SEQUENTIAL PATTERN ANALYSIS IN DYNAMIC BUSINESS ENVIRONMENTS

by

CHUANREN LIU

A dissertation submitted to the

Graduate School-Newark

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Management

written under the direction of

Dr. Hui Xiong

and approved by

_____

_____

_____

_____

Newark, New Jersey

May 2015

ABSTRACT OF THE DISSERTATION

Sequential Pattern Analysis in Dynamic Business Environments

By CHUANREN LIU

Dissertation Advisor: Dr. Hui Xiong

Sequential pattern analysis targets on finding statistically relevant temporal structures where the values are delivered in sequences. This is a fundamental problem in data mining with diversified applications in many science and business fields, such as multimedia analysis (motion gesture/video sequence recognition), marketing analytics (buying path identification), and financial modelling (trend of stock prices). Given the overwhelming scale and the dynamic nature of the sequential data, new techniques for sequential pattern analysis are required to derive competitive advantages and unlock the power of the big data. In this dissertation, we develop novel approaches for sequential pattern analysis with applications in dynamic business environments.

Our major contribution is to identify the right granularity for sequential pattern analysis. We first show that the right pattern granularity for sequential pattern mining is often unclear due to the so-called "curse of cardinality", which corresponds to a variety of difficulties in mining sequential patterns from massive data represented by a huge set of symbolic features. Therefore, pattern mining with the original features may provide few clues on interesting temporal dynamics. To address this challenge, our approach, temporal skeletonization, reduces the representation of the sequential data by uncovering significant, hidden temporal structures. Furthermore, the right granularity is also critical for sequential pattern modelling. Particularly,

there are often multiple granularity levels accessible for estimating statistical models with the sequential data. However, on one hand, the patterns at the lowest level may be too complicated for the models to produce application-enabling results; and on the other hand, the patterns at the highest level may be as trivial as common sense, which are already known without analyzing the data. To dig out the most value from the data, we propose to construct the modelling granularity in a data-driven manner balancing between the above two extremes. By identifying the right pattern granularity for both sequential pattern mining and modelling, we have successful applications in B2B (Business-to-Business) marketing analytics, healthcare operation and management, and modeling of the product adoption in digit markets, as three case studies in dynamic business environments.

ACKNOWLEDGEMENTS

I would like to express my great appreciation to all the people who provided me with tremendous support and help during my graduate study.

First, I would like to sincerely thank my advisor, Prof. Hui Xiong, for his invaluable encouragement and advise. I thank him for generously giving me support, understanding, assistance and opportunities. Only with his help, I could be able to achieve the Ph.D. degree in Rutgers Business School and gain my confidence in my future career. Prof. Xiong is also a wonderful friend of all our group members. We celebrated so many memorable holidays together. I believe we will share and celebrate even more of our happiness in the future.

I also sincerely thank my other committee members: Prof. Spiros Papadimitriou, Prof. Jian Yang, and Prof. Guiling Wang. Prof. Spiros Papadimitriou is a great collaborator and mentor. He not only helped me in research but also offered career and personal guidance. I thank him particularly for his time rehearsing me for job talks and interviews. Prof. Jian Yang taught me the course on programming and optimization in the first semester of my graduate study. I thank him for providing me with constructive suggestions on my research from new perspectives. Prof. Guiling Wang has also provided many useful feedback and discussions for my research. I thank her very much for her help and valuable time for my graduation.

Special thanks are due to Prof. Xiaodong Lin at Rutgers Business School, Prof. Qiang Yang at Hong Kong University of Science and Technology, Dr. Kai Zhang at NEC Laboratories America, Prof. Fei Wang at University of Connecticut, Prof. Tianming Hu at Dongguan University of Technology, Prof. Yong Ge at University of North Carolina at Charlotte, Prof. Keli Xiao at Stony Brook University, Prof. Wenjun Zhou at University of Tennessee at Knoxville, and Prof. Panagiotis Karras at Skolkovo Institute of Science and Technology, who all helped with my research and career development. Thanks are also due to Dr. Guofei Jiang, Dr. Jianying Hu, Dr. Jianjun Xie, Dr. Chunyu Luo, Dr. Hengshu Zhu, Prof. Qi Liu, Prof. Bo Jin, Prof. Enhong Chen, Prof. Zhihua Zhou, Dr. Jing Yuan, and Dr. Xing Xie. It was a great pleasure working with all of them. I am also indebted to my colleagues and friends: Zhongmou Li, Yanchi Liu, Bin Liu, Zijun Yao, Chang Tan, Aimin Feng, Liyang Tang, Yanjie Fu, Konstantine Alexander Vitt, Xue Bai, Jingyuan Yang, Meng Qu, Can Chen, Hao Zhong, Farid Razzak, Junming Liu, Tong Xu, Guannan Liu, Qingxing Meng, Yuanchun Zhou, Hongting Niu, Jing Sun, Ling Yin, Xiaolin Li, Xinjiang Lu, Chu Guan, Bowen Du, Jiadi Du, and Mingfei Teng, for their help and friendship.

I would also like to acknowledge our department and school in Rutgers for the perfect organization and for all the facilities we have at our disposal. I want to particularly thank Ms. Luz Kosar, Ms. Monnique Desilva, and Mr. Goncalo Filipe for all of their assistance.

Finally, I would like to thank my wife, my parents, and my sisters, for their love, support and understanding. My work is always dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Sequential pattern analysis, unravelling meaningful and significant temporal structures from large-scale sequential data, is a fundamental data mining task which has diversified applications, such as mining the customer purchase sequences, Web clickstreams modelling, motion gesture/video sequence recognition, and biological sequence analysis [Han et al., 2007]. This dissertation contributes towards this fundamental task in data mining with a focus on sequential pattern modelling and mining.

In particular, sequential pattern modelling infers a statistical model with a set of parameters, with which the model is able to simulate the modelled processes without breaking statistically significant characteristics. Hence, sequential pattern modelling provides parsimonious descriptions for the sequential data and the underlying complex dynamics hidden in the data. With the sequential pattern modelling techniques, the dynamics can be proactively monitored, quantitatively audited, and intuitively inspected [Boots and Gordon, 2011, Cao et al., 2009b, Bureau et al., 2003, Lafferty et al., 2001, Galata et al., 2001].

In contrast, instead of assuming a specific statistical structure, sequential pattern mining directly searches the data for frequent associations, which might be subsets of items, subsequences, or subgraphs. These associations capture different orders of temporal correlations, which can be used for different analytic tasks. Conventionally,

research efforts focused on the computing efficiency of sequential pattern mining with a variety of constraints over large-scale data sets [Yin et al., 2012, Lo et al., 2011, Giannotti et al., 2006, Pei et al., 2004, Agrawal and Srikant, 1995].

In the following of this chapter, we first introduce our research motivation from real-world applications of sequential pattern analysis. Then we highlight the contributions of our research, and overview the major contents of this dissertation.

1.1 Research Motivation

In this dissertation, we aim to address the unique challenges of sequential pattern modelling and mining, from both theoretical and practical perspectives. For the sequential pattern mining, one unique challenge we have is the so-called "curse of cardinality", which is often observed with the growing complexity of real-world scenarios. To be specific, the curse of cardinality corresponds to a variety of difficulties in mining sequential patterns when the sequential data are symbolic and represented by a huge set of symbolic features. For example, a large number of symbols in a sequence can "dilute" useful patterns which themselves exist on a different level of granularity. Therefore, pattern mining with the original huge set of symbols may provide few clues on interesting temporal structures. In the literature, the curse of cardinality is often suppressed by performing a grouping of the symbols, for which either a taxonomy already exists [Srikant and Agrawal, 1995], or extracted from domain knowledge [Han and Fu, 1999], or through clustering on the features associated with the symbols [Giannotti et al., 2007c]. However, these grouping are performed irrespective of the temporal content in the sequences, which might fail to capture

intrinsic characteristics of sequential patterns. Instead, we propose a novel approach, temporal skeletonization, to reducing the representation of the sequential data by directly summarizing and analyzing the temporal correlations of the symbols. This part of our research is motivated by an application of customer purchase pattern analysis in B2B (Business-to-Business) marketing, where we have numerous symbolic marketing events in the sequential behavior records of the business customers.

The right granularity is also critical for inferring statistical models with sequential data (sequential pattern modelling). In practice, there are often multiple granularity levels accessible for modelling the sequential data, while the optimal granularity level is unknown. For example, for the healthcare workflow modelling with location traces of medical devices, the concept of workflow patterns is actually a hierarchy with several levels. At the lowest level, the location trace itself can be seen as an instance of workflow patterns. On the other hand, three workflow stages at the highest level are widely used in the healthcare industry to describe the workflow logistics: preprocessing maintenance stage, in-use stage, postprocessing maintenance stage. However, modeling the workflow patterns based on either the raw locations or the three stages will be difficult to produce useful results with the location traces of moving objects in the indoor space. In other words, the right modelling granularity level is needed to help with tasks of operation and management in a hospital. To balance between the extremes, we propose to first identify the workflow states as the location spots in the hospital environments where specific healthcare activities frequently happen. Then we transform the original location traces into the sequences of identified workflow states and model the state transitions with finite state machines. In this way, we showed

that valuable intelligent applications for healthcare operation and management can be enabled to manage, evaluate and optimize the healthcare services.

1.2 Contributions

First, in the "temporal skeletonization", our approach to identifying the meaningful granularity level for sequential pattern mining, the key idea is to summarize the temporal correlations in an undirected graph. Then, the "skeleton" of the graph serves as a higher granularity level on which hidden temporal patterns are more likely to be identified. In the meantime, the manifold embedding of the graph topology allows us to translate the rich temporal content into a metric space. This opens up new possibilities to explore, quantify, and visualize sequential data. Furthermore, by extending the robust temporal correlations, our approach can be utilized to model the dynamic systems which are generally measured by multivariate time series. Evaluation on a Business-to-Business (B2B) marketing application demonstrates that our approach can effectively discover critical purchase patterns from noisy customer behavior records. Indeed, our work will not only provide new opportunities to improve the marketing practice but also further the research of marketing science. For example, since we can identify dynamic buying stages of customers, we can improve the traditional static customer segmentation practice with dynamic extensions, which allows us to target each customer with the marketing campaigns most relevant to his/her current buying stage. In addition, by aggregating the behavior data of all the customers, we have a systematic way to audit and visualize the marketing effects of the campaigns. From the management perspective, this will help the marketing

managers on the tasks of managing and inventing new campaigns to reduce the customer conversion cycle and increase the customer conversion ratio. This part of our work has been published in the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014) [Liu et al., 2014b].

Second, for statistical sequential pattern modelling, we provide a focused study of workflow modelling with the indoor location traces. In comparison with conventional workflow models, our approach can proactively unravel the workflow patterns hidden in the location traces, by automatically constructing the workflow states and estimating parameters describing the workflow transitions. Specifically, we identify the workflow states as the location spots in the hospital environments where specific healthcare activities frequently happen. The identified workflow states correspond to the right modelling granularity in the indoor space. Thus, we can transform the original location traces into the sequences of workflow states and model the state transitions with finite state machines. Moreover, due to the dynamic indoor structure, we need an automatic and adaptive modelling framework to support the critical applications in real time. However, during some specific periods of observation, the estimation of parameters in the finite state machines may be unstable due to the data scarcity issue of the location traces. Meanwhile, there are some natural correlations in the location traces of a group of medical devices, which are often used together for a particular medical procedure and healthcare task. Therefore, using the Hidden Markov Random Fields (HMRFs), we leverage the correlations in the location traces between related types of medical devices to overcome the data scarcity issue and ultimately reinforce the modelling performance. In this way, we showed that valuable

intelligent applications for healthcare operation and management can be enabled to manage, evaluate and optimize the healthcare services. This part of our work has been published in the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014) [Liu et al., 2014a].

Third, finding the right granularity for sequential pattern modelling is related to localized/context-aware statistical inference. Particularly, although the right granularity is unknown for the overall data, the local scenario can be much clearer. To see this, in another application of sequential pattern analysis with the indoor location traces, we develop the context-aware stochastic model to identity abnormal location traces with missing events of the medical devices. We observed that the abnormal missing events happen in special spots in the indoor environment, and the traces leading to the missing events can be discriminated by stochastic models. Thus, we first use clustering algorithms to identify 'hotspots' (cluster) of missing events. Then, using the hotspots as context of location traces passing nearby, we use a Markov model to quantify the anomaly degree of the nearby location traces. This part of our work has been published in the 12th IEEE International Conference on Data Mining (ICDM 2012) [Liu et al., 2012].

Indeed, identifying the right pattern granularity is often the key for practical sequential pattern analysis. As another example, Hidden Markov Model (HMM) is one widely used method for sequence analysis. However, for the unsupervised HMMs, the Expectation Maximization (EM) based model estimation could suffer from the non-convexity of the objective likelihood, and consequently converge to a local optimum. As a consequence, it is difficult to interpret the estimated states since they

could arbitrarily distribute over the observations. To solve this problem, we introduce a bipartite based method to pre-cluster various observations. Specifically, in the bipartite graph, the observations and sequential records compose the two partite sets, and the frequencies of their co-occurrence are represented with the edges. By clustering the observations in the bipartite graph, we can group temporally correlated observations to initialize the subsequent unsupervised HMM estimation and guide its convergence. As shown with real-world data from digit market, this approach significantly improves the model robustness and interpretability and is able to model the dynamic product adoption in the market. This part of our work has been published in IEEE Transactions on Cybernetics [Zhu et al., 2014].

1.3 Overview

Chapter 2 addresses the curse of cardinality for sequential pattern mining with application to B2B marketing analytics. First, we discuss the motivation of the problem. Then we compute robust temporal correlations for the symbolic observations in the sequential data. We adopt the graph-based algorithms in a novel way to analyze the topology of the correlation graph. Experimental results show that the graph topology is useful to identify typical purchase patterns of B2B customers.

Chapter 3 presents a framework for healthcare workflow modelling. First, we show that it is essential to construct the workflow states with the right granularity in the indoor space. Then, we analyze the characteristics of indoor location traces to define meaningful observation density of the healthcare activities. The density definition can be used in a density-based clustering algorithm to identify the right workflow

states. Further, we integrate the workflow modelling of multiple types of moving objects with a joint learning algorithm. Extensive experimental results demonstrate that the proposed framework can provide not only interpretable workflow patterns but also accurate predictions.

Chapter 4 presents a stochastic model for context-aware anomaly detection in indoor location traces. In detail, we first develop a density-based method to identify the hotspots filled with high-level abnormal activities in the indoor environment. The discovered hotspots serve as the context for nearby trajectories. Then, we introduce an N-gram based method for measuring the degree of anomaly based on the detected hotspots, which is able to predict the missing events possibly due to the devices being stolen. Besides, to address the noisy nature of the indoor sensor networks, we also propose an iterative algorithm to estimate the transition probabilities. This algorithm allows to effectively recover the missing location records which are critical for the abnormality estimation. Finally, the experimental results on the real-world date sets validate the effectiveness of the proposed context-aware anomaly detection method for identifying abnormal events.

Chapter 5 presents a sequential approach based on Hidden Markov Model (HMM) for modeling the popularity information of mobile Apps towards mobile App recommendation. Specifically, we first propose a Popularity based HMM (PHMM) to model the sequences of heterogeneous popularity observations of mobile Apps. Then, we introduce a bipartite based method to pre-cluster the popularity observations. This can help to learn the parameters and initial values of the PHMM model. Furthermore, we demonstrate that the PHMM model is a general model and can be applicable

for various App recommendation services, such as ranking fraud detection, App recommendation, and rating and comment spam detection. Finally, we validate our approach on a real-world data set collected from the Apple App Store. Experimental results clearly show both the effectiveness and efficiency of our approach.

