alternative methods for learning and inference in non-conjugate probabilistic model are numerical algorithms, such as Markov chain Monte Carlo (MCMC). However, in practical, they are relatively too slow to analyze large data sets (Gershman, Hoffman & Blei 2012).

5.5 Empirical Studies

In this section, we apply the proposed JISM in several real-world data sets. All algorithms were implemented in matlab[8] and performed on a 2.9GHz 20MB L3 Cache Intel Xeon E5-2690 (8 Cores) cluster node with 32GB 1600MHz ECC DDR3-RAM (Quad Channel), running on a Red Hat Enterprise Linux 6.2 (64bit) operating system.

[8]The code will be made publicly available on https://sites.google.com/site/yinsong1986/codes soon.

5.5.1 Data Sets

We use three public available real-world data sets: Lastfm, Douban and Flixster, which are crawled from the internet. To be specific, Lastfm[9] has 1892 users, 17632 artists (i.e., items), 92834 user/artist listen weights and 12717 user/user friendship relations; Douban[10] has 129, 490 users, 58, 541 movies (i.e., items), 16, 830, 839 user/movie ratings and 855, 901 user/user friendship relations; Flixster[11] has 787, 210 users, 48, 794 movies, 8, 196, 077 user/movie ratings and 5, 897,316 user/user friendship relations.

5.5.2 Evaluation Metrics

We use two metrics, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE), to measure the prediction accuracy. The metric RSME is defined as:

$$RSME = \sqrt{\frac{\sum_{i,j}(r_{ij} - \hat{r}_{ij})^2}{N}} \tag{5.31}$$

The metric RSME is defined as:

$$MAE = \frac{\sum_{i,j}|r_{ij} - \hat{r}_{ij}|}{N} \tag{5.32}$$

where r_{ij} denotes the rating user i gave to item j, $\hat{r}_{ij}$ denotes the rating user i gave to item j as predicted, and N denotes the number of tested ratings. From the above definitions, it is obvious that a smaller RSME or MAE means a better performance.

5.5.3 Comparison with State-of-the-art Methods

In this section, to demonstrate the effectiveness of our proposed JISM, we compare its performance with the following methods:

[9]Available at www.grouplens.org/node/462.

[10]Available at www.cse.cuhk.edu.hk/irwin.king/pub/data/douban.

[11]Available at www.cs.sfu.ca/ sja25/personal/datasets/.

- RANDOM: this algorithm uses random numbers to predict the missing ratings.
- ItemMean: this algorithm uses the mean value of ratings on each item to predict the missing ratings.
- UserMean: this algorithm uses the mean value of each user's ratings to predict the missing ratings.
- FriendMean: this algorithm uses the mean value of each user's friends' ratings on each item to predict the missing ratings.
- PMF[12]: this algorithm is proposed by Minh and Salakhutdinov (Mnih & Salakhutdinov 2007) and only uses user/item matrix for prediction.
- SocRec[13]: this algorithm is proposed by (Ma et al. 2008) and uses both the user/item matrix and the user/user networks.

In-matrix Prediction

We use 5-fold cross-validation. Specifically, for users who have more than 5 ratings, we evenly split their user/item ratings into 5 folds. We iteratively consider each fold to be a test set and the others to be the training set. We use different numbers (i.e., 4, 3 and 2) of folds to form the training set, which mean approximately 80%, 60% and 40% of the data is used for training, respectively. For users who have less than 5 ratings, we always put them into the training set. In addition, the friendship networks are available during the training process. Through this way, we guarantee the users in the test set always have ratings in the training set and is exactly the scenario for in-matrix prediction.

Figure 5.2-5.4 show the performance (i.e., the RSME and MAE) of SocRec and JISM (with $Q = 5, 10, 15$, respectively) by varying the number of latent

[12]Source code is available at http://www.utstat.toronto.edu/ rsalakhu/BPMF.html
[13]This algorithm can be generalized to the FIP model (Yang et al. 2011).

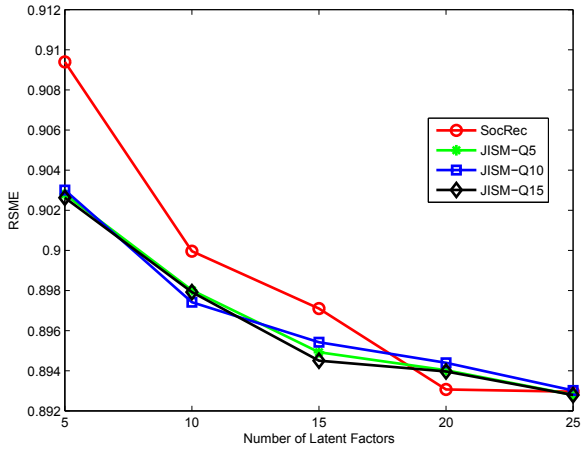
factors K on the LastFM dataset. We do not show other comparative methods since their performance are much more worse than the SocRec and JISM methods. This provides a positive signal that the friendship networks can indeed improve the recommendation accuracy, since the SocRec and JISM algorithms use both the user/item matrix and the user/user networks while others do not. As can be seen from the chart, the proposed JISM generally outperforms the SocRec method on almost all cases. This is reasonable because the JISM models a heterogeneous latent factor space of users, which may better simulate the true behavior of rating generation. In addition, we can observe from Figure 5.2-5.4 that the RMSE/MAE usually converges at $K = 25$ and the increasing of Q does not always decrease the value of RMSE/MAE. Thus, we need to choose proper Q for model selection. Other data sets have similar observations and we omit them for conciseness.

To comprehensively compare the performance on the methods, we provide the detailed experimental results shown in Table 5.2 and we choose $K = 25$ and $Q = 10$. As shown in Table 5.2, our proposed JISM generally outperforms the other methods on all the three data sets. To further validate the statistical significance of our experiments, we also perform the paired t-test (2-tail) between JISM, SocRec and other models on the experimental results. The p-level of t-tests is always smaller than 0.01, which proves the improvements of JISM over other models are statistically significant.

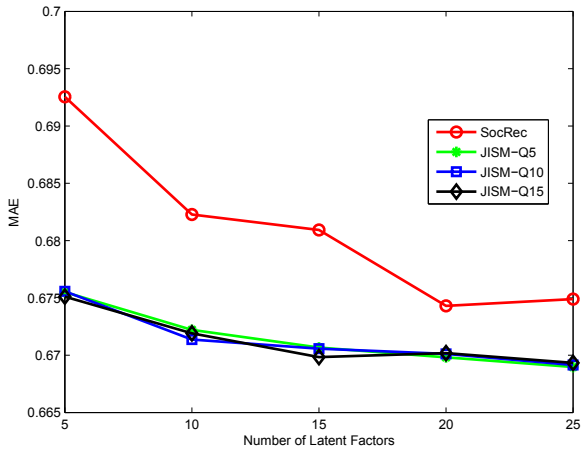
Out-of-matrix Prediction

We also use 5-fold cross validation. We evenly group all the users into 5 folds. For each fold, we train the model by the ratings of these out-of-fold users and all the friendship networks, and then predict the ratings for each user in the fold. Through this way, we guarantee the users in the test set always have no rating in the training set and is exactly the scenario for out-of-matrix prediction.

Table 5.3 shows the detailed experimental results when $K = 15$ and $Q = 10$. The PMF and UserItem are excluded from the table since they



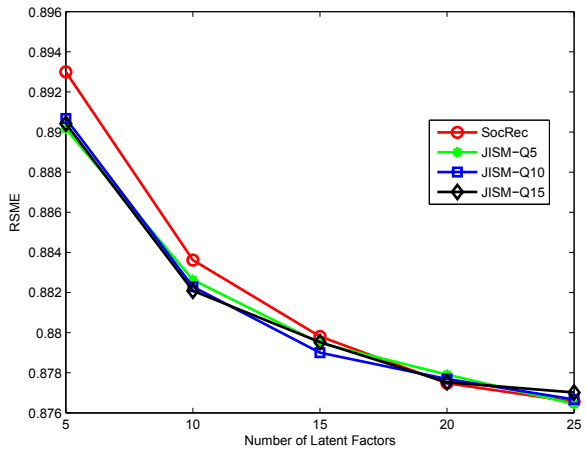
(a) RSME of 40% Training Data



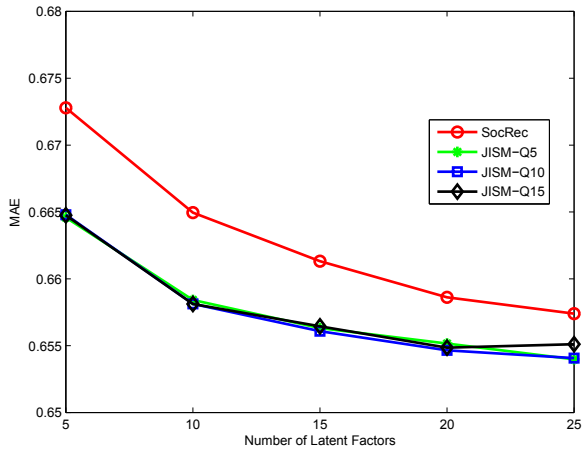
(b) MAE of 40% Training Data

Figure 5.2: Comparison of In-matrix Performance for the Flixster Dataset by Using 40% As Training Data.

cannot generalized for out-of-matrix prediction. Although the performance our proposed JISM is better than other methods on the LastFM and Douban



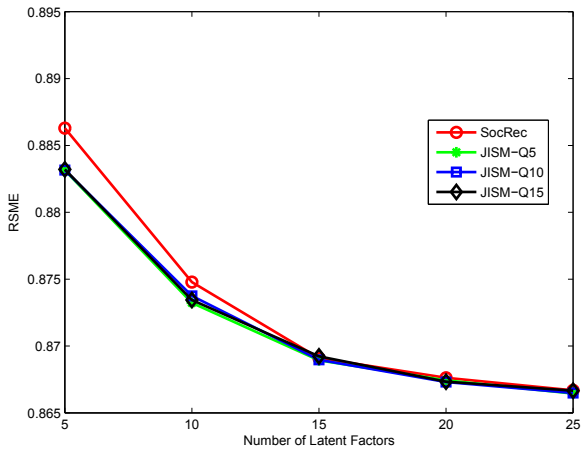
(a) RSME of 60% Training Data



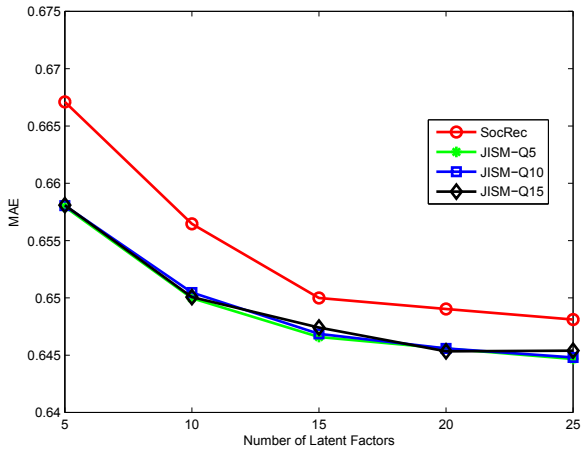
(b) MAE of 60% Training Data

Figure 5.3: Comparison of In-matrix Performance for the Flixster Dataset by Using 60% As Training Data.

dataset, it is worse than that of the ItemMean method on the Flixster dataset. It is surprising on the first look, but a carefully look on the trained model



(a) RSME of 80% Training Data



(b) MAE of 80% Training Data

Figure 5.4: Comparison of In-matrix Performance for the Flixster Dataset by Using 80% As Training Data.

leads the conclusion that the friendship networks are too noisy to utilize without the ratings on the Flixster data set. This is further analyzed in

Section 5.5.5.

5.5.4 Performance Study on Varying the Properties of Users

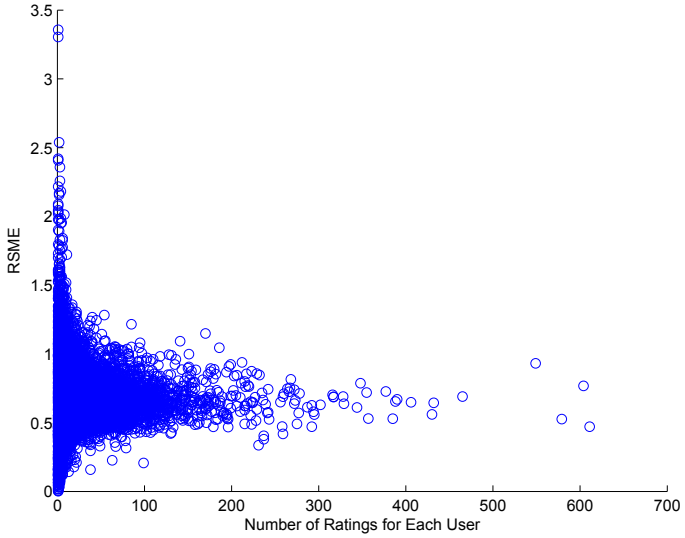
We also study the relationship between recommendation performance and properties of the users for the proposed JISM. Figure 5.5(a) and 5.5(b) show how the RSME/MAE for in-matrix prediction varies as a function of the number of ratings of one user. Figure 5.5(a) and 5.5(b) show how the RSME/MAE varies as a function of the number of one user's friends. As can be seen from Figure 5.5(a) and 5.5(b), users with more ratings tend to have less variance in term of RSME/MAE performance. By contrast, users with few ratings tend to have a diverse RMSE/MAE performance. Another important finding is that the RSME/MAE tends to decrease with the increase of the rating number, which is because more ratings provide more sufficient data for training. Similar trends are also found in Figure 5.5(a) and 5.5(b). Other data sets and the out-of-matrix prediction task have similar observation on the above studies, and we omit the details here for conciseness.

5.5.5 Visualization of Some Interesting results

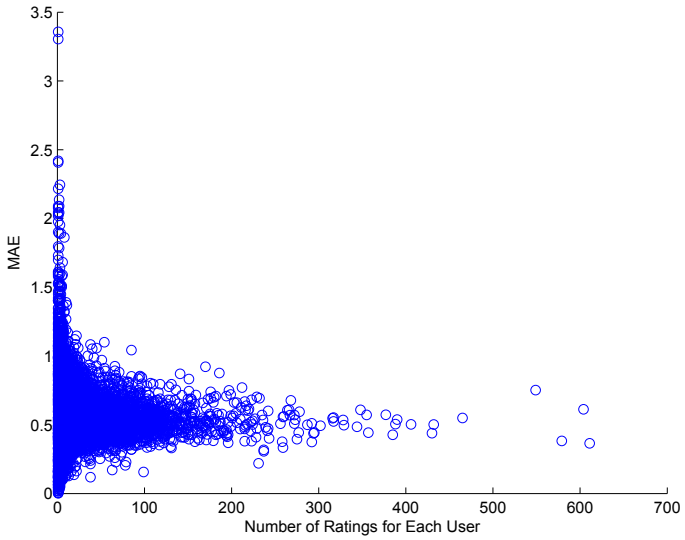
We now turn to an exploratory analysis of our results on the JISM model.

Interest/Social Similarity Contributions

It is interesting to examine how *interest* latent factor and *social* latent factor contribute to the formation of friendships. We define The Interest/Social Contribution as the ratio of $\frac{|t_1|}{|t_1|+|t_2|}$ of the JISM trained on all the data. From its definition, we can see a higher value means a higher contribution of *interest* latent factor on the formation of the friendships. Figure 5.7 shows the the Interest/Social contribution for different number of latent factors on the three data sets. As can be seen from the picture, the Interest/Social contribution is

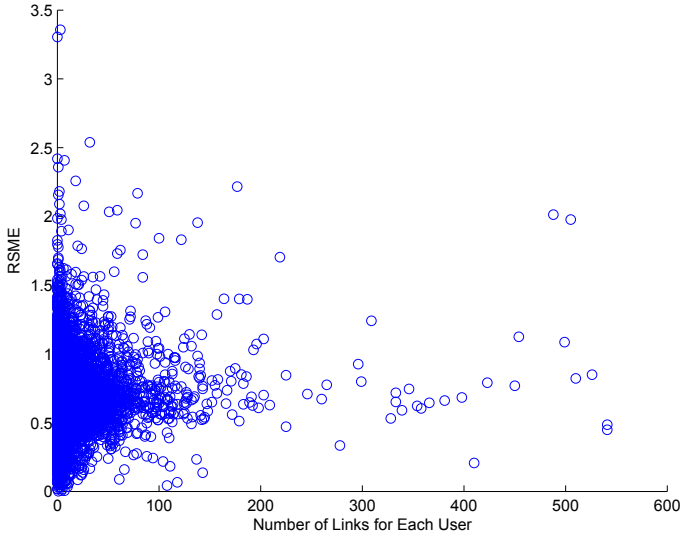


(a) RSME vs # of User Ratings

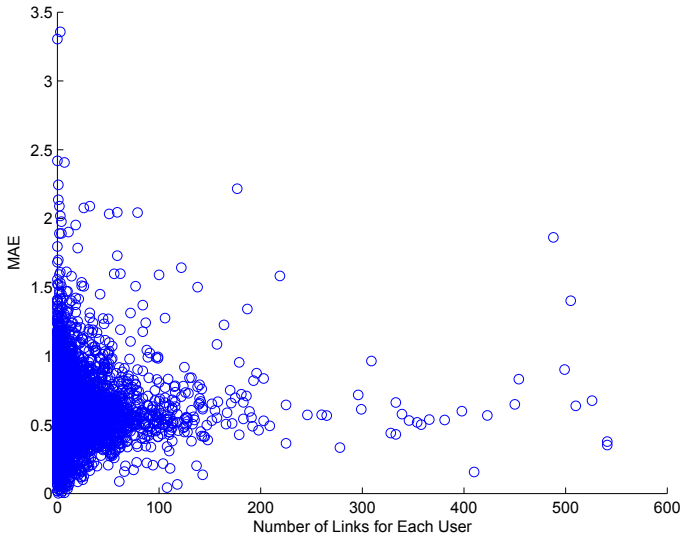


(b) MAE vs # of User Ratings

Figure 5.5: In-matrix Prediction Performance Study against # of Ratings on the Douban Dataset.



(a) RSME vs # of User Links



(b) MAE vs # of User Links

Figure 5.6: In-matrix Prediction Performance Study against # of Links on the Douban Dataset.

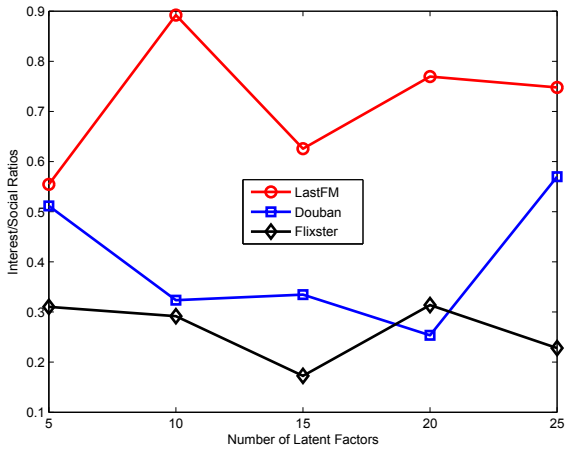
highest on the LastFM data set and this indicates the friendships within this data set is highly correlated to users' interest. The other two data sets have lower Interest/Soical contribution, which indicates the friendships formed there are more caused by extern social affiliation rather than interest. The lowest Interest/Soical contribution on the Flixster data set also supports the results reported in Section 5.5.3, since the friendship networks reveal little information on users' interest and the noise may influence the learning of the JISM.

Grouping of Similar Items Based on the Latent Interest Factors

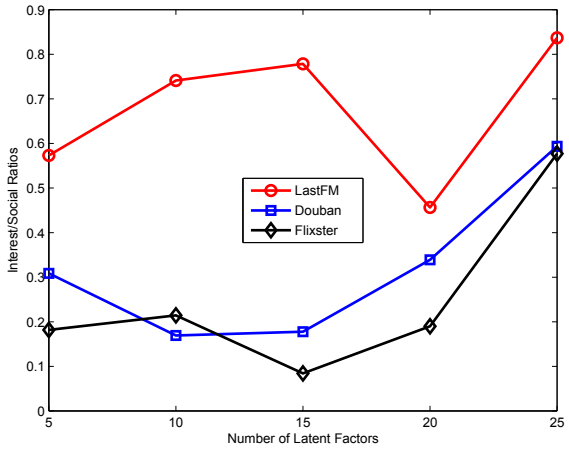
It is also interesting to examine how to group the items by the learned JISM. Because the JISM assigns each item an latent vector, i.e., $\boldsymbol{\lambda}_{2j}$ $(1 \leq j \leq M)$, each dimension of which is a natural grouping indicator, we can list the top items with the greatest values on each dimension as naturally grouping of these items. Since the Douban and Flixster data sets are anonymous, we only plot the result on the LastFM data set. We randomly plot top 4 items for 4 dimensions out of 25 dimensions (i.e., the JISM is trained on the setting of $K = 25$). The results are shown in Figure 5.8. It is notable that some famous singers/bands, such as beatles and Britney Spears are top listed in many dimension. This is good explanation for why they are popular because they can attract users with different interest at the same time.

5.6 Summary

We proposed a probabilistic model for the *social recommendation* problem by using both *preference matrix* and friendship networks. Our study showed that this algorithm outperforms the traditional CF-based methods and other social recommendation methods. In addition, our model can provide some qualitative impression on the data set. For example, we can calculate the Interest/Social contribution to see the friendship formation is more caused by the *interest* or *social* factors.



(a) In-matrix Prediction



(b) Out-of-matrix Prediction

Figure 5.7: The Interest/Social Contribution for different # of Latent Factors and Data sets.

(a) The Beatles (b) Duran Duran (c) Glee Cast (d) Nine Inch Nails

(e) Lady Gaga (f) Radiohead (g) Madonna (h) The Beatles

(i) Depeche Mode (j) Britney Spears (k) Avenged Sevenfold (l) Madonna

(m) Duran Duran (n) Madonna (o) Iron Maiden (p) Envy

Figure 5.8: The Clustering of Artists.

Table 5.1: Notations in the JISM model

	Notation	Meaning	Dimension
Parameters	$\boldsymbol{\mu}_1$	Gassian mean prior on user i's latent interest factor $\mathbf{u}_i$ $(1 \leq i \leq N)$	$K \times 1$
	$\boldsymbol{\Sigma}_1$	Gassian variance prior on user i's latent interest factor $\mathbf{u}_i$ $(1 \leq i \leq N)$	$K \times K$
	$\boldsymbol{\mu}_2$	Gassian mean prior on item j's latent interest factor $\mathbf{v}_j$ $(1 \leq j \leq M)$	$K \times 1$
	$\boldsymbol{\Sigma}_2$	Gassian variance prior on item j's latent interest factor $\mathbf{v}_j$	$K \times K$
	$\boldsymbol{\mu}_3$	Gassian mean prior on user i's latent social factor $\mathbf{w}_i$ $(1 \leq i \leq N)$	$Q \times 1$
	$\boldsymbol{\Sigma}_3$	Gassian mean prior on user i's latent social factor $\mathbf{w}_i$ $(1 \leq i \leq N)$	$Q \times Q$
	$\mathbf{B}$	Social bias parameters	$Q \times K$
	$\mathbf{t}$	Link formation Parameters	3×1
	$\boldsymbol{\tau}$	Rating formation Gaussian variance	1×1
Variational Parameters → Latent Variables	$\boldsymbol{\lambda}_{1i} \rightarrow \boldsymbol{u}_i$	Gassian mean → user i's latent interest factor $(1 \leq i \leq N)$	$K \times 1$
	$\boldsymbol{\nu}_{1i} \rightarrow \boldsymbol{u}_i$	Gassian variance → user i's latent interest factor $(1 \leq i \leq N)$	$K \times 1$
	$\boldsymbol{\lambda}_{2j} \rightarrow \boldsymbol{v}_j$	Gassian mean → item j's latent interest factor $(1 \leq j \leq M)$	$K \times 1$
	$\boldsymbol{\nu}_{2j} \rightarrow \boldsymbol{v}_j$	Gassian variance → item j's latent interest factor $(1 \leq j \leq M)$	$K \times 1$
	$\boldsymbol{\lambda}_{3i} \rightarrow \boldsymbol{w}_i$	Gassian mean → user i's latent social factor $(1 \leq i \leq N)$	$Q \times 1$
	$\boldsymbol{\nu}_{3i} \rightarrow \boldsymbol{w}_i$	Gassian variance → user i's latent social factor $(1 \leq i \leq N)$	$Q \times 1$
	$\boldsymbol{\xi}$	Free variational parameters	$N \times N$
Constants	N	The number of users	1×1
	M	The number of items	1×1
	K	The number of dimensions for the latent interest factor	1×1
	Q	The number of dimensions for the latent social factor	1×1
Data	E	The link data	$N \times N$
	R	The rating data	$N \times M$

Table 5.2: Detailed comparison of in-matrix Prediction (K = 25, Q = 10)

Dataset	Training	Metrics	**Random**	**UserMean**	**ItemMean**	**FriendMean**	**PMF**	**SocRec**	**JISM**
LastFM	40%	**RSME**	2.0628 ± 0.0076	0.8209 ± 0.0064	0.7888 ± 0.0046	0.8369 ± 0.0080	0.8167 ± 0.0064	0.7746 ± 0.0058	**0.7718 ± 0.0064**
		MAE	1.7295 ± 0.0083	0.5309 ± 0.0030	0.4804 ± 0.0021	0.5329 ± 0.0034	0.5328 ± 0.0034	0.4699 ± 0.0024	**0.4496 ± 0.0032**
	60%	**RSME**	2.0635 ± 0.0102	0.8152 ± 0.0075	0.7825 ± 0.0046	0.8365 ± 0.0073	0.8099 ± 0.0062	0.7658 ± 0.0056	**0.7648 ± 0.0056**
		MAE	1.7329 ± 0.0125	0.5271 ± 0.0044	0.4762 ± 0.0025	0.5335 ± 0.0043	0.5274 ± 0.0032	0.4558 ± 0.0037	**0.4454 ± 0.0036**
	80%	**RSME**	2.0633 ± 0.0096	0.8117 ± 0.0074	0.7790 ± 0.0046	0.8401 ± 0.0064	0.8073 ± 0.0059	0.7607 ± 0.0057	**0.7604 ± 0.0058**
		MAE	1.7301 ± 0.0100	0.5242 ± 0.0034	0.4732 ± 0.0028	0.5353 ± 0.0038	0.5251 ± 0.0031	0.4485 ± 0.0033	**0.4423 ± 0.0033**
Douban	40%	**RSME**	1.6914 ± 0.0006	0.7822 ± 0.0005	0.8482 ± 0.0004	0.9342 ± 0.0006	0.7964 ± 0.0006	**0.7066 ± 0.0005**	0.7125 ± 0.0154
		MAE	1.3819 ± 0.0005	0.6266 ± 0.0003	0.6824 ± 0.0003	0.7334 ± 0.0005	0.6230 ± 0.0004	**0.5582 ± 0.0004**	0.5619 ± 0.0088
	60%	**RSME**	1.6912 ± 0.0005	0.7814 ± 0.0005	0.8458 ± 0.0005	0.9346 ± 0.0003	0.7551 ± 0.0006	0.6987 ± 0.0004	**0.6969 ± 0.0006**
		MAE	1.3817 ± 0.0003	0.6261 ± 0.0003	0.6806 ± 0.0004	0.7309 ± 0.0002	0.5918 ± 0.0003	0.5511 ± 0.0002	**0.5501 ± 0.0004**
	80%	**RSME**	1.6912 ± 0.0008	0.7810 ± 0.0005	0.8446 ± 0.0005	0.9332 ± 0.0005	0.7334 ± 0.0006	0.6926 ± 0.0006	**0.6916 ± 0.0005**
		MAE	1.3817 ± 0.0007	0.6258 ± 0.0003	0.6797 ± 0.0003	0.7281 ± 0.0004	0.5758 ± 0.0004	0.5458 ± 0.0004	**0.5449 ± 0.0003**
Flixster	40%	**RSME**	2.0063 ± 0.0008	1.0897 ± 0.0006	0.9482 ± 0.0011	1.2189 ± 0.0003	0.9434 ± 0.0005	**0.8930 ± 0.0010**	**0.8930 ± 0.0012**
		MAE	1.6448 ± 0.0010	0.8866 ± 0.0008	0.7058 ± 0.0007	0.9645 ± 0.0001	0.7137 ± 0.0005	0.6749 ± 0.0026	**0.6691 ± 0.0016**
	60%	**RSME**	2.0055 ± 0.0011	1.0871 ± 0.0006	0.9393 ± 0.0009	1.2176 ± 0.0005	0.9276 ± 0.0010	**0.8765 ± 0.0009**	0.8767 ± 0.0010
		MAE	1.6438 ± 0.0011	0.8833 ± 0.0008	0.7008 ± 0.0006	0.9620 ± 0.0004	0.6966 ± 0.0008	0.6574 ± 0.0023	**0.6541 ± 0.0012**
	80%	**RSME**	2.0060 ± 0.0013	1.0851 ± 0.0006	0.9346 ± 0.0010	1.2164 ± 0.0002	0.9155 ± 0.0005	0.8667 ± 0.0003	**0.8665 ± 0.0005**
		MAE	1.6442 ± 0.0016	0.8806 ± 0.0008	0.6980 ± 0.0006	0.9598 ± 0.0002	0.6847 ± 0.0004	0.6481 ± 0.0018	**0.6448 ± 0.0010**

Table 5.3: Detailed comparison of out-of-matrix Prediction (K = 15, Q = 5)

Dataset	Metrics	**Random**	**ItemMean**	**FriendMean**	**SocRec**	**JISM**
LastFM	**RSME**	2.0584 ± 0.0122	0.8131 ± 0.0129	0.8412 ± 0.0115	0.8760 ± 0.0197	**0.7928 ± 0.0131**
	MAE	1.7253 ± 0.0128	0.5242 ± 0.0074	0.5356 ± 0.0069	0.6623 ± 0.0195	**0.5053 ± 0.0060**
Douban	**RSME**	1.8352 ± 0.0027	0.8066 ± 0.0029	0.9474 ± 0.0024	0.8387 ± 0.0105	**0.8020 ± 0.0045**
	MAE	1.5085 ± 0.0026	0.6595 ± 0.0020	0.7518 ± 0.0023	0.6876 ± 0.0092	**0.6564 ± 0.0036**
Flixster	**RSME**	2.1190 ± 0.0323	**1.2191 ± 0.0232**	1.3982 ± 0.0326	1.2598 ± 0.0126	1.2353 ± 0.0193
	MAE	1.7464 ± 0.0293	**1.0036 ± 0.0215**	1.1074 ± 0.0163	1.0746 ± 0.0129	1.0389 ± 0.0180

Chapter 6

Enhance the Sequence Anomaly Detection

In certain scenarios, purely BNs-based modeling of the heterogeneous data may fail to provide a well result on predictive tasks due to the inaccurate estimation of parameters, which is commonly seen in latent variable models (Jaakkola & Haussler 1999, Tsuda, Kawanabe & Muller 2002). Thus, to overcome the above problem, this chapter presents a novel hybrid framework of combining BNs-based models and discriminative classifiers to detect abnormal sequences in an one-class setting (i.e., only normal data are available), which is applicable to various domains. Examples include intrusion detection, fault detection and speaker verification. Detecting abnormal sequences with only normal data presents several challenges for anomaly detection: the weak discrimination of normal and abnormal sequences; the unavailability of the abnormal data and other issues. Traditional model-based anomaly detection techniques (Chandola, Banerjee & Kumar 2009) can solve some of the above issues but with limited discrimination power (because of directly modeling the normal data). In order to enhance the discriminative power for anomaly detection, we try to extract discriminative features from generative models based on some theoretical analysis, and develop a new anomaly detection framework on top of the feature extractor. The proposed approach

firstly projects all the sequential data into a model-based equal length feature space (this is theoretically proven to have better discriminative power than the model itself) , and then adopts a classifier learned from the transformed data to detect anomalies. Experimental evaluation on both the synthetic and real-world data shows that our proposed approach outperforms several anomaly detection baseline algorithms for sequential data.

6.1 Introduction

Anomaly detection has traditionally been an important part of behavior analysis, whose aim is to find abnormal patterns in data that do not conform to expected (normal) behavior (Chandola, Banerjee & Kumar 2009). Most of the traditional anomaly detection techniques focus on static behavioral records or transactional data (Barnett & Lewis 1994). But in many real life scenarios, behaviors are dynamic and naturally organized as sequential data and the target of anomaly detection is collections of behaviors other than individual ones. One such example could be seen in intrusion detection for the operating system, i.e., to detect malicious programs (processes) from the normal execution processes. Each process (program) is denoted by its trace, which is a sequence of system calls used by that process from the beginning of its execution to the end. Table 6.1 shows three example programs in which normal and malicious ones are mixed (Chandola, Banerjee & Kumar 2009). Each row records the sequential system calls (e.g., read and open) of one program. Another example could be found in detecting abnormal Electrocardiogram (ECG) signals. ECG signals record the dynamic behaviors of the heart over a period of time, which could be further utilized to characterize the heart's condition. Figure 6.1(a) depicts two sampled ECG signals, one of which is from a healthy heart (i.e., normal) and the other is from an attacked heart (i.e., abnormal). From the above examples, we can intuitively find two things: firstly, these sequences are characterized by their dynamics; secondly, the normal and abnormal sequences are very similar by their appearance.

Table 6.1: Some Sample Data of Operating System Call Traces.

open	read	mmap	mmap	open	read	$\cdots$
open	mmap	mmap	read	open	$\cdots$	$\cdots$
open	close	open	close	open	mmap	$\cdots$

For the purpose of detecting these abnormal sequential behaviors, we should consider the dynamic characteristics of sequential data, which is different from anomaly detection in static data. Another challenging issue is how to discriminate these abnormal dynamic behaviors from highly resemblant normal behaviors.

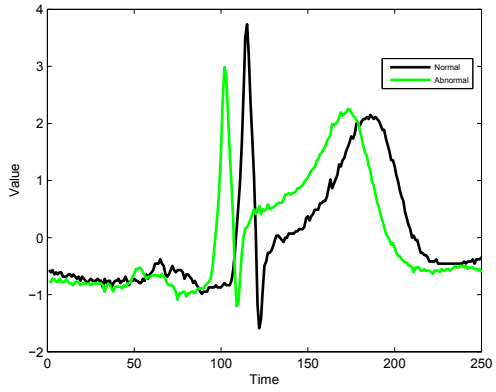
The above scenarios form a challenging issue, that is to detect abnormal behavioral sequences (which highly resemble normal behavioral sequences) in a set of sequences. To be more precise, the problem we will explore in this chapter can be formally stated as follows:

Definition 2 *Given a set of n training normal sequences, $\mathcal{X}^{tr}$, and a set of m test sequences $\mathcal{X}^{te}$, find a set of abnormal sequences $\mathcal{X}^{a} \subset \mathcal{X}^{te}$.*

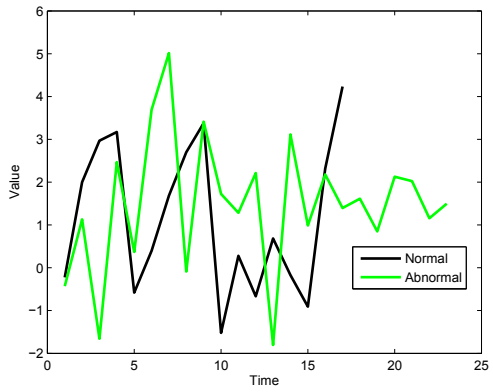
The key challenges of the above problem are listed in the following: Firstly, the sequences are quite dynamic, which is not intuitive to capture. Secondly, the abnormal sequences are usually highly similar to the normal ones in nature. This can be seen from Table 6.1 and Figure 6.1(a). In addition, other related issues with anomaly detection for sequential data include variable lengths of sequences, and imbalance betweens normal and abnormal data (i.e., one-class mode in this chapter).

Hence, we propose a novel anomaly detection framework to deal with the issue of limited discriminative power in the traditional model-based approaches. The main contributions of this chapter are listed as follows:

- Based on the analysis of Bayes error, we provide the theoretical principle of extracting discriminative features for one-class anomaly detection.



(a)



(b)

Figure 6.1: (a) Some Sampled Signals from the ECG Data Set. (b) Some Sample Sequences from the Synthetic Data Set.

- A flexible three-phase implementation framework is proposed: *Phase 1* extracts discriminative features from the sequences based on the aforementioned theoretical feature extractor principle; *Phase 2* learns a discriminative classifier (e.g., SVM) on this new feature space; *Phase 3* applies the learned classifier to detect fraudulent sequences.

- Intensive experiments done on several synthetic and real-world data sets empirically demonstrate the effectiveness of the proposed approach compared to other existing baseline anomaly detection techniques.

The remainder of this chapter is organized as follows. Section 6.2 reviews the existing model-based anomaly detection and discusses its limitations, followed by theoretical analysis for enhancing the discriminative power for anomaly detection in Section 6.3. Section 6.4 proposes an implementation framework based on the theoretical analysis. After that, Section 6.5 and 6.6 describe empirical results on both synthetic and real-world data sets. Section 6.7 summarizes this chapter.

6.2 Model-based Anomaly Detection and Its Limitations

In this part, we briefly review the commonly-used model-based framework to handle one-class anomaly detection for sequential data (Joshi & Phoha 2005, Warrender et al. 1999, Cao et al. 2010) and point out its limitation from the theoretical perspective.

6.2.1 The Anomaly Detection Algorithm

The goal of sequence anomaly detection is to take an input sequence $\mathbf{x}$ and assign it to two discrete classes y where $y = 1, -1$ (1 denotes normal class and -1 denotes abnormal class). Generally speaking, the model-based framework detects anomaly by thresholding the likelihood

$$P_{\theta_1^*}(\mathbf{x}) < Th_0 \tag{6.1}$$

where $P_{\theta_1^*}(\mathbf{x}) = P(\mathbf{x}; \theta^* | y = 1)$ (and this form of notation has similar meanings in the rest of the paper), θ_1^* is the normal model parameters (and usually estimated as $\hat{\theta}_1$ from training data $\mathcal{X}^{tr}$) for normal class. The sample $\mathbf{x}$ satisfies Equation 6.1 is detected as an anomaly. The model-based algorithm

Algorithm 6.1: Model-based Sequential Anomaly Detection.

Input : A training set $\mathcal{X}^{tr}$, A testing set $\mathcal{X}^{te}$, A threshold Th_0.