CHAPTER 2

TEMPORAL SKELETONIZATION ON SEQUENTIAL DATA: PATTERNS, CATEGORIZATION, AND VISUALIZATION

2.1 Introduction

Unraveling meaningful and significant temporal structures from large-scale sequential data is a fundamental problem in data mining with diversified applications, such as mining the customer purchasing sequences, motion gesture/video sequence recognition, and biological sequence analysis [Han et al., 2007]. While there have been a large amount of research efforts devoted to this topic and its variants [Agrawal and Srikant, 1995, Ayres et al., 2002, Giannotti et al., 2007c, Han et al., 2001, Zaki, 2001], we are still facing significant emerging challenges. Indeed, with the growing complexity of real-world dynamic scenarios, it often requires more and more symbols to encode a meaningful sequence. For example, in the Business to Business (B2B) marketing analytics, we are interested in finding critical buying paths of B2B customers from historical customer event sequences. Due to the complexity of the B2B marketing processes, as well as the difficulty of manually annotating the great variety of customer activities, a large number of symbols is often needed to represent the sequential data. This is known as the "curse of cardinality", which can impose significant challenges to the design of sequential analysis methods from several perspectives.

- **Complexity**. The computational complexity of finding frequent sequential patterns is huge for large symbol sets. Many existing algorithms have a time complexity that grows exponentially with decreasing pattern supports.

- **Rareness**. In general, the support of a specific sequential pattern decreases significantly with the growing cardinality. To see this, let us consider k symbols that appear with uniform probability in a sequence. The possibility of locating a particular pattern of length ℓ is ℓ^{-k}. In other words, the higher the cardinality, the rarer the patterns are. Since the number of unique subsequences grows with the cardinality, the number of sequences required to identify significant patterns also tends to grow drastically.

- **Granularity**. A large number of symbols in a sequence can "dilute" useful patterns which themselves exist at a different level of granularity. As we will discuss in more detail later, semantically meaningful patterns can exist at a higher granularity level, therefore pattern mining on the original, huge set of symbols may provide few clues on interesting temporal structures.

- **Noise**. Due to the stochastic nature of many practical sequential events, or the multi-modality of events, useful patterns do not always repeat exactly but instead can happen in many permutations. For example, the customers may accidentally download some trial products by mistake when they are looking for the desired information. Without dealing with such irregular perturbations, we may fail to discover some meaningful patterns.

In the literatures, there have been some related works on how to reduce the cardinality in pattern mining by performing a grouping operation on the original symbols. A commonality of these approaches is that they all exploit extra knowledge associated with the symbols as a guidance to perform clustering. For example, a taxonomy of the items may already exist in the form of domain knowledge [Srikant and Agrawal, 1995] or can be derived from the structured description of the product features [Han and Fu, 1999]. In Giannotti et al. [2007c], the 2-dimensional coordinates of spatial points are used to group them into regions to further facilitate the finding of the trajectory patterns. Generally speaking, these approaches first apply clustering on the items whose features are relatively easy to extract, and then search the patterns in different clustering levels.

While these methods have been successfully applied in some application scenarios, there are some emerging issues to be addressed when we face the overwhelming scale and the heterogeneous nature of the sequential data. First, in some applications, it might be difficult to obtain the knowledge of symbols. For example, many sequential data simply use an arbitrary coding of events either for simplicity or security reasons. Second, there are circumstances where it is difficult to define distance among symbols, and therefore clustering becomes impractical. For example, it is unclear how to define the distance between actions customers have taken in their purchasing process. Finally, the biggest concern is that the grouping in these methods is performed irrespective of the temporal content. As a result, these methods may not be able to find statistically relevant temporal structures in sequential data. Therefore, there is a need to develop a new vision and strategy for sequential pattern mining.

To this end, this chapter proposes a temporal skeletonization approach to proactively reduce the representation of sequences, so as to expose their hidden temporal structures. Our goal is to make temporal structures of the sequences more concise and clarified. Our basic assumption is the existence of symbolic events that tend to aggregate temporally. Then, by identifying temporal clusters and mapping each symbol to the cluster it belongs to, we can reduce not only the cardinality of sequences but also their temporal variations. This allows us to find interesting hidden temporal structures which are otherwise obscured in the original representation.

Exploring temporal clusters from a large number of sequences can be challenging. To achieve this, we have resorted to graph-based manifold learning. The basic idea is to summarize the temporal correlations in the data in an undirected graph. The "skeleton" of the graph (i.e., the temporal clusters) can then be extracted through the graph Laplacian, which serves as a higher granularity where hidden temporal patterns are more likely to be identified. A nice interpretation of such temporal grouping is that when individual symbols are replaced by their cluster labels, the averaged smoothness of all sequences is maximized. Intuitively, this can greatly improve the possibility of finding significant sequential patterns. In addition, the embedding topology of the graph translates the rich temporal content of symbolic sequences into a metric space for easier analysis. Compared with existing methods that reduces the cardinality via feature-based clustering, our approach does not require specific knowledge about the items. Instead, it caters directly to the temporal contents in the sequences. To the best of our knowledge, using the temporal correlations to perform clustering and reduction of representation is a novel approach in sequential pattern mining.

Temporal skeletonization can be deemed as a transformation that maps the temporal structures of sequences into the topologies of a graph. Such a dual perspective provides not only more insights on pattern mining, but also brings powerful new tools for analysis and visualization. For example, many techniques in graph theories can be used to analyze symbolic sequences, which appear as random walks on the created graph. On the other hand, due to the explicit embedding, symbolic sequences are represented as numerical sequences or point clouds in the Euclidean space, for which visualization becomes much more convenient.

Experimental results on real-world data have shown that the proposed approach can greatly alleviate the problem of curse of cardinality for the challenging tasks of sequential pattern mining and clustering. Also, we show that it is convenient to visualize sequential patterns in the Euclidean space by temporal skeletonization. In addition, the case study on a Business-to-Business (B2B) marketing application demonstrates that our approach can effectively identify critical buying paths from noisy marketing data.

2.2 Temporal Skeletonization

In this section, we introduce the detail of the proposed method. The key concept is the "temporal cluster", namely group of symbols which tend to aggregate more closely together in the sequences. By transforming the sequential data into graphs, we can identify such temporal clusters to simplify the representation of the sequences.

2.2.1 Temporal Clusters

We believe that temporal clusters often exist in practical sequential data. Otherwise, if there is no such "preferential" structures and everything becomes uniform, we may not find anything interesting. In the following, we discuss two typical scenarios. One involves stage-wise patterns where each stage can be deemed as a temporal cluster; another scenario involves associative patterns.

Case I: Stage-Wise Patterns

First, some sequential processes exhibit stage-wise behaviors; that is, the process typically goes through a number of stages before reaching the final goal, with each stage marked by a collection of representative events. For example, in B2B markets, the business customer will go through stages such as "Investigating product information", "Trial experience and evaluation", "Contacting customer service for more information", and "Contacting sales to finalize the purchase". Here, each stage includes a number of events, and the global structure of the underlying process is shaped by the stages as backbones. Note that the order of stages can vary with regard to different customers. Also, events within a stage may or may not have a dominant ordering. However, collectively, we can observe that each stage forms temporally compact clusters. It is useful to find such clusters to understand the global sequential patterns.

In case of stage-like sequences, it is obviously more meaningful to detect patterns at the stage level. However, the stages are unknown and typically cannot be determined by grouping the symbols based on their features. Therefore, few existing methods could handle such situations. In the following we use one simple example

to show that the large number of symbols in stage-like sequences can "dilute" useful patterns which themselves exist at a different level of granularity, posing a big challenge on existing methods.

$$1 : \mathtt{m}, \mathtt{h}, \mathtt{j}, \mathtt{f}, \mathtt{d}, \mathtt{a}, \mathtt{i}, \mathtt{k}, \mathtt{b}$$

$$2 : \mathtt{j}, \mathtt{l}, \mathtt{m}, \mathtt{a}, \mathtt{n}, \mathtt{f}, \mathtt{b}, \mathtt{o}, \mathtt{g}$$

$$3 : \mathtt{e}, \mathtt{h}, \mathtt{l}, \mathtt{c}, \mathtt{f}, \mathtt{n}, \mathtt{i}, \mathtt{b}, \mathtt{o}$$

$$4 : \mathtt{h}, \mathtt{l}, \mathtt{e}, \mathtt{c}, \mathtt{a}, \mathtt{f}, \mathtt{k}, \mathtt{o}, \mathtt{i}$$

For the four sequences above, if we apply pattern mining on the original level of symbols, we would be unable to find any frequent pattern. However, if we properly group the symbols in the following way

$$\mathtt{A} = \{\mathtt{m}, \mathtt{h}, \mathtt{j}, \mathtt{e}, \mathtt{l}\}$$

$$\mathtt{B} = \{\mathtt{a}, \mathtt{f}, \mathtt{c}, \mathtt{d}, \mathtt{n}\}$$

$$\mathtt{C} = \{\mathtt{k}, \mathtt{g}, \mathtt{o}, \mathtt{b}, \mathtt{i}\}$$

then all the four sequences read as

$$\mathtt{A}, \mathtt{A}, \mathtt{A}, \mathtt{B}, \mathtt{B}, \mathtt{B}, \mathtt{C}, \mathtt{C}, \mathtt{C}$$

It is obvious that $(\mathtt{A}, \mathtt{B}, \mathtt{C})$ is a frequent (stage) pattern with 100% support.

Case II: Frequent Associative Patterns

Temporal cluster also has an interesting connection with frequent associative patterns. Since associative items tend to occur closely to each other, they are temporally more coherent and can likely form temporal clusters. In other words, there must exist temporal clusters if there are significant frequent patterns. However, temporal clusters

can be more general than frequent patterns. Another challenge for finding frequent patterns is the noise in the data. If frequent sub-sequences are somewhat perturbed, special cares have to be taken in finding exact patterns. In comparison, the temporal clusters we try to discover are identified via the temporal distribution of all event pairs, thus our approach is inherently more resistant to noise.

2.2.2 Temporal Graph and Skeletonization

Since large cardinality hampers pattern mining, we propose to find meaningful temporal clusters to alleviate it. The key role of temporal clusters is that they can be used to re-encode the original sequences. Since temporal clusters are composed of symbols that are temporally more coherent, the newly encoded sequences will be temporally smoother than the original sequences. By doing this, we can greatly reduce not only the cardinality but also the temporal variations of the sequences. The latter makes it much easier to find semantically useful patterns. Specifically, we put this under the following optimization framework.

Problem 1 (Temporal skeletonization) *We have sequences* $\{S_n | n = 1, 2, \cdots, N\}$, *where the n-th sequence is* $S_n = (s_1^n, s_2^n, \cdots, s_{T_n}^n)$ *with length* T_n. *We have a set of symbols* $\mathcal{S} = \{e_1, e_2, \cdots, e_{|\mathcal{S}|}\}$, *where* $s_t^n \in \mathcal{S}$. *We want to find a new encoding scheme of the symbols* $e \in \mathcal{S}$, *denoted by the mapping* $f : e \mapsto f(e) \in \{1, 2, \cdots, K\}$, *such that when encoded with* f, *the temporal variation of resultant sequences is minimized*

$$\min_f \frac{1}{N} \sum_{n=1}^{N} \sum_{\substack{1 \le p,q \le T_n \\ |p-q| \le r}} \left(f(s_p^n) - f(s_q^n) \right)^2 . \tag{2.1}$$

Here, the cardinality of the encoding scheme, K, is a pre-defined integer that is much smaller than that of the original representation $|\mathcal{S}|$. Also, r is a pre-defined integer that controls the range that local sequence variations are computed. In other words, in each of the N sequences, we only consider pairs of events s_p^n and s_q^n that are within r intervals to each other, such that when they are re-encoded with $f(s_p^n)$ and $f(s_q^n)$, they are similar to each other.

This is an integer programming problem which has shown to be NP-hard. Therefore, we propose to relax the integer constraint to real numbers. In addition, we will define the so-called "temporal graph" to re-phrase the problem as a graph optimization one.

Definition 1 (Temporal graph) *Let G be a weighted graph $G = \langle V, E \rangle$ with vertex set $V = \mathcal{S}$ and edges E. The i-th node of G corresponds to the i-th symbol e_i in the symbol set $\mathcal{S}$. The weight of the edge between node i and node j is defined as the ij-th entry of an $|\mathcal{S}| \times |\mathcal{S}|$ matrix W, where*

$$W_{ij} = \frac{1}{N} \sum_{n=1}^{N} \sum_{\substack{1 \le p,q \le T_n \\ |p-q| \le r}} [e_i = s_p^n \wedge e_j = s_q^n]. \tag{2.2}$$

We show an example of the temporal graph in Figure 2.1 for the 4 sequences in section 2.2.1. We call $G = \langle V, E \rangle$ "temporal graph", because the edge weight of the graph captures the averaged temporal closeness between any pair of symbols/events across all the input sequences. It is straightforward to show $W_{ij} = W_{ji} \ge 0$, thus G is a symmetric graph. With 1, and let $\mathbf{y} \in \mathbb{R}^{|\mathcal{S}|}$ where $\mathbf{y}_i = f(e_i)$, the objective function

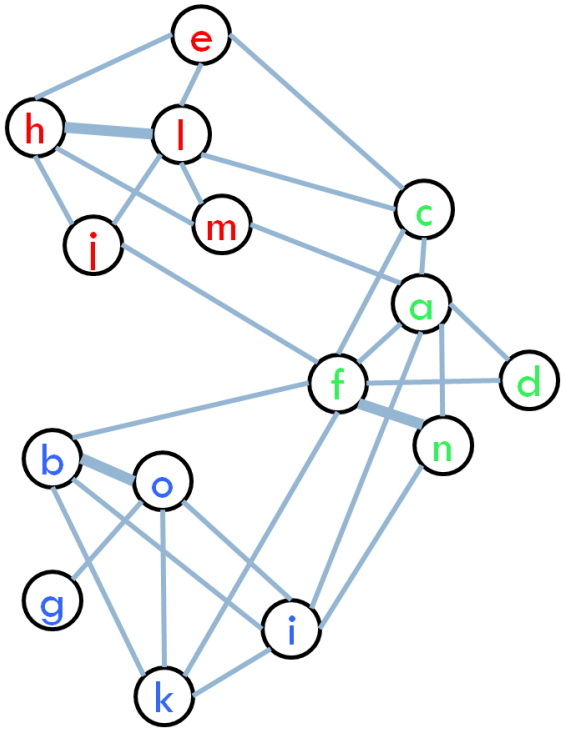


Figure 2.1. An example of the temporal graph (15 vertexes and 29 weighed edges). The 3 bold edges are weighted by 0.5 and the other edges are weighted by 0.25.

in Problem 1 can be written in the following compact form

$$\begin{aligned}
&\frac{1}{N}\sum_{n=1}^{N}\sum_{\substack{1\le p,q\le T_n\\ |p-q|\le r}}\left(f(s_p^n)-f(s_q^n)\right)^2\\
=&\frac{1}{N}\sum_{n=1}^{N}\sum_{\substack{1\le p,q\le T_n\\ |p-q|\le r}}\sum_{i,j}[s_p^n=e_i\wedge s_q^n=e_j]\left(f(e_i)-f(e_j)\right)^2\\
=&\sum_{i,j}W_{ij}(\mathbf{y}_i-\mathbf{y}_j)^2
\end{aligned}$$

Thus, Problem 1 has a standard form of graph-based optimization. Let us define the graph Laplacian of G as $L=D-W$, where D is a diagonal degree matrix with

$D_{ii} = \sum_j W_{ij}$. Then, Problem 1 can be further written as

$$\begin{aligned} \min_{\mathbf{y}\in\mathbb{R}^{|\mathcal{S}|}} \quad & \mathbf{y}'(D-W)\mathbf{y} \\ s.t. \quad & \mathbf{1}'D\mathbf{y} = 0 \\ & \mathbf{y}'D\mathbf{y} = 1 \end{aligned} \tag{2.3}$$

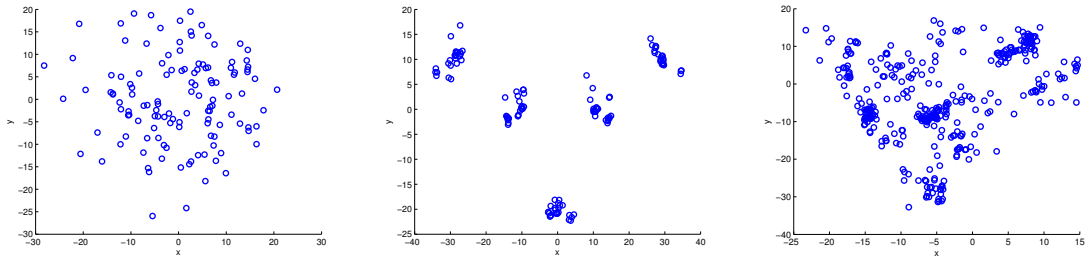
where $\mathbf{1}$ is a vector of all 1's. Here, the translation and scale constraints are added to avoid trivial solutions. This is also known in the literatures as Laplacian eigenmap [Belkin and Niyogi, 2001], which has also been applied in spectral clustering [Ng et al., 2002]. To the best of our knowledge, it is a novel application to use graph-based algorithm to extract temporal structures from multiple sequences.