Output: An anomaly set $\mathcal{X}^a$.

```
1 X^a → ∅;
2 Learn the normal model θ̂_1 on the training set X^tr;
3 forall the x ∈ X^te do
4 |   Compute the likelihood of x given θ̂_1: P_θ̂1(x|y = 1);
5 end
6 if P_θ̂1(x|y = 1) < Th_0 then
7 |   x → X^a;
8 end
9 Output the anomaly set X^a;
```

consists of two stages: the first stage is to profile the normal sequence with a generative model $\hat{\theta}_1$ while the second stage is to detect abnormal sequences in the test data set according to Equation 6.1. Algorithm 6.1 summarizes the above algorithm as following.

6.2.2 Limitations: Theoretical Analysis

As reviewed in the above, the one-class sequence anomaly detection is to predict discrete class labels (i.e., normal or abnormal), which is similar to the aim of classification problem. In fact, the difference between the problem considered in this chapter and the classification one is the availability of training data. In this chapter, only normal data is available for training and thus can be seen as a special case of classification problem, which is helpful to theoretical analysis.

For a standard classification problem, assuming we know the 'oracle' (i.e., true) parameters θ^* (θ_1^* denotes the parameters for the normal class and θ_{-1}^* denotes the parameters for the abnormal class) for generating the data, classifying an input $\mathbf{x}$ is to threshold the posterior probability $P(y = 1|\mathbf{x}; \theta^*)$

(Devroye, Gyorfi & Lugosi 1996) with a threshold $\frac{1}{2}$, which is equivalent to the following oracle classifier (and the proof can be found in Appendix A):

$$P_{\theta_1^*}(\mathbf{x}) < Th_1 \cdot P_{\theta_{-1}^*}(\mathbf{x}) \tag{6.2}$$

The sample $\mathbf{x}$ satisfies Equation 6.2 is detected as an anomaly. Compared to Equation 6.1, we can see that the model-based anomaly detection algorithm does not consider the term $P_{\theta_{-1}^*}(\mathbf{x})$ for classification decision making and thus has less discriminative power for classification, which could harm the anomaly detection result. Here the Bayes error (Devroye et al. 1996) is adopted to measure the performance of the anomaly detection algorithms. It is also an indicator of the discriminative power since good discrimination leads to good anomaly detection performance. Suppose the oracle classifier expressed as Equation 6.2 has the oracle Bayes error L^* for all $\mathbf{x} \in \mathcal{X}$, the performance of the model-based anomaly detection algorithm expressed as Equation 6.1 could not achieve good approximation to L^* in general cases. To enhance the discriminative power for anomaly detection, we try to find another method whose classification performance could have a better approximation to L^*, which will be discussed in the following sections.

6.3 How to Enhance the Discriminative Power: Theoretical Analysis

The above section has pointed out the limitation of the model-based anomaly detection algorithm and our aim is to find a method to approximate the oracle Bayes error L^*. Inspired by (Tsuda, Kawanabe, Ratsch, Sonnenburg & Muller 2002), we first propose a well-founded performance measure to theoretically evaluate the approximation in Section 6.3.1, and then suggest an approximation method of extracting proper features combined with a classifier in Section 6.3.2.

6.3.1 Objective Function

It is straightforward to see that the oracle classifier Equation 6.2 has the most discriminative power for classifying normal and abnormal sequences which achieves the oracle Bayes error L^*. Thus, it is desirable that the theoretical Bayes error of the proposed anomaly detection algorithm should approach L^* as close as possible. Here we consider a linear classifier $\mathbf{w^T f}_{\hat{\theta}}(\mathbf{x})+b$ combined with a feature extractor $f_{\hat{\theta}}(x)$ ($f_{\hat{\theta}}(x) : \mathcal{X} \to \mathbb{R}^D$ and $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$) to approximate the oracle classifier. The corresponding Bayes error is

$$R(f_{\hat{\theta}}) = \min_{\mathbf{w}\in\mathcal{S},b\in\mathbb{R}} E_{x,y}\Phi[-y(\mathbf{w}^T f_{\hat{\theta}}(\mathbf{x}) + b)] \tag{6.3}$$

where $\mathcal{S} = \{\mathbf{w}|\mathbf{w} \in \mathbb{R}^D\}$, $\Phi[a]$ is the step function (which is 1 if $a > 0$ and 0 otherwise), and $E_{x,y}$ denotes the expectation with respect to the true distribution $p(\mathbf{x}, y|\theta^*)$. $R(f_{\hat{\theta}})$ is at least as large as the oracle Bayes error L^* and $R(f_{\hat{\theta}}) = L^*$ only if the linear classifier implements the same decision rule as the oracle classifier (Fukunaga 1990). Usually $\mathbf{w}$ and b can be determined by a learning algorithm and we assume the optimal learning algorithm is used. When $\mathbf{w}$ and b are optimally chosen, the remaining part to determine is the feature extractor $f_{\hat{\theta}}(x)$ that minimize $R(f_{\hat{\theta}}) - L^*$, which describes how close the Bayes error to the oracle one.

Now it is natural to design a feature extractor that minimizes the objective function $R(f_{\hat{\theta}}) - L^*$. Direct optimization of this function is difficult because there exists a non differentiable function Φ. Alternatively, we turn to minimize its upper bound $2D(f_{\hat{\theta}})$, which generally has the following relationship with the objective function (Devroye et al. 1996):

$$R(f_{\hat{\theta}}) - L^* \leq 2D(f_{\hat{\theta}}). \tag{6.4}$$

where $D(f_{\hat{\theta}}) = \min_{\mathbf{w}\in\mathbb{R}^D,b\in\mathbb{R}} E_{\mathbf{x}}|F(\mathbf{w}^T f_{\hat{\theta}}(\mathbf{x}) + b) - P(y = 1 \mid \mathbf{x}; \theta^*)|$ and $F(t) = \frac{1}{(1+exp(-t))}$.

The relationship between $D(f_{\hat{\theta}})$ and $R(f_{\hat{\theta}})$ is illustrated as follows: Let $\hat{L}$ be the classification error rate of an arbitrary posterior probability estimator

$\hat{P}(y=+1|\mathbf{x})$. The following inequality is known:

$$\hat{L} - L^* \leq 2E_{\mathbf{x}}|P(y=+1 \mid \mathbf{x}) - P(y=+1 \mid \mathbf{x}, \theta^*)| \tag{6.5}$$

When we use $P(y=+1|\mathbf{x}) := F(\mathbf{w}^T f_{\hat{\theta}}(\mathbf{x}) + b)$, $D(f_{\hat{\theta}})$ becomes an alternative objective function to minimize whose minimization leads to the minimization of $R(f_{\hat{\theta}}) - L^*$ in terms of upper bounds.

6.3.2 Proposed Feature Extractor

On the basis of the above object function, we further propose a feature extractor that achieves small $D(f_{\hat{\theta}})$. It is straightforward to see that a feature extractor $f_{\hat{\theta}}(\mathbf{x})$ satisfies

$$\mathbf{w}^T \mathbf{f}_{\hat{\theta}}(\mathbf{x}) + b = F^{-1}(P(y=1|\mathbf{x}; \theta^*)) \text{ for all } \mathbf{x} \in \mathcal{X} \tag{6.6}$$

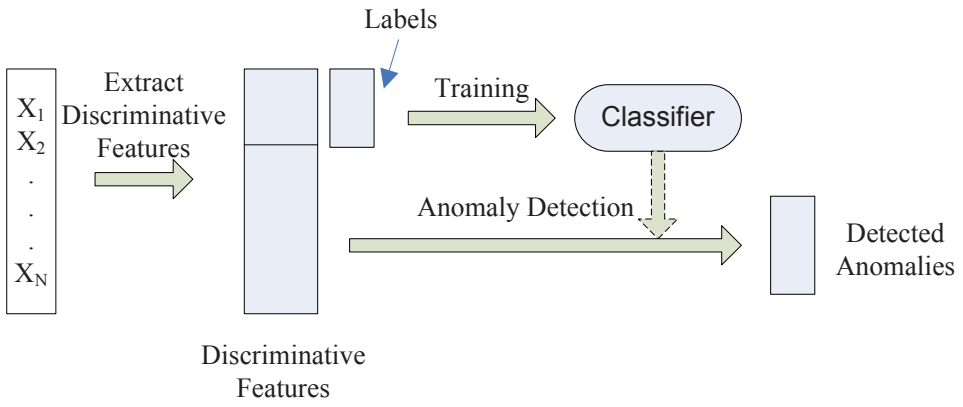
with certain values of $\mathbf{w}$ and b, we have $D(f_{\hat{\theta}}) = 0$, which is the minimum point. However, since the oracle parameter θ^* is unknown, we cannot construct this optimal feature extractor $f_{\hat{\theta}}$ according to $F^{-1}(P(y=1|\mathbf{x}; \theta^*))$. However, it can be approximated by its Taylor expansion at the point $\hat{\theta}_1$ estimated from the training data. The corresponding approximate optimal feature extractor is as follows:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) := (\partial_{\theta^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\theta^*_{1p}} g(\hat{\theta}_1))^T \tag{6.7}$$

where $g(\theta^*_1) = \log P_{\theta^*_1}(\mathbf{x})$, $\partial_{\theta^*_{1i}} g(\hat{\theta}_1)$ $(1 \leq i \leq p)$ is $g(\theta^*_1)$'s gradient with respect to θ^*_{1i} at point $\hat{\theta}_1$ and can be seen as a function of $\mathbf{x}$ since $\hat{\theta}_1$ is fixed. Thus the extracted feature is a set of functions of $\mathbf{x}$. The proof can be found in Appendix B. It is also notable that the theoretical performance of the proposed feature extractor with optimal classifier is better than that of the model-based algorithm and the proof can be found in Appendix C.

6.4 Proposed Implementation Framework

Motivated by the theoretical analysis of enhancing the discriminative power for the model-based anomaly detection algorithm, we further propose an ef-



Algorithm 6.2: The Proposed Framework

Input : A Training set $\mathcal{X}^{tr}$, A Testing set $\mathcal{X}^{te}$, A Threshold Th_2.

Output: An anomaly set $\mathcal{X}^a$.

1 Given $\mathcal{X}^{tr}$ and $\mathcal{X}^{te}$, extract discriminative features $\mathcal{S}$ based on generative models;

2 Given $\mathcal{S}^{tr}$, construct a one-class discriminative classifier C;

3 Given $\mathcal{S}^{te}$, C and Th_2, output $\mathcal{X}^a$ for detected anomalies;

Figure 6.2: The Flow Chart and Algorithm of the Proposed Framework

ficient implementation framework, called model-based discriminative feature (MDF) anomaly detection framework. A key challenge regarding implementation is to choose proper $\mathbf{w}$ of the classifier for anomaly detection, since the principle of feature extractor is already given. An overview of the MDF framework is shown in Figure 6.2. More specifically, *Phase 1* is to extract the features on the basis of $f_{\hat{\theta}}$ in the form of Equation 6.7. Then in *Phase 2*, based on the extracted features, the corresponding optimal $\mathbf{w}$ is learned using a one-class support vector machine (SVM). Finally, the anomaly detection task is performed by the learned classifier produced in *Phase 3*. The following sections will describe the details of the three phases.

6.4.1 Phase 1: Feature Extraction

For the first phase, we need to choose a proper model to extract features based on it. In this chapter, we assume that sequences could be well modeled by hidden Markov Models (HMMs), because its expressive power of modeling real-world dynamic behavioral process, such as speech signal (Rabiner 1990), biological sequence(Baldi & Brunak 2001), gestures (Alon et al. 2003) and videos (Wang & Singh 2003).

Here we first review the basic notions of HMMs and then give out the form of derivatives $\partial_{\theta^*_{1i}} g(\hat{\theta}_1)$ $(1 \leq i \leq p)$ used for feature extraction. Formally, a first-order HMM can be formally defined by:

- A set of Q possible hidden states denoted as $\mathcal{Q} = \{1, 2, \cdots, Q\}$, where $i (1 \leq i \leq Q)$ is a possible hidden state. The state at time t is denoted as q_t and $q_t \in \mathcal{Q}$.

- The hidden state transition matrix is $A = a_{ij}$, where $a_{ij} = P(q_{t+1} = j | q_t = i), 1 \leq i, j \leq Q$ is the probability for the transition from i to j.

- The observation vector $\mathbf{x}_t$ at time t is supposed to be governed by the corresponding conditional probability distribution $b_j(\mathbf{x}_t)$ $(1 \leq j \leq Q)$. When the observation vectors are discrete symbols, $b_j(\mathbf{x}_t)$ $(1 \leq j \leq Q)$ for each hidden state j is usually associated with the multinomial distribution as $b_j(\mathbf{x}_t) = \prod_{k=1}^{K} \mu_{jk}^{x_{tk}}$. Here we use the 1-of-K scheme (i.e., $\mathbf{x}_t = [x_{t1}, \cdots, x_{tK}]^T$, subects to $\sum_k x_{tk} = 1$) to represent the discrete observation as a K-dimensional vector where K is the number of vocabulary for the discrete symbols. When the observation vectors are continuous, $\mathbf{x}_t$ (with hidden state j) is usually assumed to subject to a mixture of Gaussian distributions $\sum_{k=1}^{K} c_{jk} \mathcal{N}(\mathbf{x}_t | \mu_{jk}, \Sigma_{jk})$, where c_{jk} is the mixture coefficient for the k^{th} Gaussian mixture in the state j, $\mathcal{N}$ is a Gaussian distribution density with the mean vector μ_{jk} and the covariance matrix Σ_{jk}.

- The initial state probability distribution $\pi = (\pi_1, \pi_2, \cdots, \pi_Q)$, where

$$\pi_i = P(q_1 = i), 1 \leq i \leq Q.$$

Thus, an HMM can be denoted as $\theta = \{A, B, \pi\}$. Let $\mathbf{x}$ be an observation sequence, the parameters of an HMM are approximately learned by using the Baum-Welch algorithm (Baum et al. 1970) given a set of sequences $\mathcal{X}^{tr}$. On the other hand, the partial derivatives of $g(\theta_1^*)$ at the point of $\hat{\theta}_\mathbf{1} = \{\hat{A}, \hat{B}, \hat{\pi}\}$ can be calculated by using $\hat{\xi}_t$ and $\hat{\gamma}_t$, which can be obtained by the forward-backward algorithm (Rabiner 1990). Specifically, $\hat{\xi}_t(i,j)$ is the probability of being in state i at time t and state j at time $t+1$ given the model $\hat{\theta}_\mathbf{1}$ and the observation sequence $\mathbf{x}$, which is $\hat{\xi}_t(i,j) = P(q_t = i, q_{t+1} = j|\mathbf{x}; \hat{\theta}_\mathbf{1})$. For discrete observations, $\hat{\gamma}_t(j)$ is the probability of being in state j at time t, which is $\hat{\gamma}_t(j) = P(q_t = j|\mathbf{x}; \hat{\theta}_\mathbf{1})$; for continuous observations, $\hat{\gamma}_t(j,k)$ is the probability of being in state j at time t with the k^{th} Gaussian mixture component accounting for $\mathbf{x}_t$, which is $\hat{\gamma}_t(j,k) = P(q_t = j, M_{jt} = k|\mathbf{x}; \hat{\theta}_\mathbf{1})$, where M_{jt} is a random variable indicating the mixture component at time t in state j. Then partial derivatives of $g(\theta_1^*)$ with respect to the parameters θ_1^* at a point $\hat{\theta}_\mathbf{1}$ (estimated from the training data) are listed as following (Velivelli et al. 2006):

$$\partial_{a_{ij}^*} g(\hat{\theta}_1) = \sum_{t=1}^{T-1} \frac{\hat{\xi}_t(i,j)}{\hat{a}_{ij}} \tag{6.8}$$

for discrete observations:

$$\begin{aligned} \partial_{\pi_i^*} g(\hat{\theta}_1) &= \frac{\hat{\gamma}_t(i)}{\hat{\pi}_i} \\ \partial_{\mu_{jk}^*} g(\hat{\theta}_1) &= \frac{\sum_{t=1}^{T} \hat{\gamma}_t(j) x_{tk}}{\hat{\mu}_{jk}} \end{aligned} \tag{6.9}$$

for continuous observations:

$$\begin{aligned}
\partial_{\pi_i^*} g(\hat{\theta}_1) &= \frac{\hat{\gamma}_t(i,1)}{\hat{\pi}_i} \\
\partial_{c_{jk}^*} g(\hat{\theta}_1) &= \sum_{t=1}^{T} \frac{\hat{\gamma}_t(j,k)}{\hat{c}_{ij}} \\
\partial_{\mu_{ij}^*} g(\hat{\theta}_1) &= \sum_{t=1}^{T} \hat{\gamma}_t(j,k)[\hat{\Sigma}_{jk}^{-1}]^T(\mathbf{x}_t - \hat{\mu}_{jk}) \\
\partial_{\hat{\Sigma}_{jk}^*} g(\hat{\theta}_1) &= \sum_{t=1}^{T} \frac{\hat{\gamma}_t(j,k)}{2}[G - vec(\hat{\Sigma}_{jk}^{-1})] \qquad (6.10)
\end{aligned}$$

where $vec(F) = [F_{11}, F_{12}, \cdots, F_{M1}, F_{MN}]^T$ when F is a matrix of size $M \times N$. $G = [(\mathbf{x}_t - \hat{\mu}_{jk})^T \Sigma_{jk}^{-1} \otimes (\mathbf{x}_t - \hat{\mu}_{jk})^T \Sigma_{jk}^{-1}]^T$ and $\otimes$ denotes the kronecker product.

- $\mathcal{X}^{tr}$: a training data set consists of only normal sequences.
- $\mathcal{X}^{te}$: a testing data set consists of both normal and abnormal sequences.

Then the algorithm for the feature extractor can be summarized in Algorithm 6.3: step 1 estimates parameters $\hat{\theta}_\mathbf{1}$ of the HMM from the training data; then step 2-9 extract the discriminative feature using Equation 6.8-6.10 for each sequence $\mathbf{x} \in \mathcal{X}^{tr} \cup \mathcal{X}^{te}$.

6.4.2 Phase 2: Learning of the Optimal Linear Classifier

This phase tries to construct a linear classifier with the optimal $\mathbf{w}$, one-class SVM (Hsu, Chang & Lin 2003) has become a natural choice, since it is linear classifier and only the normal sequences are provided for training. For estimating the optimal decision boundary (i.e., the parameter $\mathbf{w}$ and b) (Joachims 2000) and (Tran, Zhang & Li 2003) have proposed different object function to optimize. Here we adopt a similar object function similar to (Tran et al. 2003) for simplicity and effectiveness. To be more specific, suppose there is a training data set $\mathcal{S}^{tr}$ consists of m training sequences

Algorithm 6.3: The Proposed Feature Extractor.

Input : A training set $\mathcal{X}^{tr}$, A testing set $\mathcal{X}^{te}$, A threshold Th_0.

Output: An anomaly set $\mathcal{X}^a$.

Given $\mathcal{X}^{tr}$ train an HMM $\hat{\theta}_{\mathbf{1}}$;

forall the $\mathbf{x} \in \mathcal{X}^{tr} \cup \mathcal{X}^{te}$ **do**

  Given $\hat{\theta}_{\mathbf{1}}$, construct the corresponding discriminative features ;

  if $\mathbf{x}$ *is discrete* **then**

    Construct features as:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) = (\partial_{a^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\pi^*_1} g(\hat{\theta}_1), \cdots, \partial_{\mu^*_{11}} g(\hat{\theta}_1), \cdots)^T$$

    according to Equation 6.8 and 6.9;

  else

    Construct features as:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) = (\partial_{a^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\pi^*_1} g(\hat{\theta}_1), \cdots, \partial_{c^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\mu^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\sigma^*_{11}} g(\hat{\theta}_1), \cdots)^T$$

    according to Equation 6.8 and 6.10;

  end

  $\mathbf{s} \to \mathcal{S}$;

end

$\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(m)}$, the learning objective function based on the maximum margin theory is (Scholkopf & Smola 2002):

$$\min_{\mathbf{w},\rho,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu m}\sum_i \xi_i - \rho, \tag{6.11}$$

$$\text{subject to } \mathbf{w}\Phi(\mathbf{x}^{(i)}) \leq \rho - \xi_i, \xi_i \geq 0, 1 \leq i \leq m. \tag{6.12}$$

Algorithm 6.4: The Learning Algorithm.

Input : A training set of extracted features $\mathcal{S}^{tr}$.

Output: A constructed classifier C.

1 Given $\mathcal{X}^{tr}$ train a generative model $\hat{\theta}_{\mathbf{1}}$. **forall the** $\mathbf{x} \in \mathcal{X}^{tr} \cup \mathcal{X}^{te}$ **do**

2 Given $\hat{\theta}_{\mathbf{1}}$, construct the corresponding discriminative features $\mathbf{s} = (\partial_{\theta_{11}} v(\hat{\theta}_{\mathbf{1}}), \cdots, \partial_{\theta_{1p}} v(\hat{\theta}_{\mathbf{1}}))^T$;

3 $\mathbf{s} \rightarrow \mathcal{S}$;

4 **end**

After proper transformation, the dual form of this problem becomes:

$$\min_{\alpha} \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(\mathbf{x_i}, \mathbf{x_j}), \tag{6.13}$$

$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{\nu m}, \sum_i \alpha_i = 1. \tag{6.14}$$

where $k(\mathbf{x_i}, \mathbf{x_j}) = \Phi(\mathbf{x_i})\Phi(\mathbf{x_j})$ is a kernel function.

Then, the estimated optimal $\mathbf{w}^*$ is obtained using α (which maximize Equation 6.11) as below:

$$\mathbf{w}^* = \sum_i \alpha_i \Phi(\mathbf{x}^{(i)}). \tag{6.15}$$

where $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \Phi(\mathbf{x}^{(i)})\Phi(\mathbf{x}^{(j)})$ is a kernel function and the $\mathbf{w}^*$ becomes the parameters of the output classifier C.

The learning algorithm is summarized in Algorithm 6.4.

6.4.3 Phase 3: Anomaly Detection

The anomaly detection phase requires the following:

- $\mathcal{S}^{te}$: the extracted feature space of the test data set consists of both normal and abnormal sequences.
- C: the constructed classifier based on the training sequences.

- Th_2: the threshold to detect abnormal sequences.

This phase is straightforward, for any sequence $\mathbf{x} \in \mathcal{S}^{te}$, apply the learned classifier C (i.e., $\mathbf{w}^{*\mathbf{T}}\mathbf{f}_{\hat{\theta}}(\mathbf{x}) + b$) and Th_2 to detect anomaly. That is, if $\mathbf{w}^{*\mathbf{T}}\mathbf{f}_{\hat{\theta}}(\mathbf{x}) + b < Th_2$, $\mathbf{x}$ is detected as anomaly and put into the anomaly set $\mathcal{S}^a$, which is the output.

6.5 Experimental Settings

6.5.1 Data Sets

The details of both synthetic and real-world data sets are reported in this section. The synthetic data is used to illustrate the performance of the proposed algorithm without considering the influence of the approximate modeling. This is because all the synthetic data are sampled from generative HMMs and thus can be reasonably modeled as HMMs. In addition, we also use a variety of real-world data sets extracted from different application domains when the behavioral sequences can be approximately modeled as HMMs.

The Synthetic Data

Here we consider a toy example to test the performance of our proposed algorithm. We assume that normal and abnormal sequences are generated from two 2-state Gaussian HMMs (θ_1, θ_{-1}) specified in Table 6.2 respectively ('1' is the label for normal class and '-1' is the label for abnormal class).

Since the two models generating the sequences are very similar (and only have a slight difference in A), the generated sequences are very similar and quite difficult to differentiate by their appearance. Figure 6.1(b) shows two sample sequences from the synthetic data. As can be seen from the chart, these sequences are quite stochastic and how to distinguish them directly is unclear. Thus, this synthetic data set provides a very challenging scenario for one-class mode sequence anomaly detection, because the abnormal sequences

Table 6.2: Parameters of the HMMs Generating the Normal and Abnormal Sequences

	A	B	π
θ_1	$\begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}$	$\Big(\mathcal{N}(0,1), \mathcal{N}(3,1)\Big)$	$\Big(0.5, 0.5\Big)$
θ_{-1}	$\begin{pmatrix} 0.4 & 0.6 \\ 0.4 & 0.6 \end{pmatrix}$	$\Big(\mathcal{N}(0,1), \mathcal{N}(3,1)\Big)$	$\Big(0.5, 0.5\Big)$

can only be differentiated from the normal sequences by their dynamical characteristics that are different in the model generating them. In other words, the abnormal sequences are very similar to the normal sequences. Thus, it is suitable for testing the discriminative power of our proposed framework. The length of each individual sequence is obtained by sampling a uniform pdf in the range of $[\mu_L(1-V/100)\ \mu_L(1+V/100)]$, where μ_L is the sequence's mean length and V is a parameter that refers to as the percentage of variation in the length ($V = 40$ in this chapter). By doing so, we hope to examine the influence of sequence length on the anomaly detection performance. All the given results are averaged over 50 randomly generated data sets.

The Real-world Data

To evaluate the performance of the proposed algorithm in real world, 5 publicly available data sets are used. From the perspective of data types, these data sets can be grouped into two categories: discrete sequences and multi-(uni-)variate time series. From the perspective of data characteristics, the data sets are from different domains of intrusion detection (ID), fault detection (FD), electrocardiogram (ECG) signals, character trajectory (CT) records and Japanese Vowels (JV) speech. The details of the real-world data sets used are given in the following:

ID Data This data set[1] were collected by the University of New Mexico to evaluate the performance of intrusion detection for system calls. The normal sequences consist of sequence of system calls generated in an operating system during the normal operation of a computer program, such as sendmail, ftp, lpr etc. The anomalous sequences consist of sequence of system calls generated when the program was run in an abnormal mode, corresponding to the operation of a hacked computer. A subset of data sets available in the repository is used here, which was processed by the same process mentioned in (Chandola, Cheboli & Kumar 2009).

Fault Detection Data This repository[2] is the basic security module (BSM) audit data, collected from a victim Solaris machine, in the DARPA Lincoln Labs 1998 network simulation data sets. The data is similar to the intrusion detection data described above.

Electrocardiogram (ECG) Data This data set[3] corresponds to an ECG recording for one subject suffering with a particular heart condition. The ECG recording was segmented into short sequences of equal lengths. Sequences that contain any annotation of a heart condition are added to the anomalous set and the remaining sequences form the normal set.

Character Trajectory This data set[4] consists of trajectories captured by a digitizing tablet when writing 20 different characters and each sample is a 3-dimensional time series differentiated and smoothed using a Gaussian kernel. In experiments, we use the sequences of one character as the normal set and use the samples of another character as the abnormal set, giving a total of 19 experiments (each experiment was repeated 10 times).

[1]Available at http://www.cs.unm.edu/∼immsec/systemcalls.htm.
[2]Available at http://www.ll.mit.edu/mission/communications/ist/.
[3]Available at http://www.physionet.org/physiobank/database/edb/.
[4]Available at http://archive.ics.uci.edu/ml/datasets/Character+Trajectories.

Japanese Vowels The data set[5] collects several utterances of nine male speakers producing two Japanese vowels /ae/ successively. 12 dimension linear predictive coding (LPC) cepstrum coefficients have been extracted from each utterance, which forms a 12-dimension time series. In experiments, we use the sequences of one speaker as the normal set and use the samples of another speaker as the abnormal set, giving a total of 8 experiments (each experiment was repeated 10 times).

Table 6.3 summarizes the data sets for experimental evaluation, where D is the dimension of each observation in the sequences, μ_L is the averaged length of the sequences and $|\mathcal{X}_i|$ $(i \in N, A, tr, te)$ is the number of sequences. For each data set, we have done repetitive experiments and report the averaged results of 10 times at least. The general methodology to create the data sets is as the following (Chandola, Cheboli & Kumar 2009): For each data set, a normal data set, $\mathcal{X}^N$, and an anomalous data set $\mathcal{X}^A$ are created. A training data set $\mathcal{X}^{tr}$ is created by randomly sampling a fixed number of sequences from $\mathcal{X}^N$. A test data set $\mathcal{X}^{te}$ is created by randomly sampling a fixed number of normal sequences from $\mathcal{X}^N - X^{tr}$ (i.e., $\mathcal{X}^N$ without the sequences from $\mathcal{X}^{tr}$) and a fixed number of anomalous sequences from $\mathcal{X}^A$.

6.5.2 Comparative Algorithms

We compare two variants of our proposed MDF framework (using linear and Gaussian radial basis SVM as the classifiers in phase 2) with the model-based algorithm, and four baseline methods without learning as following:

- MDF with linear SVM (MDF-SVM), which means a linear SVM is applied as the classifier in phase 2 of the MDF framework.
- MDF with Gaussian radial basis SVM (MDF-SVMrb), which means a non-linear SVM is applied as the classifier in phase 2 of the MDF framework.

[5]Available at http://archive.ics.uci.edu/ml/datasets/Japanese+Vowels.

Table 6.3: The Details of the Real Data Sets

Dataset	ID	FD	ECG	CT	JV
D	discrete	discrete	1	3	12
μ_L	839	143	250	166	16
$\lvert\mathcal{X}_N\rvert$	2030	2000	500	186	30
$\lvert\mathcal{X}_A\rvert$	130	67	50	119-171	30
$\lvert\mathcal{X}_{tr}\rvert$	1030	1000	500	136	10
$\lvert\mathcal{X}_{te}\rvert$	1050	1050	550	60	30

- The Model-based Algorithm (use HMM as the model, as described in Section 6.2.1).
- MDF with k-nearest neighbor classifier (MDF-kNN), which means a lazy classifier kNN is applied directly after phase 1 of the MDF framework without phase 2. In particular, we set $k = 4$, which is suggested by (Chandola, Cheboli & Kumar 2009).
- Oracle Model (ORACLE). This baseline method uses the true model information of both the normal and the abnormal sequences. The classifier is constructed using the Bayes Rule. In particular, for a given sequence X_i, $P(y = 1|X_i; \theta_1, \theta_{-1})$ is calculated. If it is lower than a predefined threshold Th_0 then X_i is detected as anomaly.
- Semi-Oracle Model (Semi-ORACLE). This baseline method uses the true model information of only the normal sequences. The other setting is similar to the ORACLE model.
- Random Model (RANDOM). As indicated by the name, this model predicts the class label for each sequence randomly.

6.5.3 Performance Measures

To evaluate the performance of the above anomaly detection algorithms, we choose the area under receiver operating characteristic curve (AUC) (Han, Kamber & Pei 2011) and a higher AUC usually means a better classification performance. The reason for this choice is the anomaly detection problem in this chapter can be treated as a special case of a binary classification problem, and the AUC is widely accepted for evaluating the classification results summarizing the performance at various threshold settings.

6.6 Experimental Results

6.6.1 Synthetic Data

Figure 6.3 shows the results of the performance comparison of different anomaly detection techniques against different numbers of training sequences. It can be seen that, the number of training sequences does not have significant impact on the performance of the algorithms. This may be because of the sequences are generated by simple synthetic models and can be modeled by the HMMs using relatively small samples. Figure 6.4 shows the results of the performance comparison of different anomaly detection techniques against different mean sequence lengths. As shown in the picture, the algorithms tend to have better performance when the length of sequences increases. This conforms to our intuition that longer sequences have clearer dynamic characteristics to capture, which is very helpful to further anomaly detection. Figure 6.5 shows the results of the performance comparison of different anomaly detection techniques against different number hidden states Q of the HMMs. As can be seen from the chart, the performance of MDF-SVM, MDF-SVMrb and MDF-kNN decreases when the model structure varies. A possible explanation is that improper model structures may generate redundant dimensions in the extracted feature space and degrade the anomaly detection result. Figure 6.6 shows the results of the performance

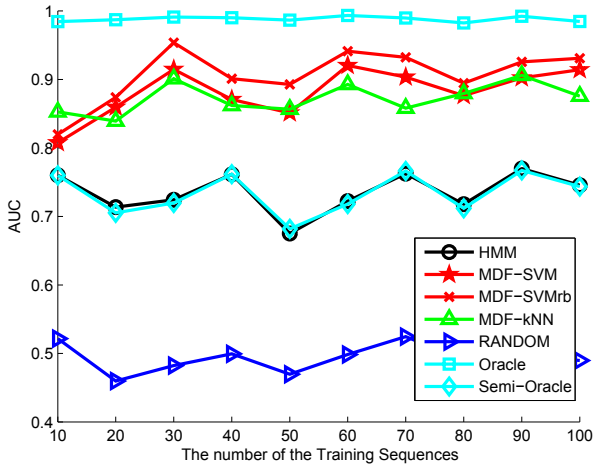


Figure 6.3: AUC Obtained by the Comparative Algorithms against Different Numbers of Training Sequences.

comparison of different anomaly detection techniques against different ratios of the normal and abnormal sequences in a testing data set. It can be clearly seen from the picture that the ratio of the normal and abnormal sequences has little impact on the anomaly detection performance.

To sum up, the proposed MDF-SVM and MDF-SVMrb have the best result (close to ORACLE) consistently in most of different settings, which proves the stability of our proposed framework. This is because the proposed feature extractor could capture enough discriminative information to classify the normal and abnormal data and thus different settings have little impact on the anomaly detection performance. It is also noted that MDF-SVM and MDF-SVMrb generally outperforms MDF-kNN in most cases, which may benefit from their learning process in phase 2 of the framework compared to MDF-kNN. Thus, they are expected to have better performance in the real-world data sets, whose results will be reported in the following.

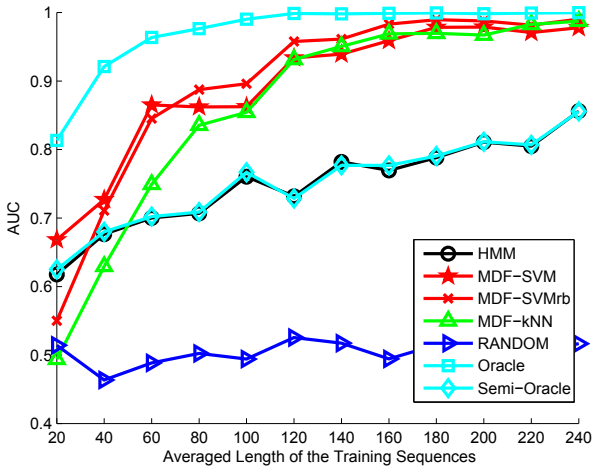


Figure 6.4: AUC Obtained by the Comparative Algorithms against Different Mean Sequence Lengths.

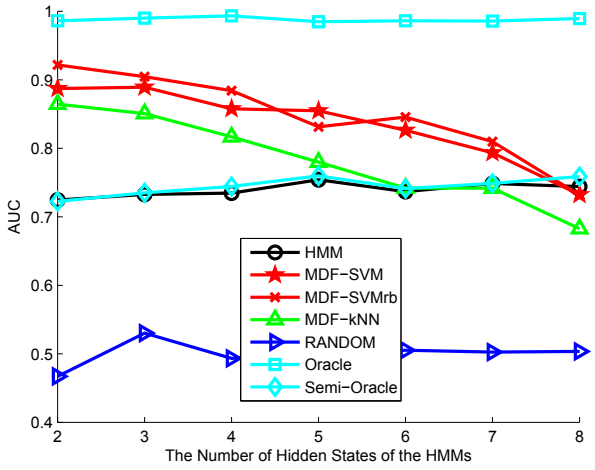


Figure 6.5: AUC Obtained by the Comparative Algorithms against Different Topologies of the HMMs.

6.6.2 Real-world Data

Table 6.4 shows experimental results (averaged AUC value of at least 10 repetitive experiments) on the five real-world data sets, with the comparison of five algorithms. In the table, Q denotes the number of hidden states of the HMMs and the ORACLE and Semi-ORACLE algorithms are excluded since

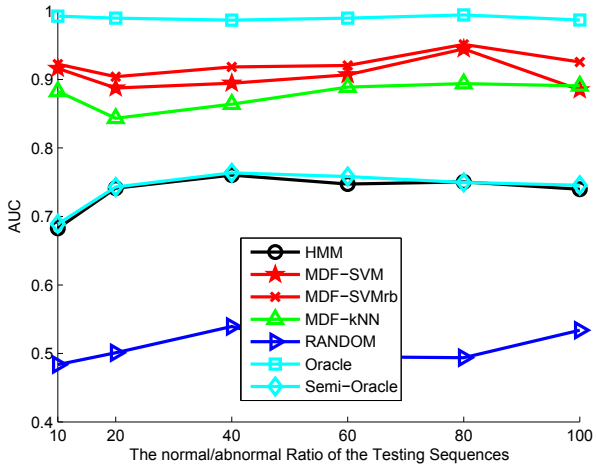


Figure 6.6: AUC Obtained by the Comparative Algorithms against Different Ratios of the Normal and Abnormal Sequences in the Testing Data Set.

we do not know the true parameters of the model in real-world data sets. All in all, the MDF-SVMrb noticeably outperforms the rest of the alternatives. This is because the MDF-SVMrb not only extracts discriminative features but also learns a non-linear decision boundary in the extracted feature space to detect the anomalies, while others may fail to do so. MDF-SVM works very well on some data sets because the normal and abnormal sequences may be linearly seperatable in the MDF space under these cases. A remarkable fact is that the proposed algorithms do not suffer a severe performance loss as the number of hidden states increases. This is because the true models of the data are more complex and our models are relatively simple, which give proper approximations to the true models with no significant difference. This indicates the robustness of the algorithms when the true model is much more complicated. It is also worth to note that the proposed MDF-SVM and MDF-SVMrb generally perform better when the averaged length of the sequences increase, which agrees with the observation from the results obtained with synthetic data. Finally, to further validate the statistical significance of our experiments, we also perform the paired t-test (2-tail) between our proposed methods and other baselines on the experimental results. All the t-test results

Table 6.4: The Experimental Results of the Real Data Sets

Dataset	Q	HMM	MDF-SVM	MDF-SVMrb	MDF-kNN	RANDOM
ID	2	0.94 ± 0.00	0.18 ± 0.18	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.51 ± 0.04
	3	0.94 ± 0.00	0.15 ± 0.09	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.99 \pm 0.00}$	0.48 ± 0.02
	4	0.94 ± 0.00	0.18 ± 0.18	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.51 ± 0.04
FD	2	0.39 ± 0.00	0.53 ± 0.2	$\mathbf{0.91 \pm 0.01}$	$\mathbf{0.91 \pm 0.00}$	0.50 ± 0.06
	3	0.39 ± 0.00	0.4 ± 0.12	$\mathbf{0.92 \pm 0.01}$	$\mathbf{0.92 \pm 0.00}$	0.51 ± 0.05
	4	0.39 ± 0.00	0.58 ± 0.13	$\mathbf{0.93 \pm 0.01}$	0.91 ± 0.00	0.50 ± 0.05
ECG	2	0.27 ± 0.00	$\mathbf{0.67 \pm 0.00}$	$\mathbf{0.67 \pm 0.00}$	0.61 ± 0.00	0.49 ± 0.04
	3	0.28 ± 0.00	$\mathbf{0.64 \pm 0.02}$	$\mathbf{0.64 \pm 0.02}$	0.61 ± 0.01	0.50 ± 0.04
	4	0.28 ± 0.00	$\mathbf{0.65 \pm 0.00}$	$\mathbf{0.65 \pm 0.00}$	0.61 ± 0.00	0.50 ± 0.05
CT	2	0.82 ± 0.2	0.71 ± 0.33	$\mathbf{0.96 \pm 0.04}$	$\mathbf{0.96 \pm 0.04}$	0.50 ± 0.10
	3	0.91 ± 0.1	0.75 ± 0.30	$\mathbf{0.97 \pm 0.03}$	$\mathbf{0.97 \pm 0.03}$	0.52 ± 0.10
	4	0.94 ± 0.07	0.77 ± 0.28	$\mathbf{0.98 \pm 0.06}$	$\mathbf{0.98 \pm 0.03}$	0.51 ± 0.10
JV	2	0.94 ± 0.07	0.95 ± 0.05	$\mathbf{0.96 \pm 0.06}$	0.94 ± 0.06	0.52 ± 0.13
	3	0.92 ± 0.07	$\mathbf{0.95 \pm 0.06}$	$\mathbf{0.95 \pm 0.06}$	$\mathbf{0.95 \pm 0.06}$	0.50 ± 0.15
	4	0.94 ± 0.06	0.95 ± 0.05	$\mathbf{0.96 \pm 0.05}$	0.95 ± 0.04	0.50 ± 0.14

are less than 0.01, which proves the differences of our proposed methods versus other baselines are statistically significant.