Note that the more often symbol e_i and symbol e_j appear close to each other in the sequences, the higher the W_{ij} is and the larger the penalty it induces on the objective (Equation 2.3), and as a result, the closer $\mathbf{y}_i$ and $\mathbf{y}_j$ should be. This equivalently achieves a grouping of the symbols, which are the temporal clusters we try to extract. As can be expected, by re-encoding the sequence with the label of the temporal clusters, we can improve the temporal smoothness which is beneficial to subsequent pattern mining.

2.2.3 Generalizations of Temporal Graph

In practice, instead of only considering event pairs that are within r intervals, one can also use a smoother function such as

$$W_{ij} = \frac{1}{N}\sum_{n=1}^{N}\sum_{1\le p,q\le T_n} \rho_r\left(|p-q|\right)\left[e_i = s_p^n \wedge e_j = s_q^n\right], \tag{2.4}$$



(a) Random sequences. (b) Stage-wise sequences. (c) Customer event seqs.

Figure 2.2. The embedding of symbols in different types of sequence data.

where ρ_r is a non-increasing function parametrized by r. The Equations 2.2 and 2.4 are equivalent with $\rho_r(u) = [u \leq r]$. For an example of smoother function, we can use the exceedance of the Exponential distribution $\rho_r(u) = \exp(-u/r)$.

Moreover, we can also construct the temporal graph using the exact event happening time if it is recorded in the sequences. Let the happening time of s_l^n be t_l^n, then we extend the construction in Equation 2.4 to

$$W_{ij} = \frac{1}{N} \sum_{n=1}^{N} \sum_{1 \leq p,q \leq T_n} \rho_r \left(|t_p^n - t_q^n| \right) [e_i = s_p^n \wedge e_j = s_q^n]. \tag{2.5}$$

It is easy to see that, when $\rho_r(d) = [d \leq r]$, the skeletonization coding scheme derived from Equation 2.5 is optimal with respect to

$$\min_f \frac{1}{N} \sum_{n=1}^{N} \sum_{\substack{1 \leq p,q \leq T_n \\ |t_p^n - t_q^n| \leq r}} \left(f(s_p^n) - f(s_q^n) \right)^2.$$

Thus, we consider pairs of events s_p^n and s_q^n that happened within r time period, such that they have similar encoding $f(s_p^n)$ and $f(s_q^n)$.

2.2.4 Embedding and Visualization

The optimal solution of Problem 2.3 is the eigenvector of the graph Laplacian corresponding to the second smallest eigenvalue. In practice, one usually computes several

(e.g., d) eigenvectors as columns in $\mathbf{y} \in \mathbb{R}^{|\mathcal{S}| \times d}$, and then applies clustering algorithms (subsection 2.2.5) with it to identify temporal clusters. The useful eigenvectors of the graph Laplacian not only provide a relaxed solution of temporal clusters, but also more interestingly, naturally connect to the manifold embedding of the graph.

Note that the eigenvectors of the graph can be deemed a low-dimensional embedding, in which the proximity relation among objects preserves that in the original space [Yan et al., 2007]. Since the similarity measurements in W_{ij} of the graph reflect the temporal closeness of the events, the embedding eigenvectors of the graph will also inherit this configuration. Namely, if two symbols, e_i and e_j are temporally more related, their distance will also be small in the embedded space. In this way, our approach provides a direct platform for visualizing the temporal structures of sequential data. We believe that such visualization can provide interesting insights allowing domain experts to draw useful conclusions.

To provide more intuition on the temporal embedding results, in Figure 2.2, we give several examples. Figure 2.2a is the embedding of a collection of random sequences. As can be seen, the embedded symbols (each represented by one point) are distributed uniformly and there is hardly any interesting structure. Figure 2.2b is a simulated data containing 5 stages of events (more details in section 2.5). As can be seen, there are clear clusters in the embedding, each representing exactly events belonging to one stage. In Figure 2.2c we used a real-world data set composed of thousands of B2B customer event sequences. As can be seen, the cluster structures are complicated: some clusters are well separated while others are diffusing. This has to do the complicated relationships between practical events in the data set. From

these examples, we can see that our temporal skeletonization approach can translate the temporal structures in the sequential data into their topological counterparts. The resultant visualization can bring useful insights.

In the literatures, there are many algorithms for manifold learning. Many of these approaches rely on the eigenvalue decomposition of a similarity matrix to obtain the manifold embedding. For example, Isomap [Tenenbaum et al., 2000] is another popular method that embed a graph into an Euclidean space. In our experiments, we also use Isomap to visualize the data, and find that it can provide spatially more unfolded embedding.

2.2.5 Temporal Clustering

There are several simple clustering algorithms available to identify the temporal clusters in the low-dimensional embedding space of the temporal graph, e.g., K-means, GMM (Gaussian mixture model), and Mean-Shift clustering. Among them, we recommend the Mean-Shift clustering [Cheng, 1995], which does not require prior knowledge of the number of clusters, does not constrain the shape of the clusters, and works efficiently in the low-dimensional space. Given the embedding $\mathbf{y} \in \mathbb{R}^{|\mathcal{S}| \times d}$, and a kernel density estimated with a kernel κ and bandwidth h

$$p(\mathbf{y}) = \frac{1}{h^d |\mathcal{S}|} \sum_i \kappa(\frac{\mathbf{y} - \mathbf{y}_i}{h}), \tag{2.6}$$

the Mean-Shift clustering iteratively translate y to the *kernel weighted mean*

$$\hat{\mathbf{y}} = \frac{\sum_i \kappa(\frac{\mathbf{y}-\mathbf{y}_i}{h}) \mathbf{y}_i}{\sum_i \kappa(\frac{\mathbf{y}-\mathbf{y}_i}{h})}.$$

This procedure is guaranteed to converge to the maximas of the data density, and the points which converge to the same maxima are associated with the same cluster. We use this clustering solution as the coding scheme in Problem 1, such that $f(e_i) = k$ if and only if the $\mathbf{y}_i$ belongs to the k-th cluster in the embedding space. Accordingly, we define the k-th temporal cluster $\mathcal{S}_k = \{e : f(e) = k\} \subset \mathcal{S}$.

2.2.6 Post-Temporal-Smoothing

By finding temporal clusters in the embedded Euclidean space, and use it to re-encode the sequence, we can obtain temporally smoother representation. For example, we can transform the original customer event sequences to sequences of stages, with each stage being defined as the groups of symbols (marketing campaigns) in the temporal clusters identified. However, although the embedded graph is estimated robustly with the integrated data from all sequences, the individual sequence might still be noisy in some cases, for instance, the order parameter r is chosen too small. Thus, one might want to further smooth away the irregularities in the sequences. To this end, we propose a multi-series post-smoothing using fused lasso [Tibshirani et al., 2005].

Specifically, we first compute the a partition matrix $Y \in \mathbb{R}^{|\mathcal{S}| \times K}$ where Y_{sk} is the probability that the symbol $s \in \mathcal{S}$ belongs to the k-th temporal cluster. For the Mean-Shift clustering, we compute

$$Y_{sk} \propto \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \kappa(\frac{\mathbf{y}_s - \mathbf{y}_i}{h}),$$

where $\mathcal{S}_k$ is the k-th cluster. These probabilities are normalized with $\sum_k Y_{sk} = 1$.

We can use this partition matrix Y to transform each individual sequence $S_n = (s_1^n, s_2^n, \cdots, s_{T_n}^n)$ into a multiple of K sequences, denoted by $Y^n \in \mathbb{R}^{T_n \times K}$ where Y_{tk}^n

is the probability that the t-th interaction s_t^n in S_n belongs to the k-th cluster. Then we try to find a smoother version of the multiple sequences Y^n, denoted by X^n. To achieve this, we encourage sparsity of the differences between the successive rows in X^n, i.e., $\sum_{t=1}^{T_n-1} \|X_t^n - X_{t+1}^n\|_1$ where $\|X_t^n - X_{t+1}^n\|_1 = \sum_{k=1}^{K} |X_{tk}^n - X_{(t+1)k}^n|$. We optimize the approximation of X^n by maximizing the alignment $\sum_{t=1}^{T_n} \sum_{k=1}^{K} X_{tk}^n Y_{tk}^n$ and with probability constraints. Thus, we would like to maximize $\sum_{t=1}^{T_n} \sum_{k=1}^{K} X_{tk}^n Y_{tk}^n$, subject to

$$\frac{1}{T_n - 1} \sum_{t=1}^{T_n-1} \|X_t^n - X_{t+1}^n\|_1 \leq \lambda, \tag{2.7}$$

in addition to $\sum_{k=1}^{K} X_{tk}^n = 1$ and $X_{tk}^n \geq 0$. Here λ is a tuning parameter controlling smoothness of X^n. An example with the noisy matrix Y^n and the smoothed approximation X^n is shown in Figure 2.3.

To solve the approximation problem with the smooth constraint, we let $X_{tk}^n - X_{(t+1)k}^n = \alpha_{tk}^n - \beta_{tk}^n$ for $t = 1, \cdots, T_n - 1$ and $k = 1, \cdots, K$ where $\alpha_{tk}^n, \beta_{tk}^n \geq 0$. Let $A^n \in \mathbb{R}^{(T_n-1)\times T_n}$ with $A_{tt}^n = 1$, $A_{t(t+1)}^n = -1$ and $A_{tt'}^n = 0$ otherwise so that $A^n X^n = \alpha^n - \beta^n$. We can rewrite the smooth approximation as a linear programming problem:

$$\begin{aligned} \max \quad & \sum_{t=1}^{T_n} \sum_{k=1}^{K} X_{tk}^n Y_{tk}^n, \\ s.t. \quad & \sum_{k=1}^{K} X_{tk}^n = 1, \forall t = 1, \cdots, T_n, \\ & A^n X^n = \alpha^n - \beta^n, \\ & \frac{1}{T_n - 1} \sum_{t=1}^{T_n-1} \sum_{k=1}^{K} (\alpha_{tk}^n + \beta_{tk}^n) \leq \lambda, \end{aligned}$$

in addition to the non-negativity $X^n, \alpha^n, \beta^n \geq 0$.

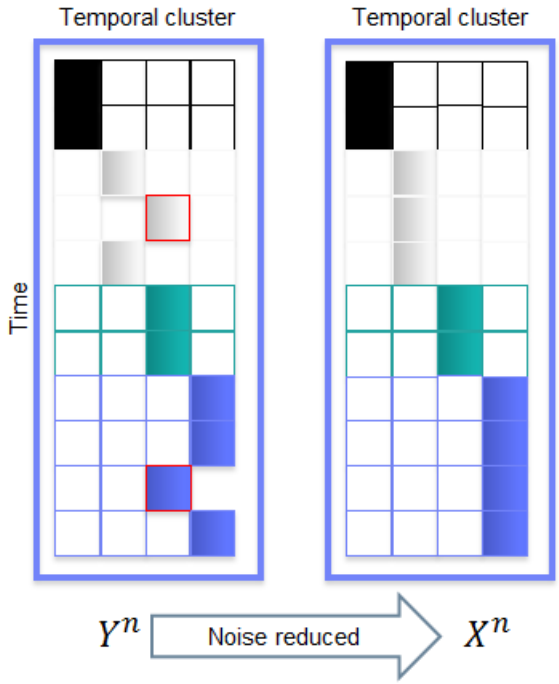


Figure 2.3. An example of the post-temporal-smoothing. The two noisy stages in red boxes are removed.

2.3 Parameter Selection

Now we summarize the important parameters in our temporal skeletonization approach, and we provide principled guidance on selecting the appropriate parameter values, including the temporal order parameter, the mean-shift clustering kernel and bandwidth, and the degree of post-temporal regularization.

2.3.1 The temporal order parameter

The level of smoothness can be adjusted effectively by the order parameter r, which controls the resolution on which clusters are extracted. A larger r captures the similarities among events in a longer temporal range, which potentially increase the connectivity of the temporal graph and lead to fewer clusters, while a small r only considers directly adjacent symbols as similar, which make the temporal graph more spread and lead to more clusters. In the extreme case when r approaches infinity, all appearing events will be fully connected. Indeed, the order parameter r is related to the progression rate of the underlying sequences, thus the domain knowledge can be used for selecting a appropriate value for r. For example, in a specific application, if we know there is little correlation between events happened with time gap larger than some period, namely r, and we have the happening time recorded, we can use the application specific parameter r in Equation 2.5.

2.3.2 The Mean-Shift kernel and bandwidth

In the Mean-Shift clustering, the choice of the kernel function κ is not crucial to the accuracy of the kernel density estimation and clustering, so we use the Gaussian

kernel function $\kappa(u) = \frac{1}{\sqrt{(2\pi)^d}} \exp(-\frac{\|u\|^2}{2})$, for which, the optimal choice for bandwidth is $\hat{h} = (\frac{4}{3n})^{1/5}\sigma$ where σ can be the standard deviation of the pairwise distances [Silverman, 1986]. When the kernel κ and bandwidth h are selected, the Mean-Shift clustering can be applied automatically without knowing the number of clusters K. Note that, when introducing the temporal skeletonization, we assumed that we have a given K, which can also be determined according to domain knowledge. In this case, a mixed strategy to determine the bandwidth h can be used: we search the optimal bandwidth between $0.25\hat{h}$ and $1.5\hat{h}$ to get the clustering solution with the number of clusters close to K.