In addition, the computational cost of the proposed framework mainly spends on the feature extractor stage and it scales to $O(Q^2TN)$, where Q is the number of hidden states, T is the averaged length of the sequences and N is the number of sequences. Thus, the computational time of the MDF-SVM, MDF-SVMrb and MDF-kNN is very similar but a little higher than the HMM and RANDOM algorithms (the proposed framework, however, has much better anomaly detection performance). This is proved empirically in the experiments and we do not report the details here due to the space limit.

6.7 Summary

This chapter examines a challenging issue of detecting abnormal sequences in an one-class setting and presents a reasonable MDF framework by theoretically analyzing the nature of the problem. To be more specific, the proposed framework is composed of three phases: the generative model-based feature extractor phase, the optimal classifier training phase and the anomaly detection phase. Theoretical analysis has demonstrated that the proposed method leads to a better approximation to the oracle Bayes error (i.e., the anomaly detection performance in this chapter). To evaluate the superiority of our proposed framework, several experiments have been conducted on synthetic data sets. The empirical results show that the proposed framework generally outperforms the other comparative schemes. We also explore a wide range of real-world problems, such as speaker verification and ECG signal detection (i.e., detecting hearts with problematic conditions) and the corresponding experimental results show the effectiveness of our proposed framework.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we have presented a set of ideas and algorithms for learning from heterogeneous data. The key contributions are listed in the below from three aspects. From the perspective of data type, we have explored various types of heterogeneous data with different mixed structures, such as sequential data with different lengths, relational data consisting of the user/user friendship networks and the user/item preference matrix, and miscellaneous data made up of time series and multivariate data. From the perspective of application domains, we have analyzed a wide range of heterogeneous data in different application scenarios. These domain includes web-browsing log analysis, bio-informatics, clinical gait analysis, social recommendation, and anomaly detection. From the perspective of methodologies, not only have we designed various new probabilistic models (i.e., BNs) for modeling the above mentioned data, but also combined the BNs with discriminative classifiers, such as SVM, to achieve more accurate performance for specific data analysis task, such as sequence anomaly detection.

To be more specific, we have examined the problem of characterizing (modeling) a database of sequential behaviors (sequences) that are of different lengths, which is commonly arisen from the area of web-browsing log min-

ing or bio-informatics. Most state-of-the-art models, such as hidden Markov models, consider the non-i.i.d. relationships between the behaviors within each sequence but simply assume these sequences are identically independent and identically distributed. By contrast, we break the i.i.d. assumption on the both the sequence-level and the database-level and have proposed a hierarchically probabilistic model, LDHMM, to capture these relationships in different levels. Through this manner, we expect to have a more comprehensive modeling of the sequential behaviors both globally and locally. The benefits of the proposed LDHMM is demonstrated in comparison with other state-of-the-art models, such as LDA, HMM and HMMV, based on empirical studies on many real-world data sets.

We also study the problem of jointly modeling the *static* and *dynamic* data arisen from the area of clinical gait analysis. The data is heterogeneous since each patient has both the static (i.e., physical examination) data and dynamic (i.e., gait) data and they correlated. Existed algorithms, such as HMM and CRF, can only model the sequential relationship between the data points in the dynamic data. To overcome this, we have proposed a unified probabilistic model, named CSDM, to comprehensively capture the correlated relationships that may exist in the data. We not only consider the correlation between the data points within the dynamic data but also the correlation between the static and dynamic data. Empirical study on both the synthetic and real-world data sets show the effectiveness of the CSDM. The real-world data is extracted from the clinical records of the Royal Children Hospital, Melbourne. One main advantage of the CSDM is that we can extract some interesting knowledge from the model, to give the users an intuition, such as what kinds of static data lead to what kinds of dynamic data.

Another research performed by this thesis is the social recommendation problem, which tries to recommend items, such as movies, songs and interesting stuff, in a web 2.0 socialized environment. In this problem, there are two types of relational data, which is mixed. One is the friendship networks

that describe the interactions between the users; the other is the preference (rating) matrix that record the interactions between the users and the items. Unlike the traditional collaborative filtering methods that only model the preference matrix, we have proposed a unified probabilistic model, JISM, for jointly modeling both the friendship networks and rating matrix. Specifically, we represent each user with a *interest* latent factor and a *social* latent factor, and each item with a *interest* latent factor. The user/user and user/item interactions are assumed to be controlled by the above heterogeneous latent factor space. We use variation EM to estimate the parameters of the JISM and recommend items to users based on the learned JISM. Empirical results on three real-world data sets crawled from the internet show the effectiveness of the proposed JISM, with comparison to other state-of-the-art models. In addition, the JISM can also provide some interesting visualized results, such as the clustering of the items.

The above problems are all solved with purely Bayesian networks. Sometimes, for certain data mining tasks, the predictive power of BNs is not satisfactory since the approximate estimation of the parameters. Thus, we then investigate how to enhance the performance of the sequence anomaly detection, which identifies abnormal sequence in a set of sequences. This problem emerges in many application domains, such as intrusion detection, fault detection and speaker recognition. We theoretically analyze the essence of the problem and propose a three-stage general framework on the basis of the theoretical analysis. Specifically, the first stage extracts discriminative features from the sequences based on a predefined BNs model, which is theoretically proved to have better discriminative power in terms of Bayes error. Then, the second stage uses the extracted features to train a discriminative classifier (i.e., SVM) which is further used to detect the anomaly in the test data set in the third stage. The experimental results on both the synthetic and real-world data sets exhibit the superiority of the proposed framework.

7.2 Future Work

From the high-level aspect, there are two directions valuable to be further explored. Firstly, the dependency structure expressed in Bayesian networks are inherently directional, which limits its modeling power when the dependent structure is not of clear direction. One possible solution to this problem is to utilize Markov networks that can express the undirected dependent structures. Secondly, as mentioned in (Cao 2013), the strong *coupling relationships* are usually embedded in *mixed structured (heterogeneous)* data. This thesis, however, only focus on the modeling of the heterogeneous structures. Thus, one future direction is to consider the complex coupling relationships between entities such as values, attributes, objects and data sets on top of the mixed structures.

From the perspective of each main chapter, here we also sketch some future lines for these specific topics.

In Chapter 3, we assume that the observed sequences can only have one behavior at one time stamp, which is not practical in many application domains. For example, in the field of customer transaction analysis, one customer may buy several items at one time stamp. Thus, one possible future direction is to generalize LDHMMs to cater for the above scenarios. Additionally, it is also interesting to investigate the combination of our model with discriminative classifiers, such as support vector machine (SVM), to further improve the classification performance. This is because, similar to LDA, our model can be naturally seen as a dimensionality reduction method for feature extraction.

One direction for future work of Chapter 4 is to improve the CSDM with semi-supervised learning. Currently the CSDM is learned totally unsupervised, which may generate unexpected results due to its highly stochastic nature. Further collaboration with gait analysis experts may alleviate this problem through manual labeling of some examples. We also plan to collect more real-world data and include all static and dynamic outputs from clinical gait analysis.

One possible further exploration of Chapter 5 is to extend the model for directed friendship networks. This is because many web 2.0 websites now begin to support asymmetric follower/followee friendship, which is directed and different to traditional bidirectional friendship. Another research direction is to integrate more source of data to improve the prediction accuracy. For example, we can add the content description of the items for modeling.

The problem of sequence anomaly detection considered in Chapter 6 is inherently in one-class mode (i.e., only the normal data is available for training). However, in many real-world scenarios, it is unrealistic to obtain data that ideally contains only normal instances. In these situations, the anomaly detection techniques need to be operated in a mixed setting (i.e., the training data contains both normal and anomalous sequences without labels, under the assumption that anomalous sequences are very rare). The extension to a mixed mode is a possible future research direction.

To conclude, non-i.i.d. data provides great treasure of complex relationships, which can be learned and predictive. It is a young and promising field that has endless possible applications on every aspect of people's every day life. The learning tasks presented in this thesis, only scratched the surface of the non-i.i.d. data and focused on learning its heterogeneity. We expect that more research from other perspectives on learning with non-i.i.d. data will emerge and lead to other novel real-world applications.

Appendix A

Appendix for Chapter 3

A.1 Distributions

A.1.1 Dirichlet Distribution

A K-dimensional Dirichlet random variable $\boldsymbol{\theta}$ ($\sum_{i=1}^{K} \theta_i = 1$ and $\theta_i \geq 0$) and has the following form of probability distribution (Kotz, Balakrishnan & Johnson 2000):

$$p(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{K})\alpha_i}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1} \tag{A.1}$$

where the parameter $\boldsymbol{\alpha}$ is a K-dimension vector with components $\alpha_i > 0$, and where $\Gamma(\cdot)$ is the Gamma function.

A.1.2 Multinomial Distribution

A 1-of-V vector multinomial random variable $\boldsymbol{x}$ ($\sum_{i=1}^{V} x_i = 1$ and $x_i \in \{0, 1\}$) and has the following form of probability distribution (Evans, Hastings & Peacock 2000):

$$p(\mathbf{x}; \boldsymbol{\mu}) = \prod_{i=1}^{V} \mu_i^{x_i} \tag{A.2}$$

where the parameter $\boldsymbol{\mu}$ is a V-dimension vector with components $\sum_{i=1}^{K} \mu_i = 1$ and $\mu_i \geq 0$.

A.2 Variational Inference

A.2.1 The FF Form

Here we expand the expression of L for Variational Inference for the FF form. for $q_m(\boldsymbol{\pi}_m; \boldsymbol{\gamma}_m^{(\pi)})$, $q_m(\mathbf{A}_m; \boldsymbol{\gamma}_{m,1:K}^{(A)})$ and $q_m(\mathbf{B}_m; \boldsymbol{\gamma}_{m,1:K}^{(B)})$ are usually assumed to be Dirichlet distributions governed by parameters $\boldsymbol{\gamma}_m^{(\pi)}$, $\boldsymbol{\gamma}_{m,1:K}^{(A)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$. $q_m(\mathbf{z}_{mn}; \boldsymbol{\phi}_{mn})$ is assumed to be multinomial distributions governed by $\boldsymbol{\phi}_{mn}$ $(1 \leq n \leq N_m)$. Then Equation 3.2 can be expanded as follows:

$$
\begin{aligned}
L = \sum_{m=1}^{M} & [E_q[\log p(\boldsymbol{\pi}_m|\boldsymbol{\alpha}^{(\pi)})] + E_q[\log p(\mathbf{A}_m|\boldsymbol{\alpha}_{1:K}^{(A)})] + E_q[\log p(\mathbf{B}_m|\boldsymbol{\beta}_{1:K})] \\
& + E_q[\log p(\mathbf{z}_{m1}|\boldsymbol{\pi}_m)] + E_q[\sum_{n=2}^{N_m} \log p(\mathbf{z}_{mn}|\mathbf{z}_{m,n-1}, \mathbf{A}_m)] \\
& + E_q[\sum_{n=1}^{N_m} \log p(\mathbf{x}_{mn}|\mathbf{z}_{mn}, \mathbf{B}_m)] \\
& - E_q[\log q_m(\boldsymbol{\pi}_m)] - E_q[\log q_m(\mathbf{A}_m)] - E_q[\log q_m(\mathbf{B}_m)] - E_q[\log q_m(\mathbf{Z}_m)]]
\end{aligned}
$$

where

$$
\begin{aligned}
E_q[\log p(\boldsymbol{\pi}_m|\boldsymbol{\alpha}^{(\pi)})] = & \log \Gamma(\sum_{j=1}^{K} \alpha_j^{(\pi)}) \\
& - \sum_{i=1}^{K} \log \Gamma(\alpha_i^{(\pi)}) \\
& + \sum_{i=1}^{K} (\alpha_i^{(\pi)} - 1)(\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\sum_{j=1}^{K} \gamma_{mj}^{(\pi)}))
\end{aligned}
$$

$$
\begin{aligned}
E_q[\log p(\mathbf{A}_m|\boldsymbol{\alpha}_{1:K}^{(A)})] = & \sum_{i=1}^{K} [\log \Gamma(\sum_{j=1}^{K} \alpha_{ij}^{(A)}) \\
& - \sum_{k=1}^{K} \log \Gamma(\alpha_{ik}^{(A)}) \\
& + \sum_{k=1}^{K} (\alpha_{ik}^{(A)} - 1)(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K} \gamma_{mij}^{(A)}))]
\end{aligned}
$$

$$
\begin{aligned}
E_q[\log p(\mathbf{B}_m|\boldsymbol{\beta}_{1:K})] = &\sum_{i=1}^{K}[\log\Gamma(\sum_{j=1}^{K}\beta_{ij}) \\
&-\sum_{k=1}^{K}\log\Gamma(\beta_{ik}) \\
&+\sum_{k=1}^{K}(\beta_{ik}-1)(\Psi(\gamma_{mik}^{(B)})-\Psi(\sum_{j=1}^{K}\gamma_{mij}^{(B)}))]
\end{aligned}
$$

$$
E_q[\log p(\mathbf{z}_{m1}|\boldsymbol{\pi}_m)] = \sum_{i=1}^{K}\phi_{m1i}(\Psi(\gamma_{mi}^{(\pi)})-\Psi(\sum_{j=1}^{K}\gamma_{mj}^{(\pi)})) \tag{A.3}
$$

$$
E_q[\sum_{n=2}^{N_m}\log p(\mathbf{z}_{mn}|\mathbf{z}_{m,n-1},\mathbf{A}_m)] = \sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{k=1}^{K}\phi_{mn-1,i}\phi_{mnk}(\Psi(\gamma_{mik}^{(A)})-\Psi(\sum_{j=1}^{K}\gamma_{mij}^{(A)})) \tag{A.4}
$$

$$
E_q[\sum_{n=1}^{N_m}\log p(\mathbf{x}_{mn}|\mathbf{z}_{mn},\mathbf{B}_m)] = \sum_{n=1}^{N_m}\sum_{i=1}^{K}\sum_{j=1}^{V}\phi_{mni}(x_{mnj}(\Psi(\gamma_{mij}^{(B)})-\Psi(\sum_{v=1}^{V}\gamma_{miv}^{(B)}))) \tag{A.5}
$$

$$
\begin{aligned}
E_q[\log q_m(\boldsymbol{\pi}_m)] = &\log\Gamma(\sum_{j=1}^{K}\gamma_{mj}^{(\pi)}) \\
&-\sum_{i=1}^{K}\log\Gamma(\gamma_{mi}^{(\pi)}) \\
&+\sum_{i=1}^{K}(\gamma_{mi}^{(\pi)}-1)(\Psi(\gamma_{mi}^{(\pi)})-\Psi(\sum_{j=1}^{K}\gamma_{mj}^{(\pi)}))
\end{aligned}
$$

$$
\begin{aligned}
E_q[\log q_m(\mathbf{A}_m)] = &\sum_{i=1}^{K}[\log\Gamma(\sum_{j=1}^{K}\gamma_{mij}^{(A)}) \\
&-\sum_{k=1}^{K}\log\Gamma(\gamma_{mik}^{(A)}) \\
&+\sum_{k=1}^{K}(\gamma_{mik}^{(A)}-1)(\Psi(\gamma_{mik}^{(A)})-\Psi(\sum_{j=1}^{K}\gamma_{mij}^{(A)}))]
\end{aligned}
$$

$$
\begin{aligned}
E_q[\log q_m(\mathbf{B}_m)] = &\sum_{i=1}^{K}[\log\Gamma(\sum_{j=1}^{V}\gamma_{mij}^{(B)}) \\
&- \sum_{v=1}^{V}\log\Gamma(\gamma_{miv}^{(B)}) \\
&+ \sum_{j=1}^{V}(\gamma_{mij}^{(B)}-1)(\Psi(\gamma_{mij}^{(B)}) - \Psi(\sum_{v=1}^{V}\gamma_{miv}^{(B)}))] \\
E_q[\log q_m(\mathbf{Z}_m)] = &\sum_{n=1}^{N_m}\sum_{i=1}^{K}\phi_{mni}\log\phi_{mni}
\end{aligned} \tag{A.6}
$$

Fixed $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, Update $\boldsymbol{\phi}_{m,1:N_m}$ As a functional of ϕ_{m1i} and add Lagrange multipliers:

$$
\begin{aligned}
L(\boldsymbol{\phi}_{m1}) = &\sum_{i=1}^{K}\phi_{m1i}(\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\sum_{j=1}^{K}\gamma_{mj}^{(\pi)})) \\
&+ \sum_{i=1}^{K}\sum_{k=1}^{K}\phi_{m1i}\phi_{m2k}(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K}\gamma_{mij}^{(A)})) \\
&+ \sum_{i=1}^{K}\sum_{j=1}^{V}\phi_{m1i}(x_{m1j}(\Psi(\gamma_{mij}^{(B)}) - \Psi(\sum_{v=1}^{V}\gamma_{miv}^{(B)}))) \\
&- \sum_{i=1}^{K}\phi_{m1i}\log\phi_{m1i} + \lambda(\sum_{i=1}^{K}\phi_{m1i} - 1) + const
\end{aligned}
$$

Setting the derivative to zero yields the maximizing value of the variational parameter ϕ_{m1i} as Equation 3.5. Similarly, the updated equation for ϕ_{mni} $(2 \le n \le N_m - 1)$ and ϕ_{mN_mi} can be obtained as Equation 3.6 and 3.7.

Use similar technique as above, we can fix $\boldsymbol{\phi}_{m,1:N_m}$, $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, $\boldsymbol{\gamma}_{m,1:K}^{(B)}$, update $\boldsymbol{\gamma}_m^{(\pi)}$ as Equation 3.8; fix $\boldsymbol{\phi}_{m,1:N_m}$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$, update $\boldsymbol{\gamma}_{m,1:K}^{(A)}$ as Equation 3.9; fix $\boldsymbol{\phi}_{m,1:N_m}$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, estimate $\boldsymbol{\gamma}_{m,1:K}^{(B)}$ as Equation 3.10.

A.2.2 The PF From

The expansion of L for Variational Inference for the PF form is similar to the FF form except for changing ϕ_{mni} to γ_{mni} $(1 \le n \le N_m)$ in Equation A.3 and

A.5, and $\phi_{m,n-1,i}\phi_{mnk}$ to $\xi_{m,n-1,i,n,k}$ $(2 \leq n \leq N_m)$ in Equation A.4, where

$$\gamma_{mnk} = p(z_{mnk}|\mathbf{X}_m, \boldsymbol{\gamma}_m^{(\pi)}, \boldsymbol{\gamma}_{m,1:K}^{(B)}, \boldsymbol{\gamma}_{m,1:K}^{(A)})$$
$$\xi_{m,n-1,j,n,k} = p(z_{m,n-1,j}, z_{mnk}|\mathbf{X}_m, \boldsymbol{\gamma}_m^{(\pi)}, \boldsymbol{\gamma}_{m,1:K}^{(B)}, \boldsymbol{\gamma}_{m,1:K}^{(A)})$$

Then, the variational inference can be done as following:

Fixed $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, Update $q_m(\mathbf{Z}_m)$ As a functional of $q(\mathbf{z})$ the lower bound of log-likelihood can be expressed as follows:

$$\begin{aligned}
L(q_m(\mathbf{Z}_m)) = &\sum_{i=1}^{K} \gamma_{m1i}(\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\sum_{j=1}^{K} \gamma_{mj}^{(\pi)})) \\
&+ \sum_{n=2}^{N_m}\sum_{i=1}^{K}\sum_{k=1}^{K} \xi_{m,n-1,i,n,k})(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K} \gamma_{mij}^{(A)})) \\
&+ \sum_{n=1}^{N_m}\sum_{i=1}^{K}\sum_{j=1}^{V} \gamma_{mni}(x_{mnj}(\Psi(\gamma_{mij}^{(B)}) - \Psi(\sum_{v=1}^{V} \gamma_{miv}^{(B)}))) \\
&+ const
\end{aligned}$$

Now, defining

$$\begin{aligned}
\boldsymbol{\pi}_m^* \equiv &\quad \exp(\Psi(\gamma_{mi}^{(\pi)}) - \Psi(\sum_{j=1}^{K} \gamma_{mj}^{(\pi)})) \\
\mathbf{A}_m^* \equiv &\quad \exp(\Psi(\gamma_{mik}^{(A)}) - \Psi(\sum_{j=1}^{K} \gamma_{mij}^{(A)})) \\
\mathbf{B}_m^* \equiv &\quad \exp(\Psi(\gamma_{mij}^{(B)}) - \Psi(\sum_{v=1}^{V} \gamma_{miv}^{(B)})))
\end{aligned}$$

The above form of L is similar to the log-likelihood object function of standard HMMs (MacKay 1997) and the relevant posteriors $\boldsymbol{\gamma}$, $\boldsymbol{\xi}$ can be calculated by the forward-backward (FB) algorithm (Rabiner 1990), which will be briefly reviewed in the following.

Here define the following auxiliary variables $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$ $(1 \leq m \leq M, 1 \leq n \leq N_m, 2 \leq n' \leq N_m, 1 \leq j \leq K, 1 \leq k \leq K$ and $\boldsymbol{\theta}_m^* = \{\boldsymbol{\pi}_m^*, \mathbf{A}_m^*, \mathbf{B}_m^*\})$:

$$\alpha'_{mnk} = p(\mathbf{x}_{m1}, \cdots, \mathbf{x}_{mn}, z_{mnk}|\boldsymbol{\theta}_m^*)$$
$$\beta'_{mnk} = p(\mathbf{x}_{m,n+1}, \cdots, \mathbf{x}_{mN}|z_{mnk}, \boldsymbol{\theta}_m^*)$$

Algorithm A.1: ForwardBackward()

input : An initial setting for the parameters $\boldsymbol{\theta}_m^*$

output: Inferred posterior distributions $\boldsymbol{\gamma}$, $\boldsymbol{\xi}$

`/* Calculation of` $\boldsymbol{\alpha}'$, $\boldsymbol{\beta}'$ `*/`

`// Forward;`

$\alpha'_{m1k} = \pi^*_{mk} p(\mathbf{x}_{m1}; \mathbf{b}_k)$ for k;

1 **for** *n=1* **to** *N-1* **do** `// Induction`

2 | **for** *k=1* **to** K **do**

3 | | $\alpha'_{m,n+1,k} = \sum_{j=1}^{K} \alpha'_{mnj} a^*_{jk} p(\mathbf{x}_{m,n+1}; \mathbf{b}_k)$;

4 | **end**

5 **end**

`// Backward;`

$\beta'_{mNk} = 1$ for all k;

6 **for** *n=N-1* **to** *1* **do** `// Induction`

7 | **for** *j=1* **to** K **do**

8 | | $\beta'_{mnk} = \sum_{j=1}^{K} a^*_{jk} p(\mathbf{x}_{m,n+1}; \mathbf{b}_k) \beta'_{m,n+1,j}$;

9 | **end**

10 **end**

`/* Calculation of` $\boldsymbol{\gamma}$, $\boldsymbol{\xi}$ `*/`

11 $p(\mathbf{X}_m|\boldsymbol{\theta}^*_m) = \sum_{k=1}^{K} \alpha_{sgmNk}$;

12 **for** *n=1* **to** N **do**

13 | $\gamma_{mnk} = \frac{\alpha'_{mnk}\beta'_{mnk}}{p(\mathbf{X}_m|\boldsymbol{\theta}^*_m)}$;

14 | $\xi_{m,n-1,j,n,k} = \frac{\alpha'_{m,n-1,k} p(\mathbf{x}_{mn}; \mathbf{b}_k) a^*_{jk} \beta'_{mnk}}{p(\mathbf{X}_m|\boldsymbol{\theta}^*_m)}$ $(n > 2)$;

15 **end**

Then FB algorithm can be summarized in Algorithm A.1. Specifically, line 1-5 calculate the forward variables $\boldsymbol{\alpha}'$, while line 6-10 calculate the backward variables $\boldsymbol{\beta}'$. Then line11-15 calculate the value of each element of the posteriors $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ on the basis of the $\boldsymbol{\alpha}'$, $\boldsymbol{\beta}'$ and $\boldsymbol{\theta}^*_m$.

Use similar techniques described in Appendix A.2.1, we can fix $q_m(\mathbf{Z}_m)$,

$\boldsymbol{\gamma}_{m,1:K}^{(A)}$, $\boldsymbol{\gamma}_{m,1:K}^{(B)}$, update $\boldsymbol{\gamma}_m^{(\pi)}$ as Equation 3.11; fix $q_m(\mathbf{Z}_m)$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(B)}$, update $\boldsymbol{\gamma}_{m,1:K}^{(A)}$ as Equation 3.12; fix $q_m(\mathbf{Z}_m)$, $\boldsymbol{\gamma}_m^{(\pi)}$ and $\boldsymbol{\gamma}_{m,1:K}^{(A)}$, Infer $\boldsymbol{\gamma}_{m,1:K}^{(B)}$ as Equation 3.13.

Appendix B

Appendix for Chapter 4

B.1 The Phases of a Gait Cycle

To describe the processes that occur during walking then it is useful to divide the gait cycle into a number of phases. The simplest such division is to divide the cycle for a given limb into the stance phase, when the foot is in contact with the floor, and the swing phase, when it is not. In healthy walking at comfortable speed this happens about 60% into the gait cycle (some studies suggest 62is a closer estimate). The point at which stance ends is *foot-off* (often referred to as toe-off). To develop this scheme to further sub-divide the gait cycle it is possible to depict what is happening to the other leg at the same time. If the walking pattern is symmetrical then the *opposite foot contact* will occur half-way through the gait cycle. Opposite foot off (from the preceding gait cycle) precedes this by the duration of the opposite swing phase (approx 40% of the cycle in normal walking). This subdivides stance into first double support (from foot contact to opposite foot off), single support (from opposite foot off to opposite foot contact) and second double support (from opposite foot contact to foot off).

According to (Perry & Davids 1992), it suggested the definition of additional phases. As shown in Figure B.1, She saw *initial contact* as a separate phase occurring over the first 2% of the gait cycle and named the rest of first

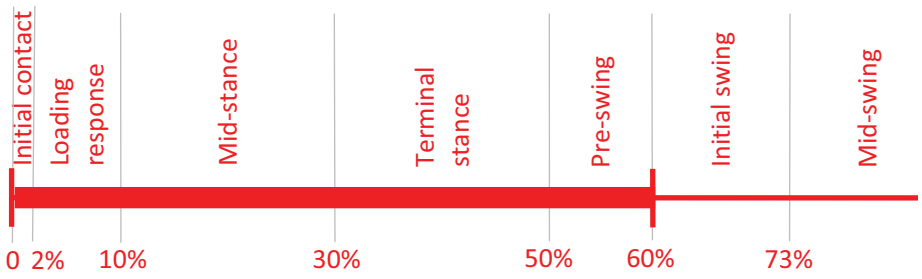


Figure B.1: The Phases of a Gait Cycle.

double support *loading response*. Single support was divided equally into *mid-stance* and *terminal stance* and second double support called *pre-swing*. The swing phase was then divided equally into *initial*, *mid-* and *late swing*.

B.2 The Full Decision tree

1. if $ir\_r < 57$ then node 2 elseif $ir\_r >= 57$ then node 3 else 2
2. class = 2
3. if $er\_r < 21.5$ then node 4 elseif $er\_r >= 21.5$ then node 5 else 1
4. if $a\_l < 22.5$ then node 6 elseif $a\_l >= 22.5$ then node 7 else 1
5. if $ir\_l < 54$ then node 8 elseif $ir\_l >= 54$ then node 9 else 2
6. class = 1
7. if $er\_l < 29$ then node 10 elseif $er\_l >= 29$ then node 11 else 3
8. class = 2
9. if $a\_r < 23.5$ then node 12 elseif $a\_r >= 23.5$ then node 13 else 4
10. if $er\_r < 3$ then node 14 elseif $er\_r >= 3$ then node 15 else 3
11. class = 4

12. class = 2
13. if $h\_r < 1.5$ then node 16 elseif $h\_r >= 1.5$ then node 17 else 4
14. class = 1
15. class = 3
16. class = 3
17. if $h\_r < 2.5$ then node 18 elseif $h\_r >= 2.5$ then node 19 else 4
18. class = 1
19. if $er\_l < 31$ then node 20 elseif $er\_l >= 31$ then node 21 else 4
20. if $h\_r < 4.25$ then node 22 elseif $h\_r >= 4.25$ then node 23 else 4
21. if $ir\_r < 68.5$ then node 24 elseif $ir\_r >= 68.5$ then node 25 else 1
22. if $a\_r < 34$ then node 26 elseif $a\_r >= 34$ then node 27 else 2
23. if $er\_r < 40.5$ then node 28 elseif $er\_r >= 40.5$ then node 29 else 4
24. class = 3
25. class = 1
26. if $ir\_l < 60.5$ then node 30 elseif $ir\_l >= 60.5$ then node 31 else 2
27. class = 4
28. class = 4
29. class = 2
30. class = 1
31. class = 2

Appendix C

Appendix for Chapter 5

C.1 Distributions

C.1.1 Multivariate Gaussian Distribution

The multivariate normal distribution of a K-dimensional random vector $\mathbf{x} = [x_1, x_2, \cdots, x_K]$ can be written in the following format:

$$(2\pi)^{-\frac{k}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{C.1}$$

where $\boldsymbol{\mu}$ is K-dimensional mean vector and $\boldsymbol{\Sigma}$ is $K \times K$ covariance matrix.

C.1.2 Bernoulli distribution

The Bernoulli distribution, named after Swiss scientist Jacob Bernoulli, of a discrete random variable x ($x \in \{0, 1\}$) can be written in the format:

$$p^x(1-p)^{1-x} \tag{C.2}$$

where p denotes the probability of x's taking value of 1.

C.2 Proofs Related to the Lower Bound of the Log-likelihood

Proof 1 (Proof of the Lemma 5.4.1.1)

$$\begin{aligned}\log p(R,E|\mathcal{P}) =& \log \int\int\int p(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N},R,E|\mathcal{P})d\mathbf{u}_{1:N}d\mathbf{v}_{1:M}d\mathbf{w}_{1:N} \\ =& \log \int\int\int \frac{p(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N},R,E|\mathcal{P})q(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N})}{q(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N})} \\ & d\mathbf{u}_{1:N}d\mathbf{v}_{1:M}d\mathbf{w}_{1:N} \\ =& \log(E_q[\frac{p(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N},R,E|\mathcal{P})}{q(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N})}])\end{aligned}$$

Apply Jensen's inequality (Blei et al. 2003) on the right part of the above equation, then

$$\begin{aligned}\log p(R,E|\mathcal{P}) \geq& E_q[\log \frac{p(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N},R,E|\mathcal{P})}{q(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N})}] \\ =& E_q[\log p(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N},R,E|\mathcal{P})] - E_q[q(\mathbf{u}_{1:N},\mathbf{v}_{1:M},\mathbf{w}_{1:N}|\mathcal{P}')]\end{aligned}$$

Proof 2 (Proof of the Corollary 5.4.1.1) *According to the graphical model of the JISM, we can expand the L_0 as Equation 5.3.*

Proof 3 (Proof of the Lemma 5.4.1.2)

$$
\begin{aligned}
&E_q[\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\log p(R_{ij}|\mathbf{u}_i^T\mathbf{v}_j+\mathbf{w}_i^T\mathbf{B}\mathbf{v}_j,\tau)]\\
=&(-\frac{K}{2}\log 2\pi-\frac{1}{2}\log|\tau^2|)\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\\
&-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}E_q[(R_{ij}-\boldsymbol{u}_i^T\boldsymbol{v}_j-\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{v}_j)^T\frac{1}{\tau^2}(R_{ij}-\boldsymbol{u}_i^T\boldsymbol{v}_j-\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{v}_j)]\\
=&(-\frac{K}{2}\log 2\pi-\frac{1}{2}\log|\tau^2|)\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\\
&-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\frac{1}{\tau^2}E_q[R_{ij}^2-2R_{ij}\boldsymbol{u}_i^T\boldsymbol{v}_j-2R_{ij}\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{v}_j\\
&+\boldsymbol{u}_i^T\boldsymbol{v}_j\boldsymbol{v}_j^T\boldsymbol{u}_i+\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{v}_j\boldsymbol{v}_j^T\mathbf{B}\boldsymbol{w}_i\\
&+\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{v}_j\boldsymbol{v}_j^T\boldsymbol{u}_i+\boldsymbol{u}_i^T\boldsymbol{v}_j\boldsymbol{v}_j^T\mathbf{B}^T\boldsymbol{w}_i]
\end{aligned}
\tag{C.3}
$$

Since

$$
\begin{aligned}
&E_q[\boldsymbol{u}_i^T\boldsymbol{v}_j\boldsymbol{v}_j^T\boldsymbol{u}_i]\\
=&\int\boldsymbol{u}_i^TE_q[\boldsymbol{v}_j\boldsymbol{v}_j^T]\boldsymbol{u}_i d\boldsymbol{u}_i\\
=&E_q[\boldsymbol{u}_i^T(diag(\boldsymbol{\nu}_{2j}^2)+\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T)\boldsymbol{u}_i]\\
=&E_q[\boldsymbol{u}_i^Tdiag(\boldsymbol{\nu}_{2j}^2\boldsymbol{u}_i+\boldsymbol{u}_i^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{u}_i]\\
=&tr(diag(\boldsymbol{\nu}_{2j}^2)diag(\boldsymbol{\nu}_{1i}^2))+\boldsymbol{\lambda}_{1i}^Tdiag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}\\
&+tr(\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^Tdiag(\boldsymbol{\nu}_{1i}^2))+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i}
\end{aligned}
\tag{C.4}
$$

where

$$
\begin{aligned}
&tr(\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^Tdiag(\boldsymbol{\nu}_{1i}^2))\\
=&tr(\boldsymbol{\lambda}_{2j}^Tdiag(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{2j})\\
=&\boldsymbol{\lambda}_{2j}^Tdiag(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{2j}
\end{aligned}
\tag{C.5}
$$

Thus,

$$
\begin{aligned}
&E_q[\boldsymbol{u}_i^T\boldsymbol{v}_j\boldsymbol{v}_j^T\boldsymbol{u}_i]\\
=&tr(diag(\boldsymbol{\nu}_{1i}^2)diag(\boldsymbol{\nu}_{2j}^2))+\boldsymbol{\lambda}_{1i}^Tdiag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{2j}^Tdiag(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i}
\end{aligned}
\tag{C.6}
$$

Similarly,

$$\begin{aligned} E_q[\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{v}_j\boldsymbol{v}_j^T\mathbf{B}^T\boldsymbol{w}_i] =& tr(diag(\boldsymbol{\nu}_{3i}^2)(\mathbf{B}diag\boldsymbol{\nu}_{2j}^2\mathbf{B}^T)) + \boldsymbol{\lambda}_{3i}^T\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T\boldsymbol{\lambda}_{3i} \\ &+ \boldsymbol{\lambda}_{2j}^T\mathbf{B}^T diag(\boldsymbol{\nu}_{3i}^2)\mathbf{B}\boldsymbol{\lambda}_{2j} + \boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T\boldsymbol{\lambda}_{3i} \end{aligned} \tag{C.7}$$

$$E_q[\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{v}_j\boldsymbol{v}_j^T\boldsymbol{u}_i] = \boldsymbol{\lambda}_{3i}^T\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i} + \boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i} \tag{C.8}$$

$$E_q[\boldsymbol{u}_i^T\boldsymbol{v}_j\boldsymbol{v}_j^T\mathbf{B}^T\boldsymbol{w}_i] = \boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T\boldsymbol{\lambda}_{3i} + \boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T\boldsymbol{\lambda}_{3i} \tag{C.9}$$

Substitute the variables in Equation C.3 with Equation C.6 and C.9, we can obtain Equation 5.4.

Proof 4 (Proof of the Lemma 5.4.1.3) *Since* $p(E_{ii'}|\mathbf{u}_i^T\mathbf{u}_{i'}+\mathbf{w}_i^T\mathbf{w}_{i'},\mathbf{t})] = \sigma(t_1\mathbf{u}_i^T\mathbf{u}_{i'} + t_2\mathbf{w}_i^T\mathbf{w}_{i'} + t_3)$*, where* σ *is the logistic function and* $\sigma(x) = \frac{1}{1+exp(-x)}$.