2.3.3 The post-temporal-smoothing regularization

In the final step when transforming the raw sequences into sequences of temporal clusters, we use the post-temporal-smoothing to further remove the noises in individual sequences. For an appropriate degree of the smoothing regularization, we suggest

$$\lambda = \alpha \frac{K-1}{T_n - 1}. \tag{2.8}$$

Intuitively, let X_t^n be the hard clustering indicator of s_t^n ($X_{tk}^n = 1$ if and only if s_t^n is from the stage k), then $\sum_{t=1}^{T_n-1} \|X_t^n - X_{t+1}^n\|_1$ is the times of stage transitions in the sequence s_n. Suppose the sequence went through all the K stages, with one stay in each stage, then we have $\sum_{t=1}^{T_n-1} \|X_t^n - X_{t+1}^n\|_1 = K - 1$. For some sequences, the number of stages visited during the observation period may be less than $K-1$, so $\lambda = \frac{K-1}{T_n-1}$ is a realistic upper bound in Equation 2.7. We also suggest $\alpha > 1$ in Equation 2.8 to allow some perturbation, e.g., $\alpha = 1.2$.

2.4 Applications

In this section, we discuss the applications of the temporal skeletonization method in several interesting problems. The reduced representation makes it much easier to perform these tasks than on the original sequences.

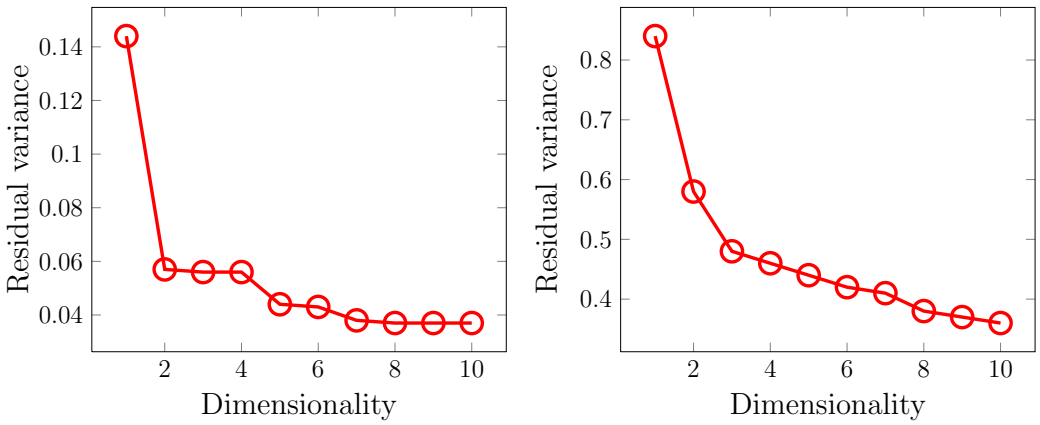
2.4.1 Sequence Visualization

Our framework embeds symbolic events in sequences into an Euclidean space, which allows to visualize each sequence as a trajectory. Such visualization can provide insights on the relationship between events, which, when subject to examination of domain expert, can greatly facilitate them making analysis and decisions. In section 2.6, we will report how temporal clusters in the customer event data can help to understand purchase patterns.

In choosing the embedding dimensions, we typically choose two or three dimensions, which correspond to the dominant components of the temporal graph. Usually, these few dimensions can encode a sufficient amount of the temporal relations. To see this, we show the residual variance in the embedding with regard to the number of selected embedding dimensions. As shown in Figure 2.4, in both the simulation and real-world data sets, the residual variance drops most significantly with the first few dimensions.

2.4.2 Sequential Pattern Mining

Our method transforms the original sequence of events to the sequence of temporal clusters (the cluster labels are used as a new set of symbols to encode events). This



(a) Simulation data. (b) B2B customer data.

Figure 2.4. The residual variance vs. dimensionality.

helps to reduce the cardinality of the sequence representation, and the supports of the sequential patterns are increased. As a result, consequent sequential pattern mining is able to discover significant knowledge which otherwise would be diluted in the raw data. Indeed, the patterns discovered in this way are defined with a higher level of granularity, i.e., the temporal clusters. Therefore, to interpret the patterns, we can first annotate the temporal clusters with domain knowledge.

For example, in the customer event data, the temporal event clusters can be semantically labelled as Tradeshow, Direct Mail Ads, Web Ads, Trial Product Download, and Unsubscribe, etc. As we shall see, these temporal clusters correspond to stages in the purchasing route of the customers, which are much easier to understand and interpret compared with the raw sequences.

2.4.3 Sequence Clustering

Sequence clustering is an important task, however, it is not always easy to extract appropriate features or define distances among sequences so that clustering can be

performed properly. This is particularly true when sequences are represented by a huge number of symbols. The temporal skeletonization method we have proposed can be used to tackle these difficulties. This is because the temporal skeletonization can remove noises in the sequences based on their collectively temporal behaviours. More importantly, it re-summarizes the events in the form of groups of events, therefore we will observe much more repeated subsequence on which sequential features can be more meaningful.

For example, when there is only a reasonably small number of symbols in the sequences, we can extract the following useful features, such as the counts of each temporal cluster passed by the sequence, or how many times one symbol appears in precedence of another, and so on. It turns out such a straightforward approach can effectively cluster the sequences. To incorporate more temporal information, we can also leverage the frequent sequential patterns discovered by the aforementioned sequential pattern mining, as suggested by Lee et al. [2011].

2.5 Empirical Evaluation

In this section, we evaluate the performances of our approach in comparison with several state-of-the-art methods. All the experiments are performed on a GNU/Linux system with 8 CPUs (Intel i7 2.93GHz) and 8G RAM.

2.5.1 Synthetic Data

We have simulated symbolic sequential data composed of stages of events. We define 5 stages $\{A, B, C, D, E\}$, where each contains 25 symbols. Then, we create 5000 sequences that are of two patterns. The first 2500 sequences mainly follow stage

pattern $A \rightarrow B \rightarrow C \rightarrow D$; the other 2500 sequences follow $B \rightarrow E \rightarrow C$. The simulation proceeds as follows. After deciding which stage to sample from based on the two patterns, we randomly pick d symbols from that stage, where d is a random integer. Then, we inject the selected symbols into the sequence, and continue to the next stage in the pattern. Indeed, such a simulation process is equivalent to a standard Hidden Markov Model (HMM), where 5 stages correspond to 5 hidden states and symbols within each stage correspond to observations. Let the transition probability from each stage to itself be p, and that to the next stage (as specified in the two patterns) be $1-p$. Then, the stage duration d follows a geometric distribution $d \sim (1-p)p^{d-1}$, with the expected value $\mathbb{E}[d] = \frac{1}{1-p}$. To have significant stage-wise patterns in the produced sequences, we have used a large probability $p = \frac{14}{15}$, leading to $\mathbb{E}[d] = 15$. In other words, on average, we randomly pick 15 symbols for each stage.

2.5.2 Baselines

First, we apply state-of-the-art Frequent Sequence Mining (FSM) algorithms, including GSP [Srikant and Agrawal, 1996], SPADE [Zaki, 2001], PrefixSpan [Han et al., 2001], SPAM [Ayres et al., 2002]. The results in Figure 2.5 show that, when desired pattern support drops, the time consumption of these algorithms grow super-exponentially, indicating the difficulties introduced by the large numbers of symbols. The number of detected patterns also becomes explosive, most of which are non-informative and provide no clear insight of the underlying sequence generating processes (as shown in Table 2.1). In comparison, using the temporal clusters identified via our approach (more details in subsection 2.5.3), the mining process succeeds

quickly in about one second.

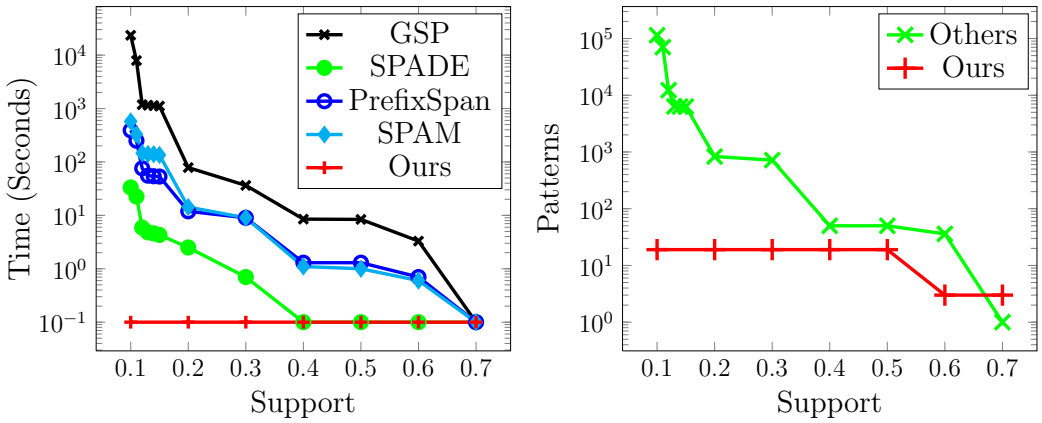


Figure 2.5. FSM algorithms on the simulated data.

In addition to the improvement on efficiency, we also compare the pattern mining results on the original and the re-encoded sequences via our method in Table 2.1. For the task of pattern mining, we compute the precision (fraction of discovered patterns that are relevant) and recall (fraction of the relevant patterns that are discovered) of the discovered patterns against the ground truth. The results show that when working on the raw data, FSM performs poorly with an F-measure around 0.281. In contrast, after re-encoding using our approach, it can lead to an 100% accuracy.

Task	Pattern Mining		Sequence Clustering		Stage Recovery	
Method	FSM	Ours	HMM	Ours	HMM	Ours
Precision	0.725	**1.0**	0.997	**1.0**	0.488	**1.0**
Recall	0.174	**1.0**	0.997	**1.0**	0.448	**1.0**

Table 2.1. Utility comparison on the simulated data.

Since data simulation process follows the Markov property, the second baseline

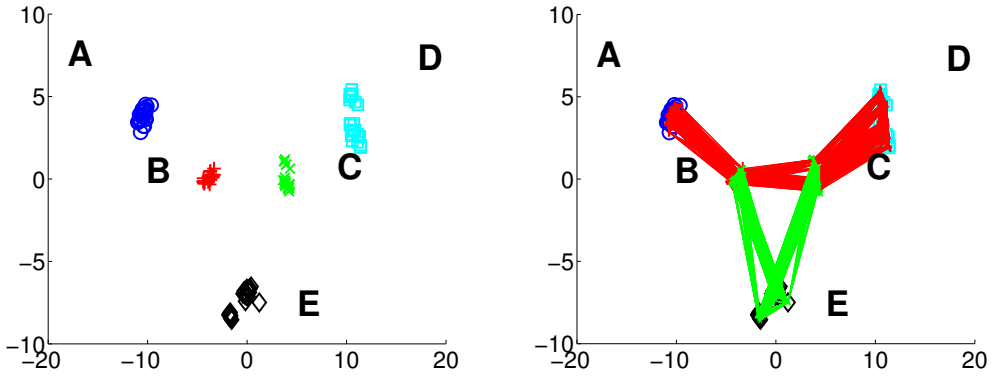
approach we have experimented with is the classical HMM. In our data, there are two hidden patterns, thus we adopt the HMM based clustering (HMMC) [Owsley et al., 1997] to simultaneously cluster the sequences and estimate the HMM parameters for each cluster. Specifically, to group sequences into M clusters, the HMMC randomly allocates all sequences to M disjoint subsets as initial clusters, then the following two procedures are iterated until convergence. First, for all sequences in cluster C_m, we estimate a transition matrix ϕ_m and a emission matrix θ_m; Second, we reallocate each sequence S_n to the cluster C_m on whose transition and emission matrices it has the highest probability of being produced, i.e., $m = \arg\max_m \Pr(S_n|\phi_m, \theta_m)$.

We have provided the HMMC method with some ground truth parameters, i.e., the number of clusters $M = 2$, and the number of hidden states (stages) for each cluster. Table 2.1 shows the accuracy of the HMM based method for the task of sequence clustering and stage recovery. To be specific, for these two tasks, we first compute the so-called confusion matrix C, where C_{ij} is the number of instances in resulted group i and ground truth class j. Then, the precision is computed as $\frac{1}{N}\max_\sigma \sum_j C_{\sigma(j)j}$ where $N = \sum_{ij} C_{ij}$ and σ maps classes to different groups; the recall is computed as $\frac{1}{N}\sum_i \max_j C_{ij}$, which is also termed as clustering purity. Again, our method gives perfect results on both tasks. The HMMC only works well for sequence clustering, while it performs almost randomly for the stage recovery.

2.5.3 Our Results

Our approach can successfully recover patterns hidden in the sequences. In Figure 2.6a, we see that our approach embeds altogether 125 symbols in such a way that

5 dominant clusters emerge. This is in perfect consistency with the ground truth structure we have used in the simulation. In Figure 2.6b, we can also correctly group the 5000 sequences into two clusters (in red and green), by extracting simple features as discussed in subsection 2.4.3.



(a) The temporal clusters. (b) The sequence clusters.

Figure 2.6. The embedding of simulated data.

Our approach does not require any prior knowledge on the simulated data. We tried different ways to construct the temporal graph, and the results were robust with respect to parameters (e.g., $1 \leq r \leq 5$ in both Equation 2.2 and Equation 2.4). Also, we clearly identify the temporal clusters as stages based on well clustered symbols.

2.5.4 Noisy Cases

Now we examine the performances of our approach in case of noisy data. We have injected two types of noises. The first is introduced on items of each sequence, such that one stage could contain symbols that belong to other stages. Such a noisy behavior is quite natural in the buying process of customers. For example, customers might occasionally participate events not very relevant to their current buying stages.

The second kind of noise is introduced as random sequences not following any certain patterns. In real world, these random sequences might correspond to event sequences of some unintended customers. We have 5% noisy observations for each type of noise and two more stage-wise patterns making the problem more challenging.