Then according to (Jaakkola & Jordan 1997), $\sigma(x) \geq \sigma(x)exp(\frac{x-\xi}{2} + g(\xi)(x^2-\xi^2))$ *(where* $g(\xi) = (\frac{1}{2} - \sigma(\xi))/2\xi$*), we obtain:*

$$\begin{aligned} & E_q[\sum_{i=1}^{N}\sum_{i'=1}^{N}\delta_{ii'}\log p(E_{ii'}|\mathbf{u}_i^T\mathbf{u}_{i'} + \mathbf{w}_i^T\mathbf{w}_{i'},\mathbf{t})] \\ \geq & E_q[\sum_{i=1}^{N}\sum_{i'=1}^{N}\delta_{ii'}[\log\sigma(\xi_{ii'}) + \frac{1}{2}((t_1\mathbf{u}_i^T\mathbf{u}_{i'} + t_2\mathbf{w}_i^T\mathbf{w}_{i'} + t_3) - \xi_{ii'}) \\ & + g(\xi_{ii'})(t_1\mathbf{u}_i^T\mathbf{u}_{i'} + t_2\mathbf{w}_i^T\mathbf{w}_{i'} + t_3)^2 - g(\xi_{ii'})\xi_{ii'}^2]] \end{aligned} \tag{C.10}$$

Let

$$\begin{aligned} f_{51} =& E_q[t_1\mathbf{u}_i^T\mathbf{u}_{i'} + t_2\mathbf{w}_i^T\mathbf{w}_{i'} + t_3] \\ =& t_1\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'} + t_2\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'} + t_3 \end{aligned} \tag{C.11}$$

and

$$\begin{aligned} f_{52} =& E_q[(t_1\mathbf{u}_i^T\mathbf{u}_{i'} + t_2\mathbf{w}_i^T\mathbf{w}_{i'} + t_3)^2] \\ =& t_1^2(tr(diag(\boldsymbol{\nu}_{1i}^2)diag(\boldsymbol{\nu}_{1i'}^2)) + \boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{1i'}^2)\boldsymbol{\lambda}_{1i} \\ &+ \boldsymbol{\lambda}_{1i'}^T diag(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{1i'} + \boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{1i'}^T\boldsymbol{\lambda}_{1i}) \\ &+ t_2^2(tr(diag(\boldsymbol{\nu}_{3i}^2)diag(\boldsymbol{\nu}_{3i'}^2)) + \boldsymbol{\lambda}_{3i}^T diag(\boldsymbol{\nu}_{3i'}^2)\boldsymbol{\lambda}_{3i} \\ &+ \boldsymbol{\lambda}_{3i'}^T diag(\boldsymbol{\nu}_{3i}^2)\boldsymbol{\lambda}_{3i'} + \boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}\boldsymbol{\lambda}_{3i'}^T\boldsymbol{\lambda}_{3i}) \\ &+ t_3^2 + 2t_1t_2\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'} + 2t_1t_3\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'} + 2t_2t_3\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'} \end{aligned} \tag{C.12}$$

Substitute the variables in the right part of Equation C.10 with Equation C.11 and C.12, we can obtain Equation 5.5.

Proof 5 (Proof of the Theorem 5.4.1.1)

$$\begin{aligned} & E_q[\sum_{i=1}^{N}\log p(\mathbf{u}_i|\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)] \\ =& -\frac{KN}{2}\ln 2\pi - \frac{N}{2}\ln|\boldsymbol{\Sigma}_1| - \frac{1}{2}\sum_{i=1}^{N}E_q[(\mathbf{u}_{1i}-\boldsymbol{\mu}_1)^T\Sigma_1^{-1}(\mathbf{u}_{1i}-\boldsymbol{\mu}_1)] \\ =& -\frac{KN}{2}\ln 2\pi - \frac{N}{2}\ln|\boldsymbol{\Sigma}_1| \\ & -\frac{1}{2}\sum_{i=1}^{N}[tr(diag(\boldsymbol{\nu}_{1i}^2)\Sigma_1^{-1}) + (\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1)^T\Sigma_1^{-1}(\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1)] \\ =& f_1 \end{aligned} \tag{C.13}$$

Similarly, we can obtain $E_q[\sum_{j=1}^{M}\log p(\mathbf{v}_j|\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)]$ *equals* f_2 *in Equation 5.8,* $E_q[\sum_{i=1}^{N}\log p(\mathbf{w}_i|\boldsymbol{\mu}_3,\boldsymbol{\Sigma}_3)]$ *equals* f_3 *in Equation 5.9,* $E_q[\sum_{i=1}^{N}\log q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, diag(\boldsymbol{\nu}_{1i}^2))]$ *equals* $f_1^{'}$ *in Equation 5.10,* $E_q[\sum_{j=1}^{M}\log q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j}, diag(\boldsymbol{\nu}_{2j}^2))]$ *equals* $f_2^{'}$ *in Equation 5.11 and* $E_q[\sum_{i=1}^{N}\log q(\mathbf{w}_i|\boldsymbol{\lambda}_{3i}, diag(\boldsymbol{\nu}_{3i}^2))]$ *equals* $f_3^{'}$ *in Equation 5.12.*

Substitute the variables in the right part of Equation 5.3 with Equation 5.7-5.9 and 5.10-5.12, and Bound the variables in the right part of Equation 5.3 with Equation 5.5 and 5.10, we can obtain Equation 5.6.

C.3 Proofs Related to the E and M Steps

Proof 6 (Proof of the Proposition 5.4.1.1)

$$\begin{aligned}
L(\boldsymbol{\lambda}_{1i}) =& -\frac{1}{2}(\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1) \\
&-\frac{1}{2\tau^2}\sum_{j=1}^{M}\delta_{ij}(-2R_{ij}\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}+\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i} \\
&+\boldsymbol{\lambda}_{3i}^T\mathbf{B}diag(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{3i}^T\mathbf{B}\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)\mathbf{B}^T\boldsymbol{\lambda}_{3i} \\
&+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T\mathbf{B}^T\boldsymbol{\lambda}_{3i}) \\
&+\sum_{i'=1,\neq i}^{N}\delta_{ii'}(\frac{1}{2}t_1\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}+g(\xi_{ii'})t_1^2(\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{1i'}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{1i'}^T\boldsymbol{\lambda}_{1i}) \\
&+2g(\xi_{ii'})t_1t_2\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'}+2g(\xi_{ii'})t_1t_3\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'})
\end{aligned} \tag{C.14}$$

$$\begin{aligned}
\frac{dL}{d\boldsymbol{\lambda}_{1i}} =& -(\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1} \\
&-\frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(-R_{ij}\boldsymbol{\lambda}_{2j}^T+\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{2j}^2)+\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T) \\
&+\boldsymbol{\lambda}_{3i}^T\mathbf{B}(diag(\boldsymbol{\nu}_{2j}^2)+\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T)) \\
&+\sum_{i'=1,\neq i}^{N}\delta_{ii'}(\frac{1}{2}t_1\boldsymbol{\lambda}_{1i'}^T+g(\xi_{ii'})t_1^2(2\boldsymbol{\lambda}_{1i}^T diag(\boldsymbol{\nu}_{1i'}^2)+2\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{1i'}^T) \\
&+2g(\xi_{ii'})t_1t_2(\boldsymbol{\lambda}_{1i'}\boldsymbol{\lambda}_{3i}^T\boldsymbol{\lambda}_{3i'})^T+2g(\xi_{ii'})t_1t_3\boldsymbol{\lambda}_{1i'}^T)
\end{aligned} \tag{C.15}$$

Set the above derivative to zero, we can obtain Equation 5.18.

Similarly, we can prove Proposition 5.14-5.19 and we omit the details here.

Proof 7 (Proof of the Proposition 5.4.1.2)

$$L(\boldsymbol{\mu}_1) = -\frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\lambda}_{1i}-\boldsymbol{\mu}_1) \tag{C.16}$$

$$\frac{dL}{d\boldsymbol{\mu}_1} = -\sum_{i=1}^{N}(\boldsymbol{\lambda}_{1i} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1} \tag{C.17}$$

Set the above derivative to zero, we can obtain Equation 5.20.

Similarly, we can prove Equation 5.21-5.30 and we omit the details here.

Appendix D

Appendix for Chapter 6

D.1 Proof of the Transformation

Lemma D.1.0.1 *$P(y=1|\mathbf{x};\theta^*) < \frac{1}{2}$ is equivalent to $P_{\theta_1^*}(\mathbf{x}) < Th_1 \cdot P_{\theta_{-1}^*}(\mathbf{x})$.*

Proof: According to Bayes' theorem:

$$P(y=1|\mathbf{x},\theta^*) = \frac{P_{\theta_1^*}(\mathbf{x})P(y=1)}{\sum_y P_{\theta_y^*}(\mathbf{x})P(y)} < \frac{1}{2} \tag{D.1}$$

After proper transformation, the above formulation becomes:

$$\frac{P_{\theta_1^*}(\mathbf{x})}{P_{\theta_{-1}^*}(\mathbf{x})} < \frac{P(y=-1)}{P(y=1)} = Th_1 \tag{D.2}$$

$$P_{\theta_1^*}(\mathbf{x}) < Th_1 \cdot P_{\theta_{-1}^*}(\mathbf{x}) \tag{D.3}$$

D.2 Proof of the Approximately Optimal Feature Extractor

Lemma D.2.0.1 *The approximate optimal feature extractor $f_{\hat{\theta}}(\mathbf{x})$ with approximate oracle Bayes error L^* is given by:*

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) := (\partial_{\theta_{11}^*} g(\hat{\theta}_1), \cdots, \partial_{\theta_{1p}^*} g(\hat{\theta}_1))^T \tag{D.4}$$

Proof: Let us define $v(\theta^*) = F^{-1}(P(y=1|\mathbf{x};\theta^*)) = \log(P_{\theta_1^*}(\mathbf{x})) - \log(P_{\theta_{-1}^*}(\mathbf{x})) = g(\theta_1^*) - g(\theta_{-1}^*)$, then By Taylor expansion around the estimated $\hat{\theta}$ up to the first order, we can approximate $v(\theta^*)$ as

$$\begin{aligned} v(\theta^*) &\approx v(\hat{\theta}) + \sum_{i=1}^{p} \partial_{\theta_{1i}^*} v(\hat{\theta}_{\mathbf{1}})(\theta_{1i}^* - \hat{\theta}_{1i}) \\ &\quad + \sum_{j=1}^{p} \partial_{\theta_{-1j}^*} v(\hat{\theta}_{\mathbf{-1}})(\theta_{-1j}^* - \hat{\theta}_{-1j}) \\ &\approx v(\hat{\theta}) + \sum_{i=1}^{p} \partial_{\theta_{1i}} v(\hat{\theta}_{\mathbf{1}})(\theta_{1i}^* + \theta_{-1i}^* - 2\hat{\theta}_{1i}) \end{aligned} \tag{D.5}$$

where $\partial_{\theta_{ki}^*} v = \frac{\partial v}{\partial \theta_{ki}^*}$ and $\partial_{\theta_{ki}^*} v(\hat{\theta}_k)$ denotes v's derivative at the point $\hat{\theta}_k$ ($k \in \{1, -1\}$ and $1 \leq i \leq p$).

Since in the semi-supervised model sequence anomaly detection, only the normal data are available for the estimation $\hat{\theta}_{\mathbf{1}}$ and $\hat{\theta}_{\mathbf{-1}}$ has no data to estimate. We use $\hat{\theta}_{\mathbf{1}}$ to approximate $\hat{\theta}_{\mathbf{-1}}$. This is reasonable because the abnormal sequences are highly similar to the normal ones (i.e., $\theta_{\mathbf{1}}^* \approx \theta_{\mathbf{-1}}^*$). In addition,

$$\begin{aligned} v(\theta) &= F^{-1}(P_\theta(y=1|\mathbf{x})) \\ &= \log(P_\theta(y=1|\mathbf{x})) - \log(P_\theta(Y=-1|\mathbf{x})) \\ &= \log(P_{\theta_{\mathbf{1}}}(\mathbf{x})) - \log(P_{\theta_{\mathbf{-1}}}(\mathbf{x})) \end{aligned} \tag{D.6}$$

Then Equation D.5 becomes:

$$\begin{aligned} v(\theta^*) &\approx \sum_{i=1}^{p} \frac{1}{P_{\hat{\theta}_{\mathbf{1}}}(\mathbf{x})} \partial_{\theta_{1i}} (P_{\hat{\theta}_{\mathbf{1}}}(\mathbf{x}))(\theta_{1i}^* - \hat{\theta}_{1i}) \\ &\quad - \sum_{j=1}^{p} \frac{1}{P_{\hat{\theta}_{\mathbf{-1}}}(\mathbf{x})} \partial_{\theta_{-1j}} (P_{\hat{\theta}_{\mathbf{-1}}}(\mathbf{x}))(\theta_{-1j}^* - \hat{\theta}_{-1j}) \\ &= \sum_{i=1}^{p} \partial_{\theta_{1i}^*} g(\hat{\theta}_1)(\theta_{1i}^* - \theta_{-1i}^*) \end{aligned} \tag{D.7}$$

Consequently, by setting

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) := (\partial_{\theta^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\theta^*_{1p}} g(\hat{\theta}_1))^T \tag{D.8}$$

and

$$\mathbf{w} := \mathbf{w}^* = (\theta^*_{11} - \theta^*_{-11}, \cdots, \theta^*_{1p} - \theta^*_{-1p})^T, b = 0. \tag{D.9}$$

the proposed feature extractor with the optimal classifier achieves a reasonable small $D(f_{\hat{\theta}}) \approx 0$ for the upper bound of classification error difference.

D.3 Theoretical Comparison of Performance

In this section, we theoretically compare the proposed feature extractor with the model-based anomaly detection in terms of approximation to the oracle Bayes error. $P(y = 1|\mathbf{x}; \theta)$ is assumed to $\in (0, 1)$[1] and $\nabla_\theta P(y = 1|\mathbf{x}; \theta))$ and $\nabla^2_\theta P(y = 1|\mathbf{x}; \theta))$ are assumed to be bounded, where $\nabla_\theta f = (\partial_{\theta_1} f, \cdots, \partial_{\theta_p} f)^T$ and the (i, j)th element of ∇^2_θ is $\frac{\partial^2 f}{\partial \theta_i \partial \theta_j}$. Then we have the upper bound of classification error difference between the model-based algorithm and the oracle classifier[2] is:

$$D(\hat{\theta}) = E_x |P(y = 1|\mathbf{x}; \hat{\theta}) - P(y = 1|\mathbf{x}; \theta^*)|. \tag{D.10}$$

Define $\mathbf{\Delta}\theta = \theta^* - \hat{\theta}$. By Taylor expansion around $\hat{\theta}$, we have

$$\begin{aligned} D(\hat{\theta}) &\approx Ex|(\mathbf{\Delta}\theta)^T \nabla_\theta P(y = 1|\mathbf{x}, \theta^*) \\ &+ \frac{1}{2}(\mathbf{\Delta}\theta)^T \nabla^2_\theta P(y = 1|\mathbf{x}, \theta_{\mathbf{0}})(\mathbf{\Delta}\theta)| \\ &= O(\|\mathbf{\Delta}\theta\|). \end{aligned} \tag{D.11}$$

By contrast, when the proposed feature extractor is used,

$$D(f_{\hat{\theta}}) = E_{\mathbf{x}} |F((\mathbf{w}^*)^T f_{\hat{\theta}}(\mathbf{x})) - P_{\theta^*}(y = 1|\mathbf{x})|, \tag{D.12}$$

[1] To prevent $|v(\theta)|$ from going to infinity.

[2] Here for simplicity, we use $P(y = 1|\mathbf{x}; \hat{\theta})$ to replace $P(\mathbf{x}|y = 1; \hat{\theta})$, where $P(\mathbf{x}|y = 1; \hat{\theta}_{-1})$ is estimated as a constant.

where $\mathbf{w}^*$ is defined as in Equation D.9. Since F is Lipschitz continuous, there is a finite positive constant M such that $|F(a) - F(b)| \leq M|a - b|$ (Tsuda, Kawanabe, Ratsch, Sonnenburg & Muller 2002). Thus,

$$\begin{aligned} D(f_{\hat{\theta}}) &\leq ME_{\mathbf{x}}|(\mathbf{w}^*)^T f_{\hat{\theta}}(\mathbf{x}) - F^{-1}(P_{\theta^*}(y = 1|\mathbf{x}))| \\ &= O(\|\mathbf{\Delta}\theta\|^2). \end{aligned} \tag{D.13}$$

Since $(\mathbf{w}^*)^T f_{\hat{\theta}}(\mathbf{x})$ is the Taylor expansion of $F^{-1}(P_{\theta^*}(y = 1|\mathbf{x}))$ up to the first order and the first-order terms of $\mathbf{\Delta}\theta$ are excluded from the right side of Equation D.5; thus, $D(f_{\hat{\theta}}) = O(\|\mathbf{\Delta}\theta\|^2)$. Since both the model-based and the proposed feature extractor algorithms depend on the parameter estimate $\hat{\theta}$, the upper bounds of error difference $D(\hat{\theta})$ and $D(f_{\hat{\theta}})$ become smaller as $\|\mathbf{\Delta}\theta\|$ decreases. However,the rate of convergence of the proposed feature extractor is much faster than that of the model-based algorithm if $\mathbf{w}$ and b are optimally chosen. To put it in another way, the proposed feature extractor has a better approximation to the optimal Bayes error theoretically.

Appendix E

List of My Publications

Papers Published

- **Song, Yin** and Cao, Longbing (2012), Graph-based coupled behavior analysis: A case study on detecting collaborative manipulations in stock markets, IJCNN 2012, pp. 1-8. (**ERA ranking: A**)

- **Song, Yin** and Cao, Longbing and Wu, Xindong and Wei, Gang and Ye, Wu and Ding, Wei (2012), Coupled behavior analysis for capturing coupling relationships in group-based market manipulations, KDD 2012, pp. 976-984. (**ERA ranking: A**)

- **Song, Yin** and Cao, Longbing and Yin, Junfu and Wang, Cheng (2013), Extracting Discriminative Features for Identifying Abnormal Sequences in One-class Mode, IJCNN 2013, pp. 1-8. (**ERA ranking: A**)

- Cao, Wei and Cao, Longbing and **Song, Yin** (2013), Coupled Market Behavior Based Financial Crisis Detection, IJCNN 2013, pp. 1-8. (**ERA ranking: A**)

- **Song, Yin** and Zhang, Jian and Cao, Longbing and Sangeux, Morgan (2013), On Discovering the Correlated Relationship between Static and Dynamic Data in Clinical Gait Analysis, 'Machine Learning and

Knowledge Discovery in Databases', Vol. 8190, Springer Berlin Heidelberg, pp. 563-578. (**ERA ranking: A**)

- **Song, Yin** and Cao, Longbing and Fan, Xuhui and Cao, Wei and Zhang, Jian. Characterizing A Database of Sequential Behaviors with Latent Dirichlet Hidden Markov Models, arXiv:1305.5734v1 [stat.ML].

- Yin, Junfu and Zheng, Zhigang and Cao, Longbing and **Song, Yin** and Wei, Wei. Efficiently Mining Top-K High Utility Sequential Patterns, The 2013 IEEE International Conference on Data Mining series (ICDM 2013). (**ERA ranking: A**).

Papers to be Submitted/Under Review

- Fan, Xuhui and Xu, Richard and Cao, Longbing and **Song, Yin**, Learning Hidden Structures with Relational Models by Adequately Involving Rich Information in A Network, submitted to ICML 2014.

- **Song, Yin** and Zhang, Jian and Cao, Longbing, A Joint Interest-Social Latent Factor Model for Social Recommendation, to be submitted as a journal paper.

- Yin, Junfu and Cao, Longbing and **Song, Yin**. UIP-Miner: An Efficient Algorithm for High Utility Inter-transaction Pattern Mining, to be submitted as a journal paper.

Bibliography

Agarwal, D. & Chen, B.-C. (2009), Regression-based latent factor models, *in* 'Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 19–28.

Agrawal, R. & Srikant, R. (1995), Mining sequential patterns, *in* 'Proceedings of the Eleventh International Conference on Data Engineering', IEEE, pp. 3–14.

Airoldi, E. M., Blei, D. M., Fienberg, S. E. & Xing, E. P. (2008), 'Mixed membership stochastic blockmodels', *The Journal of Machine Learning Research* **9**, 1981–2014.

Alon, J., Sclaroff, S., Kollios, G. & Pavlovic, V. (2003), 'Discovering clusters in motion time-series data'.

Andrieu, C., De Freitas, N., Doucet, A. & Jordan, M. I. (2003), 'An introduction to mcmc for machine learning', *Machine learning* **50**(1-2), 5–43.

Baldi, P. & Brunak, S. (2001), *Bioinformatics: the machine learning approach*, MIT Press.

Barnett, V. & Lewis, T. (1994), *Outliers in statistical data*, Wiley Chichester.

Baum, L., Petrie, T., Soules, G. & Weiss, N. (1970), 'A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains', *The Annals of Mathematical Statistics* **41**(1), 164–171.

Bayes, T. (1763), 'An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs', *Philosophical Transactions (1683-1775)* pp. 370–418.

Beal, M. J. (2003), Variational algorithms for approximate Bayesian inference, PhD thesis.

Bishop, C. (2006), *Pattern recognition and machine learning*, Information Science and Statistics, Springer, New York.

Blasiak, S. & Rangwala, H. (2011), A hidden markov model variant for sequence classification, *in* 'Proceedings of the Twenty-Second international joint conference on Artificial Intelligence', Vol. 2, AAAI Press, pp. 1192–1197.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *The Journal of Machine Learning Research* **3**, 993–1022.

Blender, R., Fraedrich, K. & Lunkeit, F. (1997), 'Identification of cyclone-track regimes in the north atlantic', *Quarterly Journal of the Royal Meteorological Society* **123**(539), 727–741.

Brand, M., Oliver, N. & Pentland, A. (n.d.), Coupled hidden markov models for complex action recognition, *in* 'Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition', pp. 994–999.

Braunstein, A., Mzard, M. & Zecchina, R. (2005), 'Survey propagation: An algorithm for satisfiability', *Random Structures & Algorithms* **27**(2), 201–226.

Budalakoti, S., Srivastava, A., Akella, R. & Turkov, E. (2006), 'Anomaly detection in large sets of high-dimensional symbol sequences', *NASA Ames Research Center, Tech. Rep. NASA TM-2006-214553* .

Budalakoti, S., Srivastava, A. & Otey, M. (2009), 'Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **39**(1), 101–113.

Cao, L. (2013), 'Non-iidness learning in behavioral and social data', *The Computer Journal* p. bxt084.

Cao, L., Ou, Y. & Yu, P. (2011), 'Coupled behavior analysis with applications', *IEEE Transactions on Knowledge and Data Engineering* .

Cao, L., Ou, Y., Yu, P. & Wei, G. (2010), Detecting abnormal coupled sequences and sequence changes in group-based manipulative trading behaviors, *in* 'Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 85–94.

Chandola, V., Banerjee, A. & Kumar, V. (2009), 'Anomaly detection: A survey', *ACM Computing Surveys (CSUR)* **41**(3), 1–58.

Chandola, V., Cheboli, D. & Kumar, V. (2009), Detecting anomalies in a timeseries database, Technical report, CS Technical Report 09-004, Computer Science Department, University of Minnesota.

Chau, T. (2001), 'A review of analytical techniques for gait data. part 1: fuzzy, statistical and fractal methods', *Gait & posture* **13**(1), 49–66.

Dawid, A. P. (1979), 'Conditional independence in statistical theory', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 1–31.

Dempster, A., Laird, N. & Rubin, D. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1), 1–38.

Desloovere, K., Molenaers, G., Feys, H., Huenaerts, C., Callewaert, B. & Walle, P. (2006), 'Do dynamic and static clinical measurements correlate

with gait analysis parameters in children with cerebral palsy?', *Gait & Posture* **24**(3), 302–313.

Devroye, L., Gyorfi, L. & Lugosi, G. (1996), *A probabilistic theory of pattern recognition*, Vol. 31, Springer Verlag.

Ding, Y. & Fan, G. (2008), Multi-channel segmental hidden markov models for sports video mining, *in* 'Proceeding of the 16th ACM international conference on Multimedia', ACM, pp. 697–700.

Evans, M., Hastings, N. & Peacock, B. (2000), 'Statistical distributions', *Measurement Science and Technology* **12**(1), 117.

Fawcett, T. (2006), 'An introduction to roc analysis', *Pattern recognition letters* **27**(8), 861–874.

Feng, L., Yu, J., Lu, H. & Han, J. (2002), 'A template model for multidimensional inter-transactional association rules', *The VLDB journal* **11**(2), 153–175.

Friedman, N. & Koller, D. (2003), 'Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks', *Machine learning* **50**(1-2), 95–125.

Fu, T.-c., Chung, F.-l., Ng, V. & Luk, R. (2001), Pattern discovery from stock time series using self-organizing maps, *in* 'Workshop Notes of KDD2001 Workshop on Temporal Data Mining', Citeseer, pp. 26–29.

Fukunaga, K. (1990), *Introduction to statistical pattern recognition*, Academic Pr.

Gaffney, S. & Smyth, P. (1999), Trajectory clustering with mixtures of regression models, ACM, pp. 63–72.

Geman, S. & Geman, D. (1984), 'Stochastic relaxation, gibbs distributions, and the bayesian restoration of images', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (6), 721–741.

Gershman, S., Hoffman, M. & Blei, D. (2012), 'Nonparametric variational inference', *arXiv preprint arXiv:1206.4665* .

Ghahramani, Z. (1997), 'On structured variational approximations', *University of Toronto Technical Report, CRG-TR-97-1* .

Ghahramani, Z. (1998), 'Learning dynamic bayesian networks', *Adaptive Processing of Sequences and Data Structures* p. 168.

Ghahramani, Z. & Beal, M. (2000), 'Variational inference for bayesian mixtures of factor analysers', *Advances in neural information processing systems* **12**, 449–455.

Ghahramani, Z. & Hinton, G. E. (2000), 'Variational learning for switching state-space models', *Neural Computation* **12**(4), 831–864.

Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S. & Ofek-Koifman, S. (2009), Personalized recommendation of social software items based on social relations, *in* 'Proceedings of the third ACM conference on Recommender systems', ACM, pp. 53–60.

Han, J., Kamber, M. & Pei, J. (2011), *Data mining: Concepts and Techniques*, 3rd edn, Morgan Kaufmann.

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. & Hsu, M. (2001), Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth, *in* 'Proceedings of the 17th International Conference on Data Engineering', pp. 215–224.

Han, J., Pei, J., Yin, Y. & Mao, R. (2004), 'Mining frequent patterns without candidate generation: A frequent-pattern tree approach', *Data Mining and Knowledge Discovery* **8**(1), 53–87.

He, Z., You, X. & Tang, Y. (2008), 'Writer identification of chinese handwriting documents using hidden markov tree model', *Pattern Recognition* **41**(4), 1295–1307.

Heckerman, D. (2008), *A tutorial on learning with Bayesian networks*, Springer.

Herlocker, J. L., Konstan, J. A., Borchers, A. & Riedl, J. (1999), An algorithmic framework for performing collaborative filtering, *in* 'Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 230–237.

Hofmann, T. (2003), Collaborative filtering via gaussian probabilistic latent semantic analysis, *in* 'Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval', ACM, pp. 259–266.

Hsu, C., Chang, C. & Lin, C. (2003), 'A practical guide to support vector classification'.

Huang, X. & An, A. (2002), Discovery of interesting association rules from livelink web log data, *in* 'Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on', IEEE, pp. 763–766.

Jaakkola, T. & Haussler, D. (1999), 'Exploiting generative models in discriminative classifiers', *Advances in neural information processing systems* pp. 487–493.

Jaakkola, T. & Jordan, M. (1997), Variational methods for inference and estimation in graphical models, PhD thesis.

Jansen, M., White, T. P., Mullinger, K. J., Liddle, E. B., Gowland, P. A., Francis, S. T., Bowtell, R. & Liddle, P. F. (2012), 'Motion-related artefacts in eeg predict neuronally plausible patterns of activation in fmri data', *Neuroimage* **59**(1), 261–270.

Jaynes, E. T. (1986), 'Bayesian methods: General background'.

Joachims, T. (2000), Estimating the generalization performance of a svm efficiently, *in* 'International Conference on Machine Learning', Morgan Kaufman, pp. 431–438.

Jordan, M., Ghahramani, Z., Jaakkola, T. & Saul, L. (1999), 'An introduction to variational methods for graphical models', *Machine learning* **37**(2), 183–233.

Joshi, S. & Phoha, V. (2005), Investigating hidden markov models capabilities in anomaly detection, ACM, pp. 98–103. Proceedings of the 43rd annual Southeast regional conference-Volume 1.

Kindermann, R. & Snell, J. L. (1980), *Markov random fields and their applications*, Vol. 1, American Mathematical Society Providence, RI.

Koren, Y. (2010), 'Collaborative filtering with temporal dynamics', *COMMUNICATIONS OF THE ACM* **53**(4), 89–97.

Koren, Y., Bell, R. & Volinsky, C. (2009), 'Matrix factorization techniques for recommender systems', *Computer* **42**(8), 30–37.

Kotz, S., Balakrishnan, N. & Johnson, N. (2000), *Continuous multivariate distributions, models and applications*, Vol. 334, Wiley-Interscience.

Kratz, L. & Nishino, K. (2009), 'Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models'.

Lafferty, J., McCallum, A. & Pereira, F. (2001), Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *in* 'Proceedings of the Eighteenth International Conference on Machine Learning', Morgan Kaufmann Publishers Inc., pp. 282–289.

Lee, H.-K. & Kim, J.-H. (1999), 'An hmm-based threshold model approach for gesture recognition', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **21**(10), 961–973.

Lim, Y. J. & Teh, Y. W. (2007), Variational bayesian approach to movie rating prediction, *in* 'Proceedings of KDD Cup and Workshop', Citeseer, pp. 15–21.

Liu, F. & Lee, H. J. (2010), 'Use of social network information to enhance collaborative filtering performance', *Expert Systems with Applications* **37**(7), 4772–4778.

Liu, H., Salerno, J. & Young, M. (2008), *Social computing, behavioral modeling, and prediction*, Springer-Verlag New York Inc.

Lu, E., Tseng, V. & Yu, P. (2011), 'Mining cluster-based temporal mobile sequential patterns in location-based service environments', *IEEE transactions on Knowledge and Data Engineering* **23**(6), 914–927.

Lu, H., Han, J. & Feng, L. (1998), Stock movement prediction and n-dimensional inter-transaction association rules, Citeseer, p. 12. Proceedings of the 1998 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.

Ma, H., Yang, H., Lyu, M. R. & King, I. (2008), Sorec: social recommendation using probabilistic matrix factorization, *in* 'Proceedings of the 17th ACM conference on Information and knowledge management', ACM, pp. 931–940.

Ma, H., Zhou, D., Liu, C., Lyu, M. R. & King, I. (2011), Recommender systems with social regularization, *in* 'Proceedings of the fourth ACM international conference on Web search and data mining', ACM, pp. 287–296.

MacKay, D. (1997), Ensemble learning for hidden markov models, Technical report, Technical report, Cavendish Laboratory, University of Cambridge.

Mahadevan, V., Li, W., Bhalodia, V. & Vasconcelos, N. (2010), 'Anomaly detection in crowded scenes'.

McPherson, M., Smith-Lovin, L. & Cook, J. (2001), 'Birds of a feather: Homophily in social networks', *Annual review of sociology* pp. 415–444.

Minka, T. (2000), 'Estimating a dirichlet distribution'.

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M. & Newman, S. (2004), 'Learning to decode cognitive states from brain images', *Machine learning* **57**(1-2), 145–175.

Mnih, A. & Salakhutdinov, R. (2007), Probabilistic matrix factorization, *in* 'Advances in neural information processing systems', pp. 1257–1264.

Murphy, K. P. (2002), Dynamic bayesian networks: representation, inference and learning, PhD thesis.

Natarajan, P. & Nevatia, R. (2007), Coupled hidden semi markov models for activity recognition, IEEE, pp. 10–10. Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on.

Novak, P. K., Lavrac, N. & Webb, G. I. (2009), 'Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining', *The Journal of Machine Learning Research* **10**, 377–403.

Obermaier, B., Guger, C., Neuper, C. & Pfurtscheller, G. (2001), 'Hidden markov models for online classification of single trial eeg data', *Pattern recognition letters* **22**(12), 1299–1309.

Olshen, L. & Stone, C. (1984), 'Classification and regression trees', *Wadsworth International Group* .

Pearl, J. (2000), *Causality: models, reasoning and inference*, Vol. 29, Cambridge Univ Press.

Perry, J. & Davids, J. (1992), 'Gait analysis: normal and pathological function', *Journal of Pediatric Orthopaedics* **12**(6), 815.

Pfanzagl, J. (1994), *Parametric statistical theory*, Walter de Gruyter.

Philips, W. (1993), 'Ecg data compression with time-warped polynomials', *Biomedical Engineering, IEEE Transactions on* **40**(11), 1095–1101.

Piateski, G. & Frawley, W. (1991), *Knowledge discovery in databases*, MIT press.

Plant, C., Wohlschlager, A. & Zherdin, A. (2009), Interaction-based clustering of multivariate time series, IEEE, pp. 914–919. Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on.

Provost, F. & Domingos, P. (2003), 'Tree induction for probability-based ranking', *Machine learning* **52**(3), 199–215.

Provost, F. & Fawcett, T. (2001), 'Robust classification for imprecise environments', *Machine learning* **42**(3), 203–231.

Rabiner, L. (1990), 'A tutorial on hidden markov models and selected applications in speech recognition', *Readings in speech recognition* **53**(3), 267–296.

Ratanamahatana, C. A. & Keogh, E. (2005), Three myths about dynamic time warping data mining, *in* 'Proceedings of SIAM International Conference on Data Mining (SDM05)', pp. 506–510.

Rennie, J. D. & Srebro, N. (2005), Fast maximum margin matrix factorization for collaborative prediction, *in* 'Proceedings of the 22nd international conference on Machine learning', ACM, pp. 713–719.

Rezek, I., Gibbs, M. & Roberts, S. (2002), 'Maximum a posteriori estimation of coupled hidden markov models', *The Journal of VLSI Signal Processing* **32**(1), 55–66.

Rezek, I. & Roberts, S. (n.d.), Estimation of coupled hidden markov models with application to biosignal interaction modelling, Vol. 2, IEEE, pp. 804–813. Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop.

Rue, H. & Held, L. (2005), *Gaussian Markov random fields: theory and applications*, CRC Press.

Sagawa, Y., Watelain, E., De Coulon, G., Kaelin, A. & Armand, S. (2012), 'What are the most important clinical measurements affecting gait in patients with cerebral palsy?', *Gait & Posture* **36**, S11–S12.

Salakhutdinov, R. & Mnih, A. (2008), Bayesian probabilistic matrix factorization using markov chain monte carlo, *in* 'Proceedings of the 25th international conference on Machine learning', ACM, pp. 880–887.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2001), Item-based collaborative filtering recommendation algorithms, *in* 'Proceedings of the 10th international conference on World Wide Web', ACM, pp. 285–295.

Scholkopf, B. & Smola, A. (2002), *Learning with kernels: Support vector machines, regularization, optimization, and beyond*, the MIT Press.

Sekar, R., Bendre, M., Dhurjati, D. & Bollineni, P. (2001), A fast automaton-based method for detecting anomalous program behaviors, IEEE, pp. 144–155. Security and Privacy, 2001. S & P 2001. Proceedings. 2001 IEEE Symposium on.

Shan, H. & Banerjee, A. (2010), Generalized probabilistic matrix factorizations for collaborative filtering, *in* 'Data Mining (ICDM), 2010 IEEE 10th International Conference on', IEEE, pp. 1025–1030.

Shani, G. & Gunawardana, A. (2011), 'Evaluating recommendation systems', *Recommender Systems Handbook* pp. 257–297.

Song, Y. & Cao, L. (2012), Graph-based coupled behavior analysis: A case study on detecting collaborative manipulations in stock markets, *in* 'Neural Networks (IJCNN), The 2012 International Joint Conference on', IEEE, pp. 1–8.

Song, Y., Cao, L., Wu, X., Wei, G., Ye, W. & Ding, W. (2012), Coupled behavior analysis for capturing coupling relationships in group-based market manipulations, *in* 'Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 976–984.

Su, X. & Khoshgoftaar, T. M. (2009), 'A survey of collaborative filtering techniques', *Advances in Artificial Intelligence* **2009**, 4.

Tang, L. & Liu, H. (2009), Scalable learning of collective behavior based on sparse social dimensions, *in* 'Proceeding of the 18th ACM conference on Information and knowledge management', ACM, pp. 1107–1116.

Tang, L. & Liu, H. (2011), 'Leveraging social media networks for classification', *Data Mining and Knowledge Discovery* pp. 1–32.

Tran, Q., Zhang, Q. & Li, X. (2003), Evolving training model method for one-class svm, *in* 'IEEE INTERNATIONAL CONFERENCE ON SYSTEMS MAN AND CYBERNETICS', Vol. 3, IEEE, pp. 2388–2393.

Tsuda, K., Kawanabe, M. & Muller, K. (2002), 'Clustering with the fisher score', *Advances in neural information processing systems* **15**, 729–736.

Tsuda, K., Kawanabe, M., Ratsch, G., Sonnenburg, S. & Muller, K. (2002), 'A new discriminative kernel from probabilistic models', *Neural Computation* **14**(10), 2397–2414.

Tung, A., Lu, H., Han, J. & Feng, L. (1999), Breaking the barrier of transactions: Mining inter-transaction association rules, ACM, pp. 297–301. Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining.

Tung, A., Lu, H., Han, J. & Feng, L. (2003), 'Efficient mining of intertransaction association rules', *IEEE transactions on Knowledge and Data Engineering* pp. 43–56.

Velivelli, A., Huang, T. & Hauptmann, A. (2006), Video shot retrieval using a kernel derived from a continuous hmm, Technical Report 980, Carnegie Mellon University.

Wang, J. & Singh, S. (2003), 'Video analysis of human dynamics–a survey', *Real-time imaging* **9**(5), 321–346.

Warrender, C., Forrest, S. & Pearlmutter, B. (1999), Detecting intrusions using system calls: Alternative data models, *in* 'Proceedings of the 1999 IEEE Symposium on Security and Privacy', pp. 133–145.

Wilpon, J. & Rabiner, L. (1985), 'A modified k-means clustering algorithm for use in isolated work recognition', *Acoustics, Speech and Signal Processing, IEEE Transactions on* **33**(3), 587–594.

Xin, X., King, I., Deng, H. & Lyu, M. R. (2009), A social recommendation framework based on multi-scale continuous conditional random fields, *in* 'Proceedings of the 18th ACM conference on Information and knowledge management', ACM, pp. 1247–1256.

Yang, S.-H., Long, B., Smola, A., Sadagopan, N., Zheng, Z. & Zha, H. (2011), Like like alike: joint friendship and interest propagation in social networks, *in* 'Proceedings of the 20th international conference on World wide web', ACM, pp. 537–546.

Zaki, M. J. (2001), 'Spade: An efficient algorithm for mining frequent sequences', *Machine learning* **42**(1), 31–60.

Zhang, B.-l., Zhang, Y. & Begg, R. K. (2009), 'Gait classification in children with cerebral palsy by bayesian approach', *Pattern Recognition* **42**(4), 581–586.

Zhong, S. & Ghosh, J. (2001), A new formulation of coupled hidden markov models, Technical report, technical report, Dept. of Electrical and Computer Eng., Univ. of Texas at Austin.

Zhong, S. & Ghosh, J. (2002), Hmms and coupled hmms for multi-channel eeg classification, Vol. 2, IEEE, pp. 1154–1159. Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on.