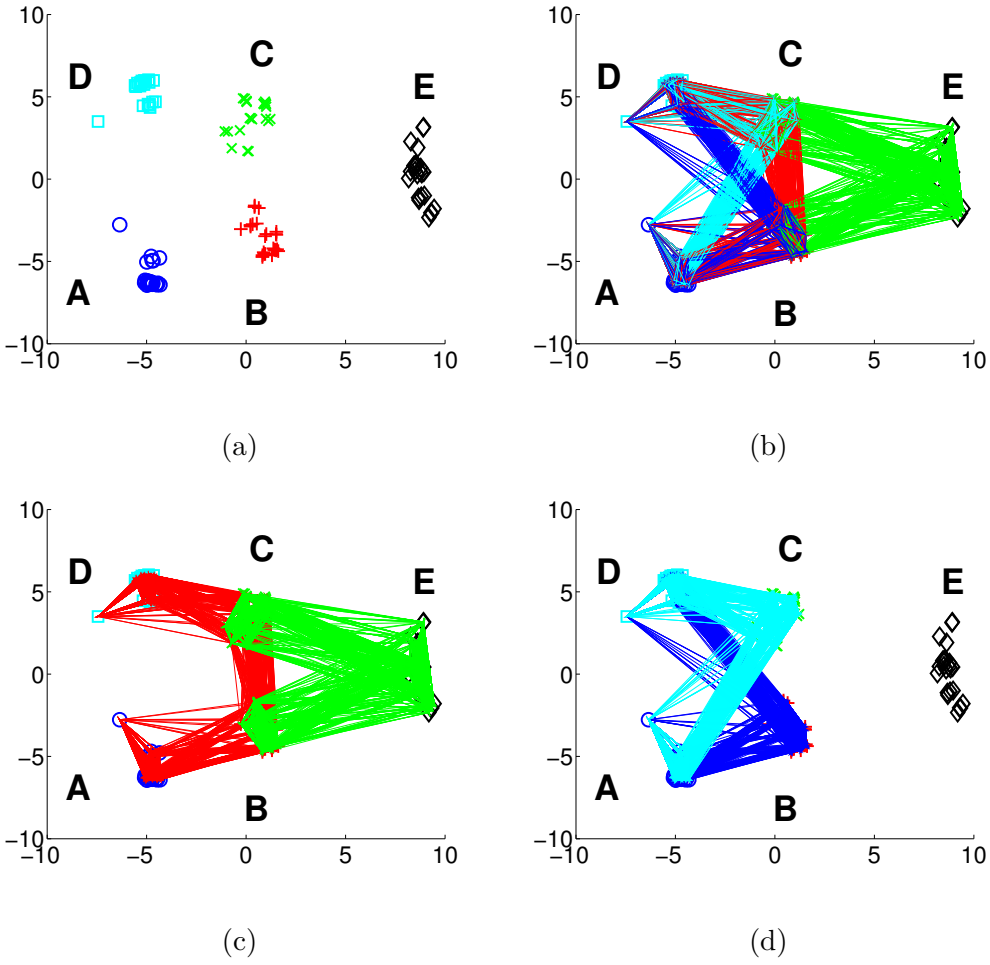


Figure 2.7. The noisy simulated data. a: The temporal clusters. b: The sequences clusters. c: Pattern (A, B, C, D) and (B, E, C). d: Pattern (A, B, D) and (A, C, D).

We can see from Figure 2.7a that, even in this noisy data, the temporal clusters are still identifiable, which correspond to stages of events. The reason is that, our temporal graph is estimated robustly with integrated temporal content from all the

sequences, therefore a small portion of individual noisy observations cannot significantly affect the result. As shown in Figure 2.7b, once noisy random events are removed, we can discover important patterns (groups of sequences). Indeed, the 4 stage-wise patterns are all discovered by our approach.

2.6 B2B Purchase Pattern Analysis

In this section, we apply our method to find critical buying paths of Business-to-Business (B2B) buyers from historical customer event sequences. Since B2B purchases are often involved with strategic development of the company, and as a result, extra cautions and extensive research efforts have to be taken in making such investment, the decision process of customers in purchasing certain products or services is much more complicated than that in our daily purchasing activities. Thus, it is of significant business value if we can discover characteristic and critical buying paths from observations. These can be used to recommend directed advertising campaigns so as to increase potential profits and also reduce the marketing cost. In addition, we would also like to visually display the buying processes of the customers. By doing this, we can better understand the customer behavior patterns and accordingly develop promising marketing strategies. In the following, we show that our framework is effective for these objectives.

2.6.1 Data Description

We have collected huge amount of purchasing event data for the customers of a big company. In more detail, we have event sequences from $N = 88040$ customers, with the number of unique events (symbols) $|\mathcal{S}| = 5028$, leading to altogether $T = \sum_n T_n =$

248725 event records. We construct the temporal graph using Equation 2.4 and $r = 5$, and then embed the graph using Isomap.

2.6.2 Embedding Results and Buying Stages

We plot the embedding of selected 503 events (top 10% nodes with the largest degrees in the temporal graph) in Figure 2.8, and mark it with the clustering results. For each detected cluster, we are able to extract dominant semantic keywords for the events in that cluster, as shown in Table 2.2. Note that the semantic information here is only used to summarize each temporal cluster for better understanding of our results, but not for the purpose of grouping the events.

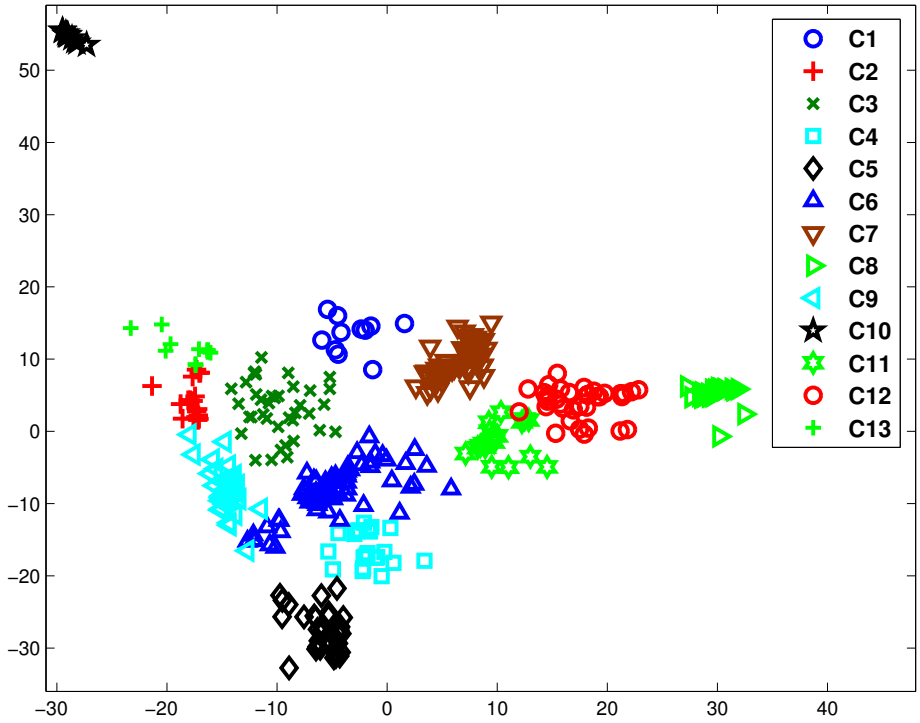


Figure 2.8. The customer event clusters.

We discuss some interesting observations on the temporal clusters. First, clusters appearing close to each other are logically related, e.g., 'search engine' (C_{13}) and 'trial

Cluster	Top keywords	Size
C_1	`Official Website`	12
C_2	`Corporate Event, Marketing Mail`	20
C_3	`Trial Product Download`	45
C_4	`Conference`	27
C_5	`Unsubscribe`	38
C_6	`Webinar`	101
C_7	`Trial Product Download`	70
C_8	`Tradeshow`	37
C_9	`Corporate Event, Marketing Mail`	65
C_{10}	`Web Marketing Ads`	13
C_{11}	`Webinar`	21
C_{12}	`Webinar`	42
C_{13}	`Search Engine`	12

Table 2.2. The semantic annotation of event clusters.

product download' (C_3); 'webinar' (C_{12}) and 'trade show' (C_8). Second, symbols with the same semantic meaning may not be in the same temporal cluster. For example, C_6, C_{11}, C_{12} are all marked with 'webinar' but they form 3 clusters. Note that these three clusters are close to 'direct marketing mail', 'trial product download', 'trade show', respectively, indicating that they have different levels of maturity towards final purchase. Thus, it is reasonable to have them separated. Nevertheless, the three clusters are still neighbors, after all, since they have the same nature ('webinar'). In

other words, temporal clusters could be partially consistent with attribute-based clusters, while meanwhile revealing more fine-grained structure by exploiting the temporal correlations. This is where the extra value comes from.

2.6.3 Critical Buying Paths

With the detected temporal clusters, we can transform the original event sequences to sequences of temporal clusters, and apply the FSM algorithms on the re-encoded sequences. For the sequence transformation, we also apply the post-temporal-smoothing, since the data set we collected is very noisy and can be subject to human errors. Figure 2.9 reports some results of pattern mining on the original and re-encoded sequences respectively. The pattern supports are generally chosen much smaller than those used in the simulation study, since we have much more symbols here and more complicated patterns. As can be observed, on the original sequences, the number of identified patterns again grows super-exponentially with decreasing support. This would indicate that it is very hard to have a conclusive and comprehensive understandings of the customers' purchasing patterns. In contrast, using the transformed sequences via temporal skeletonization, the number of patterns is more reasonable.

We then perform clustering on the transformed sequences, using the frequent patterns detected to extract features as discussed in subsection 2.4.3. We focus on a few dominant clusters in which the customers have relatively longer event sequences. The remaining customers only participated events in one or two buying stages and their behaviors are almost random. In Figure 2.10, we plot the dominant sequence clusters covering 3501 customers, by connecting each event of the sequence embedded

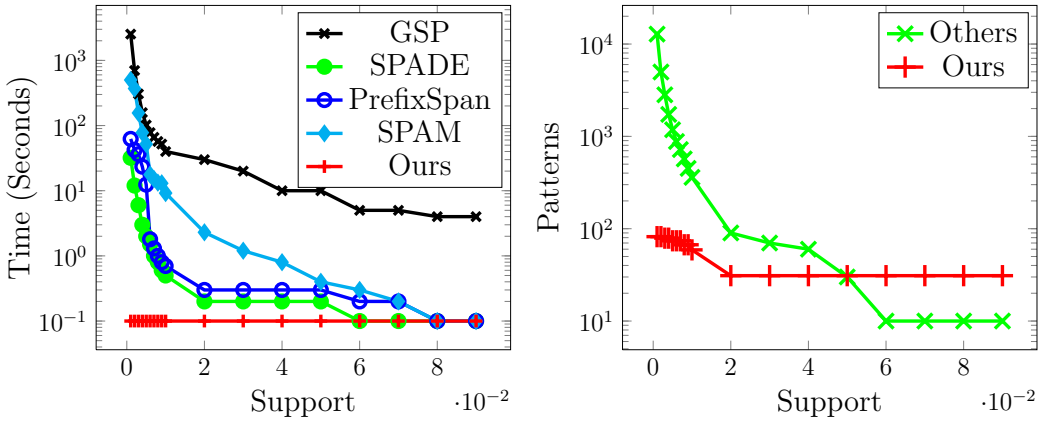


Figure 2.9. Sequential patterns in B2B customer event data.

in the two-dimensional plane. Here, each cluster corresponds to one type of customers with a unique buying path. We summarized these paths in Table 2.3.

With the semantic annotation in Table 2.2, we can see that the temporal clusters can be used to reveal several interesting buying paths. For example, the blue path P_1 passes through 5 clusters (or stages), as $C_{10} \rightarrow C_1 \rightarrow C_7 \rightarrow C_{12} \rightarrow C_8$. These customers were attracted by 'Web Marketing Ads', then they went to 'Official Website' and found the 'Trial Product Download'. When customers needed more information to make decisions, they continued to attend 'Webinar' and finally went to 'Tradeshow'. The green path P_2 also ends with 'Trade Show', but starts with 'Search Engine', indicating that these customers started from their own effort in acquiring information of the product suiting their needs. In comparison, the pattern in the red path P_3 is simpler, starting with 'Webinar' and ending with 'Tradeshow'. All these three paths can be grouped into the 'Successful' class ('+' in Table 2.3), which leads to the higher maturity of the customers. The remaining two paths, P_4 and P_5, which end with 'Unsubscribe', indicating these customers do not want to participate further events any more.

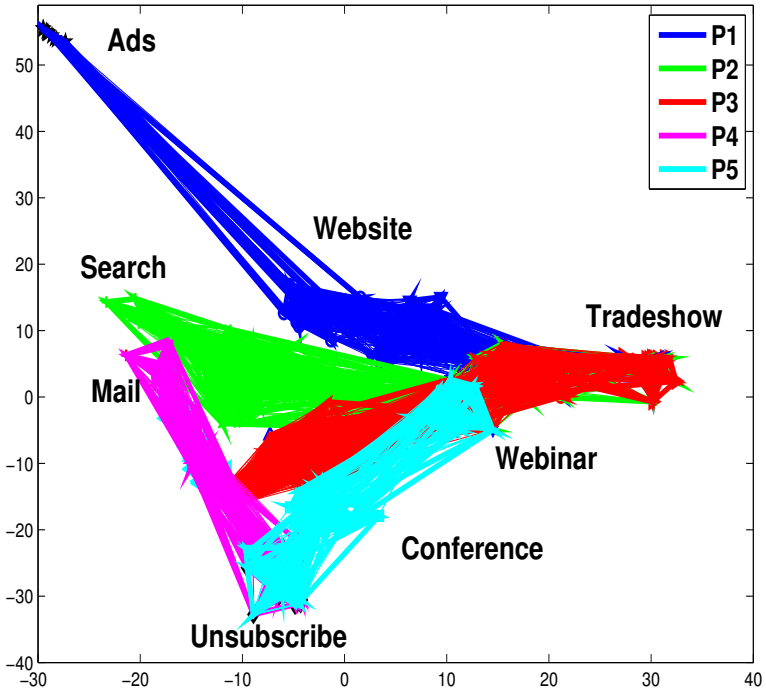


Figure 2.10. The customer buying paths.

These buying paths reveal different psychologies of B2B customers. Among the successful class, P_1 and P_2 represent customers that are comfortable with self-motivated (directed) actions, e.g., searching product information themselves or browsing advertisements. In comparison, in the unsuccessful class, P_4 represents customers passively involved via 'direct marketing mail' but finally choose to give up. For paths P_3 and P_5, although they both start from webinar, they branch to opposite routes. We speculate that P_3 are easy customers; while P_5 are customers that are relatively more difficult to persuade. These uncovered patterns can be helpful in guiding the marketing campaigns, such as identifying customer groups, initiating more customer-friendly and less commercialized advertisements, and so on.

Class	Path	Path/Keyword	Size
+	P_1	$C_{10} \rightarrow C_1 \rightarrow C_7 \rightarrow C_{12} \rightarrow C_8$ `Ads`→`Website`→`Download`→ `Webinar`→`Tradeshow`	933
	P_2	$C_{13} \rightarrow C_3 \rightarrow C_{11} \rightarrow C_{12} \rightarrow C_8$ `Search`→`Download`→`Webinar`→ `Webinar`→`Tradeshow`	1110
	P_3	$C_6 \rightarrow C_{11} \rightarrow C_{12} \rightarrow C_8$ `Webinar`→`Webinar`→`Webinar`→ `Tradeshow`	702
-	P_4	$C_2 \rightarrow C_9 \rightarrow C_5$ `Mail`→`Corporate Event`→ `Unsubscribe`	423
	P_5	$C_{11} \rightarrow C_4 \rightarrow C_5$ `Webinar`→`Conference`→ `Unsubscribe`	333

Table 2.3. The sequence clusters/buying paths.

2.6.4 Comparison with baselines

We also report the results of baseline algorithms such as FSM, HMMC, as well as the grouping based on the external knowledge (annotation) of the symbols.

Frequent pattern mining on raw sequences

We apply the FSM algorithms on the raw sequences and report the patterns identified in Table 2.4. With a very small support threshold (0.001 as shown in Figure 2.9), the maximum pattern length is only three, among which the top 10 with the largest support are listed. As can be seen, the resultant patterns mainly represent simple action sequences well expected by our common sense. Note that, there are thousands of patterns returned, which are difficult to be investigated one by one.

HMM based clustering

For the HMMC algorithm, we visualize the estimated transition matrix in one of the clusters in Figure 2.11, where the line width signifies the probabilities and edges less than 0.1 are omitted. As can be seen, there is no clear structure in the transition graph, which indicates that HMM based method cannot discover hidden patterns in such a complicated scenario. Moreover, as shown in Table 2.5, we cannot clearly annotate the hidden states by representative keywords of the campaigns with top emission probabilities to that state. For example, 'Seminar' appears in every hidden state with high probabilities. Indeed, this is consistent with the results we obtained in the simulated study (section 2.5), where the hidden states are typically quite hybrid with arbitrary subset of symbols which are hard to interpret.

Pattern/Keywords	Support
$s_{497} \rightarrow s_{616} \rightarrow s_{2134}$ `Webinar`→`Webinar`→`Webinar`	0.007
$s_{64} \rightarrow s_{2135} \rightarrow s_{2140}$ `Webinar`→`Webinar`→`Webinar`	0.007
$s_{3114} \rightarrow s_{4809} \rightarrow s_{141}$ `Webinar`→`Webinar`→`Webinar`	0.006
$s_{4809} \rightarrow s_{390} \rightarrow s_{186}$ `Webinar`→`Webinar`→`Download`	0.006
$s_{2135} \rightarrow s_{2139} \rightarrow s_{4521}$ `Webinar`→`Webinar`→`Download`	0.005
$s_{135} \rightarrow s_{329} \rightarrow s_{425}$ `Search`→`Website`→`Website`	0.005
$s_{2209} \rightarrow s_{390} \rightarrow s_{621}$ `Search`→`Webinar`→`Webinar`	0.005
$s_{2279} \rightarrow s_{2164} \rightarrow s_{3576}$ `Ads`→`Website`→`Website`	0.004
$s_{3167} \rightarrow s_{809} \rightarrow s_{3565}$ `Ads`→`Website`→`Website`	0.003
$s_{64} \rightarrow s_{141} \rightarrow s_{390}$ `Webinar`→`Webinar`→`Webinar`	0.003

Table 2.4. The raw sequential customer event patterns.

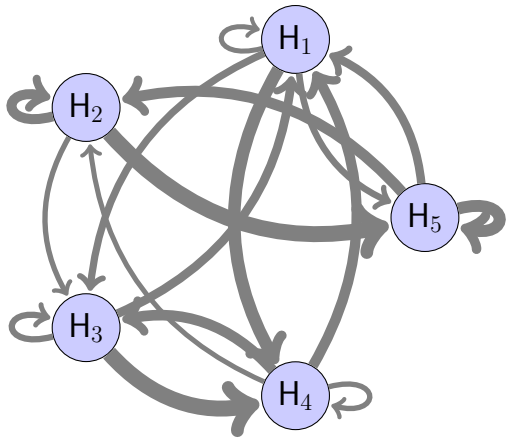


Figure 2.11. The HMMC transitions.

State	Top keywords
H_1	`Seminar, Official Website, Trial Download`
H_2	`Seminar, Corporate Event, Tradeshow`
H_3	`Trial Download, Seminar, Corporate Event`
H_4	`Seminar, Conference, Corporate Event`
H_5	`Seminar, Unsubscribe, Corporate Event`

Table 2.5. The semantic annotation of HMMC states.

External symbol grouping

Finally, we compare temporal skeletonization with the conventional practice of simply grouping the symbols based on their external attributes. In our data set, we have six categorical attributes describing the events attended by the customers. With these attributes, we do K-means clustering with Hamming distance, and use the cluster labels to re-encode the original sequence. Then, we measure the temporal variations of resultant sequences (Equation 2.1) and compare it with our approach.

As can be seen from Figure 2.12, the external grouping, which is irrespective of the temporal content of the sequences, leads to much larger temporal variations. This is undesired in subsequent pattern mining, while our approach leads to much less temporal variations even with relatively large number of clusters, meaning that it better captures the temporal structures of the sequences.

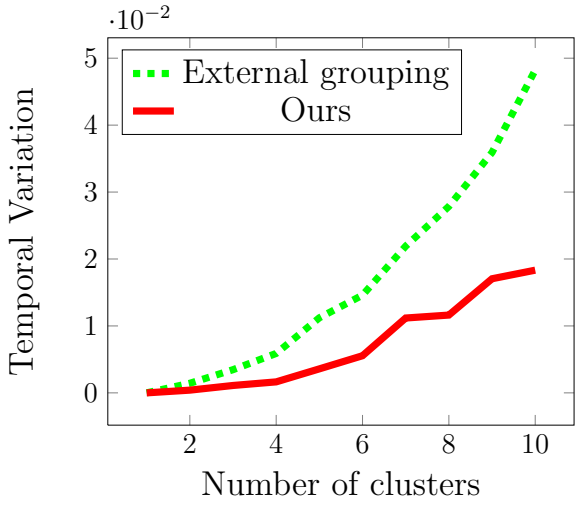


Figure 2.12. The comparison of temporal variation between external grouping and temporal skeletonization.

We also apply the FSM algorithms on the re-encoded sequences via external grouping. Table 2.6 reports top 10 patterns of length 3, with semantic annotations and support. Clearly, the patterns listed here are similar as the raw sequential patterns reported in Table 2.4, most of which represent well-known common sense. The only improvements are larger pattern support (e.g., $0.032 > 0.007$ for the first pattern), which is still much lower than that by temporal skeletonization (larger than 0.08 as shown in Figure 2.9).

Pattern/Keywords	Support
$E_6 \rightarrow E_2 \rightarrow E_1$ `Webinar`→`Search`→`Webinar`	0.032
$E_4 \rightarrow E_2 \rightarrow E_1$ `Mail`→`Search`→`Webinar`	0.026
$E_2 \rightarrow E_1 \rightarrow E_1$ `Search`→`Webinar`→`Webinar`	0.021
$E_2 \rightarrow E_7 \rightarrow E_1$ `Search`→`Conference`→`Webinar`	0.020
$E_2 \rightarrow E_2 \rightarrow E_1$ `Search`→`Search`→`Webinar`	0.019
$E_2 \rightarrow E_8 \rightarrow E_1$ `Search`→`Others`→`Webinar`	0.017
$E_2 \rightarrow E_1 \rightarrow E_9$ `Search`→`Webinar`→`Download`	0.016
$E_6 \rightarrow E_7 \rightarrow E_1$ `Webinar`→`Conference`→`Webinar`	0.015
$E_6 \rightarrow E_1 \rightarrow E_3$ `Webinar`→`Webinar`→`Webinar`	0.012
$E_5 \rightarrow E_2 \rightarrow E_1$ `Ads`→`Search`→`Webinar`	0.011

Table 2.6. The patterns with external grouping.

2.7 Related Work

Sequential pattern mining [Agrawal and Srikant, 1995] is an important topic in data mining. Given a database of customer transactions, where each transaction consists of customer-id, transaction time, and the items bought, sequential pattern mining finds frequent sequences of item sets. Some research efforts [Zaki, 2001, Han et al., 2001, Ayres et al., 2002] focused on the computing efficiency. In addition to the customer behavior analysis, sequential data from other application domains have also been exploited. For example, Giannotti et al. [2007c] proposed to find trajectory patterns from the location traces of moving objects to study their movement behaviors.

However, limited efforts [Srikant and Agrawal, 1995, Han and Fu, 1999, Giannotti et al., 2007c] have been focused on the "curse of cardinality" problem. As we discussed in section 2.1, these methods typically reduce the cardinality by performing a grouping of symbols, for which either a taxonomy already exists [Srikant and Agrawal, 1995], or extracted from domain knowledge [Han and Fu, 1999], or through clustering on the features associated with the symbols [Giannotti et al., 2007c]. These grouping are irrespective of the temporal content in the sequences, while our approach achieves the grouping of symbols based on their temporal relations. Note that combining the two types of clustering is a very interesting topic we shall pursue in the future.

Instead of reducing the cardinality of original symbols/items, one can also compress the discovered patterns for more concise interpretation. Pei et al. [2002] computed so-called condensed frequent pattern bases to approximate the support of any frequent pattern. Xin et al. [2005] proposed to compress frequent patterns with repre-

sentative patterns via clustering. In these methods, an initial set of frequent patterns has to be identified first, which could suffer from the large cardinality. In comparison, our approach can be deemed as compressing the original set of symbols.

Another category of related work is the rank aggregation [Schalekamp and van Zuylen, 2009], which tries to find a unified ranking of a set of elements that is "closest to" a given set of input (partial) rankings. For example, each customer event sequence can be deemed a ranking of the participated events. Methods for rank aggregation include position based statistics [Coppersmith et al., 2006] and permutation optimization [Dwork et al., 2001, Gionis et al., 2006], etc. However, rank aggregation is suited only when there is a dominant ordering in the data. When there are different patterns of the ordering, rank aggregation fails to give a valid result. In comparison, our approach can identify different types of orders as discussed in subsection 2.4.3.

The HMM is another widely used method for sequence analysis. Most algorithms for HMM estimation are supervised; that is, hidden states in the training data need to be provided for model estimation. However, in the B2B customer event sequence analysis, it is very difficult to obtain labels for the buying stages corresponding to individual events. For unsupervised estimation of HMM, Expectation Maximization (EM) is often used, which could suffer from the local optimum problem. In our simulated study (section 2.5), we observed that unsupervised HMM estimation often fails to recover the ground truth.

2.8 Summary

In this work, we proposed a novel approach of temporal skeletonization to address the problem of "curse of cardinality" in sequential data analysis. The key idea is to map the temporal structures of sequences into the topologies of a graph in a way that the temporal contents of the sequential data are preserved in the so-called temporal graph. Indeed, the embedding topology of the graph can allow to translate the rich temporal content into the metric space. Such a transformation enables not only sequential pattern mining at a more informative level of granularity, but also enables new possibilities to explore, quantify, and visualize statistically relevant temporal structures in the metric space. Finally, the experimental results showed the advantages of temporal skeletonization over existing methods. The case study also showed the effectiveness of the proposed method for finding interesting buying paths from real-world B2B marketing data, which otherwise would be hidden.

CHAPTER 3

PROACTIVE WORKFLOW MODELING FOR HEALTHCARE OPERATION AND MANAGEMENT

3.1 Introduction

Recent years have witnessed the increasing deployment of real-time location systems (RTLS) in hospitals. These RTLS solutions, as shown in Figure 3.1, allow people to track and monitor the movement of moving objects (e.g., medical devices, doctors and patients), and the interactions between people and medical devices. However, currently such systems are only used for basic tasks, such as simply locating a wheel chair and checking the availability of an inpatient bed. Meanwhile, hospital managers are still facing many managerial challenges. Particularly, there are three important tasks of operation and management in hospital: *workflow monitoring*, *workflow auditing*, and inspection of *workflow compliance*. For example, many healthcare providers have their own work protocols to ensure that the healthcare practices are executed in a controlled manner. Non-compliance practices may be costly and expose the healthcare providers to severe risks, such as litigation, prosecution and damage to brand reputation. However, it is not an easy task to systematically evaluate the degree to which the ongoing healthcare processes are compliant with the predefined protocols. Thus, effective inspection of workflow compliance is very needed by healthcare

providers to maintain the performance and reputation of their services.

Conventionally, to accomplish these tasks, hospital managers mainly rely on inspecting detailed workflow logs, which are managerially daunting. In particular, the logs can be heterogeneous in format and stored in different media. For example, many medical records are still documented in papers, which are not easy to be systematically processed and analyzed in real time. More importantly, these workflow logs are provided passively by the personnel, which might be biased. In contrast, a proactive approach requiring minimum human intervention would be a promising complement for the workflow management tasks. Indeed, the existing RTLS and collected location traces provide unparalleled opportunities for us to develop new ways to help with the workflow management tasks. To this end, this work provides a focused study of workflow modeling by the integrated analysis of indoor location traces in the hospital environments. One major goal is to proactively unravel the workflow patterns hidden in the location traces, by automatically constructing the workflow states and estimating parameters describing the transition patterns of medical devices. The discovered knowledge from the indoor location traces are then transformed to actionable intelligence for healthcare operation and management. Indeed, the learned workflow model is valuable in that a wide range of practical problems in hospital environments can be relieved with the modeling results.

Although modeling the workflow patterns in hospitals is of significant managerial value, it is not trivial to systematically construct and estimate the models with the massive indoor location traces. First, there are some inherent characteristics in the indoor scenario, which pose new challenges to the workflow analysis.

- The granularity and quality of indoor location traces captured by wireless location systems might vary a lot from place to place due to several factors: the density of sensor receivers, the underlying localization techniques, and the sensor device itself (e.g., the battery power percentage).

- The topologies of indoor space are often much more complex than outdoor space. Therefore, some fundamental assumptions of outdoor scenarios may not hold for indoor scenarios. For example, the widely-used similarity measurements for outdoor trajectories based on geometry distance [Ge et al., 2010] or overlapping degree [Giannotti et al., 2007a] are not much meaningful for indoor location traces.

- The structure of indoor space is very dynamic and the modeling framework should be automatic and adaptive accordingly. For instance, the structure and utility of many modern buildings may be frequently changed per the ongoing needs. Such dynamic changes will alter the semantic of indoor location traces.

As a result, even there are intensive works on the analysis of outdoor location traces [Giannotti et al., 2007a, Mamoulis et al., 2004, Ge et al., 2011a], most of them are not suitable or practical for modeling the indoor location traces of medical devices. Moreover, the results of those methods are not very helpful towards the workflow analysis in hospital environments. For example, the method developed by Giannotti et al. [2007a] aims for the discovery of frequent trajectory patterns based on given thresholds, i.e., minimum support and time tolerance, from outdoor location traces. However, such frequent patterns cannot provide a parsimonious description

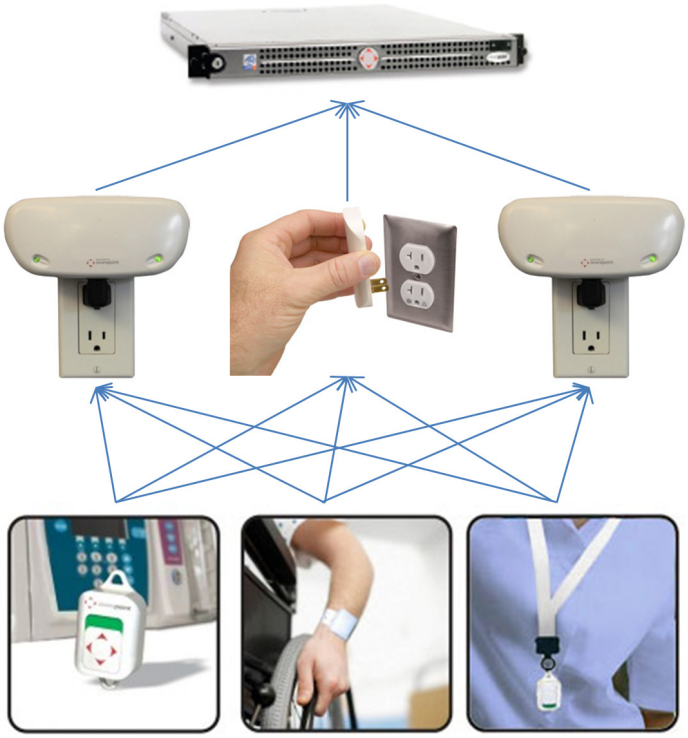


Figure 3.1. A demonstration of the real-time location system in hospitals. At the bottom layer, the moving objects (medical devices, patients, doctors, etc.) are attached with sensor tags, which send signals to the sensor receivers. The sensor receivers at the middle layer transmit the signal data to the network bridges. At the top layer, network bridges connected with data/application servers will collect the signal data and the data/application servers will calculate locations of the tracked objects.

for healthcare activities in hospitals. For example, to help with workflow compliance inspection, we need to consider all activities rather than only a subset of moving patterns, i.e., frequent patterns. Similarly, the periodic patterns mined from outdoor spatio-temporal data by Mamoulis et al. [2004] are also a subset of moving activities and cannot fully meet the needs of workflow monitoring, auditing, or compliance inspection in a hospital. While Yin et al. [2005] proposed stochastic models for indoor activities, the purpose of these models is to predict the goals of activities, rather than summarize overall activity patterns in descriptive ways. Besides, their methods are mostly supervised and need sufficient training data.

To accommodate the unique characteristics of indoor location traces of medical devices in a hospital, in this work, we propose a stochastic process-based framework to model the healthcare workflow. First, to extract each workflow state which is related to a particular type of healthcare activity, we develop a density-based clustering algorithm to partition the indoor space. Particularly, to incorporate the complex indoor topologies, we compute the neighborhood of a location record and its density in the indoor space based on the transition history rather than the geometry distance. With the clustering results, we further transform the original location traces into the sequences of workflow states. Then we analyze the workflow sequences by modeling the transitions among states with finite state machines.

Second, due to the dynamic indoor structure, we need an automatic and adaptive modeling framework to support the critical applications in real time. However, during some specific periods of observation, the estimation of parameters in the finite state machines may be unstable due to the data scarcity issue of the location traces.

Meanwhile, there is some natural correlation in the location traces of a group of medical devices, which are often used together for a particular task. By leveraging such correlation, we may overcome the data scarcity issue and improve the robustness of the parameter estimation for the finite state machines. Therefore, we explore MAP (Maximum A Posteriori) estimation in the Hidden Markov Random Fields (HMRFs) to estimate the transition probabilities, which can effectively leverage the correlation in the location traces of related medical devices. Moreover, in addition to the correlation relationship, we will be able to integrate more prior or domain information about the workflow of medical devices with the HMRFs, which will further improve the effectiveness of our workflow modeling.

Finally, we demonstrate the effectiveness of our models with the real-world data collected from multiple hospitals in US. We have also implemented and deployed a management information system, *HISflow*, based on our methods to show how the discovered knowledge can help with the three important managerial tasks in hospitals: workflow monitoring, auditing, and inspection of workflow compliance.

3.2 Preliminaries and Problem Formulation

In this section, we first introduce the data format of the indoor location traces. Then, we formally formulate the problem of workflow modeling for medical devices.

3.2.1 Data Description and Transformation

Our location traces (trajectories) of medical devices are collected from indoor environments of several US hospitals. The location trace of an object O is represented as a sequence: $O_t = (L_1, L_2, ...)$, where L_i represents the ith record in the sequence.

$L_i = (start_i, end_i; x_i, y_i, z_i)$ contains the specific coordinate and the corresponding time-stamp when that record is recorded, where $start_i$ and end_i are the start time and the end time of L_i. In other words, during the time frame from $start_i$ to end_i, the object O stays at the coordinate (x_i, y_i, z_i) in a three-dimensional indoor space.

However, the indoor wireless communication may be interrupted by environmental factors and lead to some errors or noises for the localization of moving objects. Therefore, a coordinate localized by the RTLS might not indicate the exact position, but a small area surrounding the coordinate. In addition, it may not be meaningful to directly use these coordinates for workflow modeling in the indoor space. For example, for two recorded coordinates which are close to each other on the same floor, although the geometry distance of them is very small, the actual moving distance from one coordinate to another one may be very long when there is a wall between these two coordinates.

To cope with these challenges, we normalize the original location traces for workflow modeling. Specifically, we project each raw coordinate to a semantic location of the building, such as a room in the hospital, based on the floor maps of the building. For the data and maps in this study, each hallway is also treated as a room and some long hallways have been segmented into several small rooms. Then, we map $L_i = (start_i, end_i; x_i, y_i, z_i)$ to $L_i^* = (start_i, end_i, r_i)$, where r_i is the room containing the coordinate (x_i, y_i, z_i). After this projection, two neighboring coordinates of the raw location traces may be mapped into the same room. In practice, we merge these consecutive records within the same room to one union record. Specifically, if $i < j$ and $r_i = r_{i+1} = \cdots = r_j$, we replace the subsequence $(L_i, L_{i+1}, \cdots, L_j)$ with only one

record $L_i^* = (start_i, end_j, r_i)$.

Now, each raw location record L_i is mapped to a symbolic room (graph node), and each location trace $Tr = (L_1^*, L_2^*, \cdots)$ is transformed to a symbolic sequence (traveling path). This data preprocessing drastically reduces the computational cost, since we significantly reduce the number of records in the data after the projection. Also, this preprocessing step greatly smooths out the noise and alleviates the impact of errors on the workflow modeling tasks.

3.2.2 Concepts for Workflow Modeling

The ultimate goal of our workflow pattern modeling is to automatically summarize the healthcare activities in a systematic manner. To this end, the concept of workflow patterns is actually a hierarchy with several levels. At the lowest level, the location trace itself can be seen as an instance of workflow patterns. For example, in Figure 3.2, we show location traces of an infusion pump with red line segments. However, it is difficult to comprehensively understand the pattern hidden in the raw location traces.

Locations	Functions
C_1	Post-anesthesia care unit (PACU)
C_2	Operating room (OR)
C_3	Intensive care unit (ICU)
C_4, C_5	Patient care unit (PCU)
E_1, E_2	Elevator

Table 3.1. The functions of key locations in Figure 3.2.

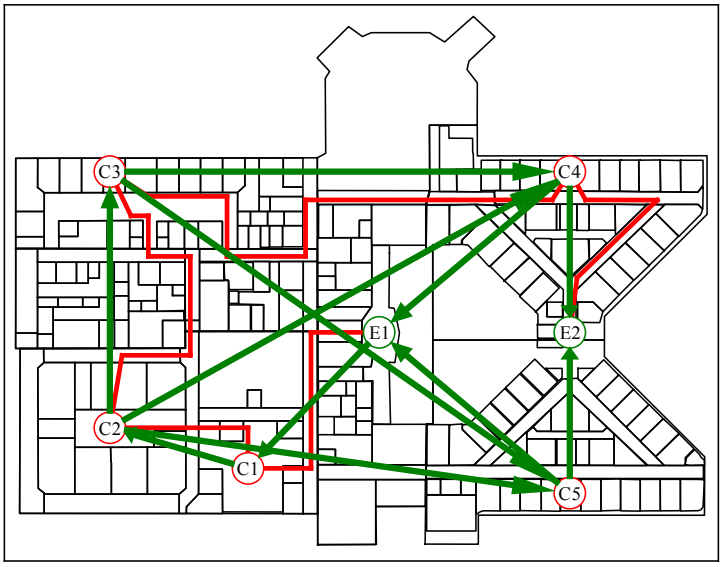


Figure 3.2. The workflow instances of infusion pump.

On the other hand, three workflow stages at the highest level are widely used in the healthcare industry to describe the workflow logistics: *preprocessing maintenance stage*, *in-use stage*, *postprocessing maintenance stage*. The *in-use stage* of a medical device corresponds to the period when it is used for any healthcare purpose. Before the *in-use stage*, a device is in the *preprocessing maintenance stage*, e.g., held in the storage room. After the *in-use stage*, the device must go through the *postprocessing maintenance stage* before the next circle of use. These maintenance processes include cleaning up, sterilization, disinfection, etc. However, modeling the workflow patterns based on these three stages is too coarse to get useful results to help with tasks of operation and management in a hospital. In particular, one important factor missed by the workflow stages is the spatial information, and the modeling results based

on these high level stages cannot facilitate the workflow evaluation from the spatial perspectives, such as the evaluation for the utilization efficiency of the indoor space. Therefore, we need to model the workflow patterns using a middle-level representation of location traces of medical devices.

Workflow States

To that end, we introduce an important concept, *workflow state*, which will serve as a basis for our workflow modeling at the proper level of granularity. Indeed, to better understand the location traces, we need to annotate a few key locations in these trajectories. For example, as shown in Figure 3.2, we annotate location traces with important location spots: C_i $(i = 1, \cdots, 5)$. The medical functions of these spots are summarized in Table 3.1. With these annotations, the location trace in red in Figure 3.2 can be represented as $C_1 \mapsto C_2 \mapsto C_3 \mapsto C_4$. Such representation with annotated spots makes it easy to understand the workflow behind the location trace.

In fact, Figure 3.2 is the map of the second floor of a hospital building, which is centered around E_1, the elevator connected to the basement. The red location trace of one infusion pump starts from the storage room in the basement to the elevator E_1. After E_1, the first spot C_1 is Post-Anesthesia Care Unit (PACU) where the patient is ready for medical procedures such as surgery and the medical devices are attached with the patients. Spot C_2 is the area of operating rooms (OR) where medical procedures happen. After the medical procedure, the patients and the medical devices are moved to C_3, Intensive Care Unit (ICU), based on the medical needs. When the situation of the patient becomes stable, the patient and the medical devices being

used will be further moved to C_4, Patient Care Unit (PCU), before the patient is discharged. After the patient is discharged, the medical devices will be moved through elevator E_2 to the basement for cleaning up in a disinfection room and depositing into the storage rooms.

From the above example, we know that each spot with a particular healthcare purpose corresponds to a workflow state, and the location traces represented with such workflow states are very meaningful for understanding and modeling the workflow patterns. Therefore, we formally define the workflow states in the context of hospitals as follows:

Definition 2 (Workflow state) *A workflow state is a location spot in the indoor space where specific healthcare activities frequently happen.*

According to this definition, a workflow state is a cluster of location records with specific activities. In the example, we have workflow states C_1, C_2, C_3, C_4. By representing the location traces with workflow states, the workflow patterns are clearer to be understood and inferred than that with low level rooms or high level stages.

3.2.3 The Problem Statement of Workflow Pattern Modeling

With workflow states, a further task is to describe the transition patterns of the moving objects among the workflow states. As shown in Figure 3.2, the transition from one state may go to several different states. For example, when the situation of a patient is stable after the medical operation at C_2, the medical devices and the patient might be moved directly to PCU without passing through ICU (C_3). Also, there are more than one PCU spots in the hospital. For example, in addition to C_4,

we have C_5 as another PCU spot in Figure 3.2. Note that, Figure 3.2 is only one floor in one building. In fact, we may have many workflow states since there might be several buildings and many floors in one hospital. Thus, the overall workflow patterns are indeed much more complex than the examples discussed above.

Furthermore, we have multiple types of medical devices moving around in the indoor space. In hospitals, many different medical devices are often used together for a particular task. Therefore, although different types of medical devices have different workflow patterns, there is some natural correlation among their location traces. Modeling such correlation not only helps reinforce the robustness of the workflow modeling, but also provides better understanding of the overall workflow patterns.

Now, we can formally state the problem of workflow pattern modeling in the indoor healthcare environments.

Problem 2 (Workflow pattern modeling) *Given the location traces ($\{Tr\}$) of moving objects, such as medical devices, workflow pattern modeling is to discover the workflow knowledge including workflow states (C), and parameters (μ) describing the transition patterns of the moving objects among the workflow states. In addition, we would also like to learn the correlation measurement (s) among the transition patterns for different types of moving objects.*

To accomplish these tasks, in section 3.3, we first propose a clustering algorithm to construct the workflow states. Next, in section 3.4, we model the transition patterns among the workflow states with finite state machines. In section 3.5, we further show how to compute the correlations and jointly build the multiple transition models.

3.3 Workflow State Construction

Inspired by Liu et al. [2012], we develop a DBSCAN-style clustering algorithm to construct the workflow states from the location traces. This density-based method is adopted to meet the requirements for the workflow analysis. Particularly, the clustering method should be able to automatically determine the number of clusters, and detect clusters of different densities and arbitrary shapes. The detected clusters should also be spatially contiguous, since we want to identify areas of semantically meaningful locations, such as '2nd floor northeast patient rooms' and 'basement central storage rooms'. However, even if we adopt the framework in [Liu et al., 2012], we have to develop some new methods to address the challenges caused by the unique characteristics of the workflow traces. First, we need a meaningful neighborhood definition for the location records in the indoor space. Second, to cope with the definition of workflow states, we need a density definition associated with workflow activities to detect clusters of different densities.

To address these challenges, let us first introduce how to define the neighborhood of a location record by exploiting the topologies of the indoor space. Since the location traces are represented symbolically, the indoor space map can be modeled with a corresponding symbolic graph. With all the symbolic nodes V as vertexes of the graph $G = (V, E)$, we can use edge E to represent neighboring relationship among the nodes. An edge is added between two vertexes if a direct transition path between the corresponding pair of symbolic location nodes exists. Formally, we define the neighboring relationship based on the transition connectivity:

Definition 3 (Transition neighborhood) *For room r_0, we denote*

$$TN(r_0) = \{r \mid \exists\, Tr, (rr_0) \in Tr\}$$

as the transition neighboring rooms of r_0. One room r is said to be transition neighbor of another room r_0 if one can transit from r to r_0 without passing through other rooms.

With 3, we define the neighborhood of a location record in the symbolic space:

Definition 4 *The neighborhood of a location record $L_0 = (start_0, end_0, r_0)$ is a set of location records observed in the rooms that are involved in transitions to r_0, denoted by*

$$N(L_0) = \{L = (start, end, r) | r \in TN(r_0)\}.$$

Thus, instead of identifying the neighborhood of a location record based on a distance threshold as used in DBSCAN, we define the neighborhood of a location record by directly querying the transition history from the data. In this way, we avoid the use of parameters, such as the distance threshold, which DBSCAN is very sensitive to.

To detect the clusters of different densities, we propose a *weighted density* definition to measure the density of individual neighborhood and identify core location records. If we know the distribution of waiting time T in room r_0, then a natural choice to weight the location record $L = (start, end, r_0)$ is the p-value $w(L) = \Pr(T \leq d)$, where $d = end - start$ is the waiting time of L. Intuitively, a longer waiting time d will lead to a bigger weight, as it takes time to carry out medical procedures and we are identifying the spot where specific medical activities frequently happen. If we do not know the distribution of T, which often happens in reality, we propose to

estimate the weight as

$$w(L) = \frac{|\{(start, end, r)|r = r_0, end - start \leq d\}|}{|\{(start, end, r|r = r_0\}|}.$$

With this formulation, we have the following definition of the weighted density for a neighborhood:

Definition 5 *The weighted density of a neighborhood $N(L_0)$ is defined as:*

$$w(N(L_0)) = \frac{\sum_{L=(start,end,r)\in N(L_0)} w(L)(end - start)}{|N(L_0)|}.$$

Proof[Example] By the definition above, suppose $N(L_0) = \{L = (start, end, r_0)\}$, let us calculate the weighted density for the following distributions of waiting times:

- $D_1 = \{1, 2, \cdots, 10\}$, 10 observations.
- $D_2 = \{5.5, 5.5, \cdots, 5.5\}$, 10 observations.
- $D_3 = \{1, 1, \cdots, 1\}$, 55 observations.

Although the total waiting time of D_1, D_2 and D_3 is the same as 55, our weighted densities are 3.85, 5.5, 1 respectively. For D_3, the neighborhood $N(L_0)$ is more likely in hallway or in front of elevator. Although the total waiting time in these places is long due to a lot of trajectories passing around, there is no long period of waiting time for carrying out medical procedures. Our weighted density definition can underrate this kind of situations. Similarly, observations with shorter waiting time in D_1 are also weighted lower.

Finally, we compare this estimated weighted density with a user-specified threshold θ to decide if a location record is a core point or not. Although there is similar

parameter in DBSCAN, i.e., minimum number of points required to form a cluster, the parameter θ is not the count of points but a weight measurement based on historical statistics. Due to the space limit, readers are referred to Liu et al. [2012] for details of the algorithm, except that the neighborhood and density definitions should be replaced by Definition 4 and 5.

3.4 Workflow Transition Estimation

With the constructed workflow states C, we can transform the original location trace into a workflow sequence $Tr = (s_1, s_2, \cdots, s_l)$, where $s_i = (c_i, d_i)$, $c_i \in C$ is the ith workflow state and d_i is the waiting duration in c_i. This transformation makes it feasible to model the workflow sequences of the objects moving among the indoor states with a continuous time Markov chain (CTMC). Specifically, with the constructed workflow states C as stochastic states in the workflow processes, a CTMC(μ) is defined with two sets of parameters $\mu = (P, q)$: waiting time parameters q_c for each state $c \in C$, and state transition probabilities $P_{cc'}$ from state c to c'. Let $S(t) \in C$ denote the state of the process at time $t \in [0, \infty)$. CTMC stays in $S(t) = c$ at time instant t for a random amount of duration $D_c \sim \text{Exp}(q_c)$. When CTMC finally leaves c, the next state of the system is c' with probability $P_{cc'}$ where $c' \neq c$. For $c = c'$, we let $P_{cc} = 0$. With these parameters, for the process Tr starting from s_1 and ending at time $d = \sum_{i=1}^{l} d_i$, the probability of Tr is

$$\Pr(Tr|P, q) = \exp(-q_{c_l} d_l) \prod_{i=1}^{l-1} q_{c_i} \exp(-q_{c_i} d_i) P_{c_i c_{i+1}}. \tag{3.1}$$

Without causing confusion, in the following, we denote states in C by integers $1, 2, \cdots, |C|$ for simplicity. For a set of processes $\{Tr\}$, the probability calculated

above can also be expressed in terms of N and D, where N_{ij} is the transition count from state i to state j, and $D_i = \int [S(t) = i]dt$ is the total waiting time in state i. Specifically,

$$\prod_{Tr} \Pr(Tr|P,q) = \Pr(N,D|P,q) = \left(\prod_{ij} P_{ij}^{N_{ij}}\right)\left(\prod_{i} q_i^{N_i}\exp(-q_i D_i)\right),$$

Here, $N_i = \sum_j N_{ij}$ is the transition counts from state i. Note that by definition we have $N_{ii} = 0$ and we compute $P_{ii}^{N_{ii}} = 1$.

One important fact in the likelihood of parameters P, q is that, we can estimate P_{ij} and q_i separately. That is, $\Pr(N,D|P,q) = \Pr(N|P)\Pr(N,D|q)$, where:

$$\Pr(N|P) = \prod_{i,j} P_{ij}^{N_{ij}} \tag{3.2}$$

$$\Pr(N,D|q) = \prod_{i} q_i^{N_i}\exp(-q_i D_i) \tag{3.3}$$

The solution to maximize $\Pr(N,D|q)$ is $q_i = \frac{N_i}{D_i}$. In other words, q_i is estimated by the reciprocal of the mean waiting time at state i. In the following, we will focus on optimizing $\Pr(N|P)$ to estimate the transition probability matrix P.

Under the normalization constraints $\sum_j P_{ij} = 1$ and $P_{ij} \geq 0$, it is straightforward to estimate the parameters P using a maximum likelihood estimation approach. However, the result estimated in this way will be unstable if the training data is not sufficient (e.g., when analyzing the daily data instead of monthly or yearly data). Fortunately, the data collected from different types of moving objects may help mutually reinforce each other to estimate the moving patterns in a robust way. To leverage this potential, we introduce the Maximum A Posteriori (MAP) estimation approach

which simultaneously estimates transition models for multiple types of moving objects. More importantly, with this approach, we will also be able to understand the collaborating relationship among different types of medical devices.

3.5 MAP Estimation in HMRFs

For the healthcare procedures of one patient, multiple types of medical devices are often needed. Thus, these different types of moving objects transit together and produce the correlated location traces. Therefore, although different medical devices follow different patterns, their workflow models can be built jointly with the consideration of these correlations. To this end, we first piece together the workflow states for all the K types of moving objects:

$$C = \cup_{k=1}^{K} C^{(k)}$$

where $C^{(k)}$ is the workflow states of the kth type of moving objects. Note that, when $C_1 \in C^{(1)}$ and $C_2 \in C^{(2)}$ overlap, i.e., $C_1 \cap C_2 \neq \emptyset$, we merge them together as $C_1 \cup C_2 \in C$. Now, all the K types of transition sequences can be expressed with the common transition states C, while each type has one transition count matrix $N^{(k)}$ and one transition probability matrix $P^{(k)}$. To estimate the K transition probability matrices $\{P^{(k)}|1 \leq k \leq K\}$ together based on $\{N^{(k)}|1 \leq k \leq K\}$, we use Hidden Markov Random Fields (HMRFs) to leverage the correlation among all types of objects. Specifically, an HMRF has the following components:

Hidden field: In our model, the hidden field includes the transition matrices $P^{(k)}$ for $1 \leq k \leq K$ for all types of objects, whose values are hidden or unobservable.

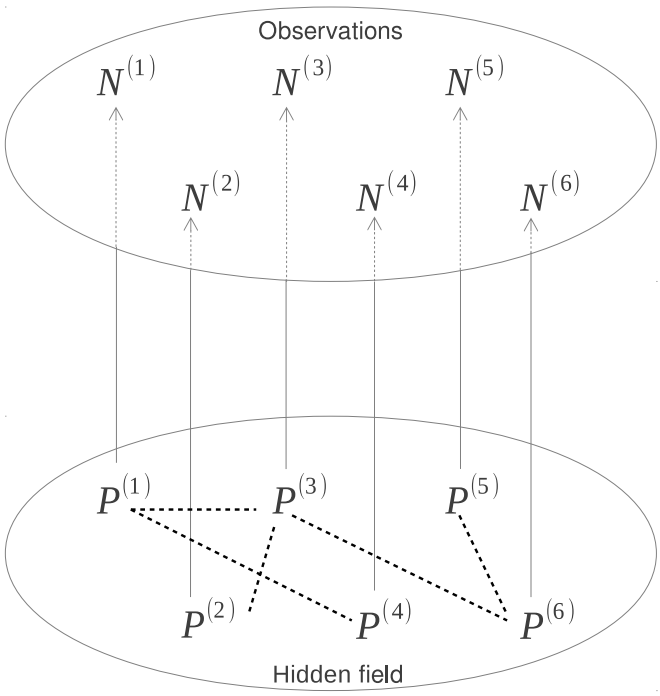


Figure 3.3. A Hidden Markov Random Field

Observations: The observed data includes the transition counts $N^{(k)}$ for $1 \leq k \leq K$.

As shown in Figure 3.3, there is also a connecting structure in the hidden field. We would like to reinforce the estimation of the connected parameters by leveraging the prior distribution of the random field. For our problem, we consider pairwise correlation among different types of objects, so the prior depends on $\sum_{k_1,k_2} V(k_1,k_2)$, where $V(k_1,k_2) \geq 0$ is the potential of the link between $P^{(k_1)}$ and $P^{(k_2)}$. Specifically, we have $\Pr(P) \propto \exp(-\sum_{k_1,k_2} V(k_1,k_2))$. By applying this prior, the objective function to be maximized is:

$$\Pr(P|N) \propto \exp(-\sum_{k_1,k_2} V(k_1,k_2)) \prod_{k,i,j} (P_{ij}^{(k)})^{N_{ij}^{(k)}}$$

or equivalently

$$\log \Pr(P|N) = -\sum_{k_1,k_2} V(k_1,k_2) + \sum_{k,i,j} N_{ij}^{(k)} \log P_{ij}^{(k)}. \tag{3.4}$$

Note that a normalization constant is omitted in the log-likelihood equation.

3.5.1 Adaptive Parameterized Correlation

HMRFs can integrate the knowledge about the neighborhood structure by formulating the potential $V(k_1, k_2)$. In the following, we demonstrate this framework by an unsupervised approach, where the neighborhood structure itself is also learned from the data. Specifically, we define the potential between them as

$$V(k_1,k_2) = w \times s(k_1,k_2) \times \mathrm{Tr}(P^{(k_1)}P^{(k_2)}), \tag{3.5}$$

where $w \geq 0$ is a scaling constant. Here, the correlation $s(k_1, k_2)$ measures the similarity between the transition patterns of types k_1 and k_2. The matrix trace $\mathrm{Tr}(P^{(k_1)}P^{(k_2)}) = \sum_{ij} P_{ij}^{(k_1)} P_{ji}^{(k_2)}$ is actually the probability that we can observe about the contradicting moves. For example, in the extreme case when $\mathrm{Tr}(P^{(k_1)}P^{(k_2)}) = 0$, we have $P_{ij}^{(k_1)} P_{ji}^{(k_2)} = 0$ for any given states i, j. In this case, we have $P_{ji}^{(k_2)} = 0$ only if $P_{ij}^{(k_1)} > 0$. In other words, for two types of objects that have no contradicting workflows, the probability of transition from state j to i is 0 for the second type of objects, only if we have a positive transition probability from state i to j for the first type of objects.

To measure the correlation adaptively with respect to the observed data, we propose to parameterize the Frobenius inner product using a non-negative matrix M,

which leads to the following formula:

$$s(k_1, k_2) = \frac{< N^{(k_1)}, N^{(k_2)} >_M}{\|N^{(k_1)}\|_M \|N^{(k_2)}\|_M},$$

where the weighted Frobenius inner product is $< X, Y >_M = \sum_{ij} X_{ij} M_{ij} Y_{ij}$ and weighted Frobenius matrix norm is $\|X\|_M = \sqrt{< X, X >_M}$. Indeed, N is directional observation and $s(k_1, k_2)$ corresponds to a von-Mises Fisher (vMF) distribution [Banerjee et al., 2003] as the underlying generative model.

3.5.2 Iterative Optimization Algorithm

We need an iterative updating algorithm to learn the correlation from the data and optimize the objective function in Equation 3.4 simultaneously. Specifically, there are two sets of parameters: Frobenius weights M and transition probabilities P. To update the Frobenius weights by increasing Equation 3.4, a closed-form solution is unattainable. In this case, gradient ascent update can be used alternatively. Particularly, the gradients are

$$\frac{\partial \log \Pr(P|N)}{\partial M_{ij}} = -w \sum_{k_1, k_2} \frac{\partial s(k_1, k_2)}{\partial M_{ij}} \operatorname{Tr}(P^{(k_1)} P^{(k_2)}) \tag{3.6}$$

and

$$\begin{aligned} \frac{\partial s(k_1, k_2)}{\partial M_{ij}} &= \frac{N_{ij}^{(k_1)} N_{ij}^{(k_2)}}{\|N^{(k_1)}\|_M \|N^{(k_2)}\|_M} \\ &- \frac{1}{2} \frac{< N^{(k_1)}, N^{(k_2)} >_M}{\|N^{(k_1)}\|_M \|N^{(k_2)}\|_M} \left(\left(\frac{N_{ij}^{(k_1)}}{\|N_{k_1}\|_M} \right)^2 + \left(\frac{N_{ij}^{(k_2)}}{\|N_{k_2}\|_M} \right)^2 \right). \end{aligned}$$

With the given M, the optimal P can be obtained by solving the following KKT system:

$$-2 \sum_{k'} s(k, k') P_{ji}^{(k')} + N_{ij}^{(k)} / P_{ij}^{(k)} - \lambda_{ki} = 0 \tag{3.7}$$

for $1 \leq k \leq K$ and $1 \leq i, j \leq |C|$. Here λ_{ki} is a KKT multiplier. With the normalization constraints $\sum_j P_{ij}^{(k)} = 1$, this quadratic system can be numerically solved. Note that, by iteratively conducting above updates, Equation 3.4 will be monotonically increased and we can stop the process when a sufficiently accurate solution is reached.

3.6 Empirical Evaluation

Here, we evaluate the performances of our workflow models in indoor healthcare environments. First, we show the identified workflow states and analyze the goodness-of-fit of our transition models with respect to important statistics. Then, we show that the learned correlation can reinforce the modeling performance, especially for real-time applications. The proposed models will also be compared with some baseline models. Finally, we showcase a few intelligent applications in healthcare operation and management that can benefit from the learned knowledge.

3.6.1 The Experimental Data

We use real-world data sets for validation. These data sets are collected from several hospitals in US. Medical devices operated in these hospitals have been attached with sensor tags and tracked by RTLS. In Table 3.2, we show basic statistics of the data collected for various types of medical devices in Hospital 1. Specifically, we will build workflow models for 7 types of medical devices. The 2nd and 3rd columns show the number of medical devices of each type operated in this hospital, and the number of location records collected during the period from January 2011 to August 2011.

Type	#Objects	#Locations	#States	θ seconds
Wheelchair	121	524415	121	1200
PCA II Pump	66	4431	2	1200
Venodyne	403	1588045	74	1200
Feeding Pump	83	157370	58	3600
Heating Pad	1	211	1	1200
PCA Pump	136	231057	74	1800
ETCO2	137	220380	79	1200

Table 3.2. Basic data statistics in Hospital 1

3.6.2 The Workflow States

In Hospital 1, there are totally 123 different workflow states identified. The last two columns in Table 3.2 give the number of workflow states identified for each device type and the corresponding threshold for density-based clustering. We can see that different types of medical devices are used in different ranges of region. For instance, wheelchairs move among 121 workflow states, while feeding pumps transit only in 58 workflow states. We also automatically identified that PCA II Pump and Heating Pad only move in a small range, which means the workflows of these two types of devices are quite simple. Thus, in the following we focus on analyzing the remaining 5 types of devices which have more complex workflow patterns. For the threshold of clustering, we empirically specify the parameter based on the mobility of medical devices and the classical work time for treatment in order to achieve well-separated workflow states.

The identified workflow states are verified to be semantically meaningful by the domain experts. For example, in Figure 3.4, we show the constructed workflow states for Venodyne, which moves most frequently, on 4 building floors. A summary of these workflow states is reported in Table 3.3. Particularly, with the developed clustering algorithm based on the the weighted density (demonstrated as transparency), we can effectively focus on the most important location spots. For example, as shown in Figure 3.4a, although most of the right part on this floor is planned to be PCU (Patient Care Unit), we find that only a small subset of rooms were frequently used. This is especially surprising in the top right corner, where only one room was used efficiently. These observations lead to an important application, workflow auditing, which we will further discuss in section 3.6.5.

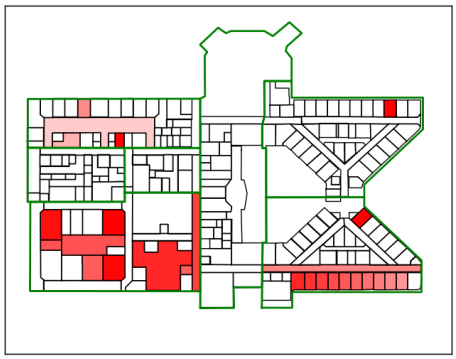
3.6.3 The Goodness-of-Fit

We experimented with different values of the scaling constant w in Equation 3.5 to build the transition models. If $w = 0$, our framework gives the maximum likelihood estimation. When $w > 0$, the scaling constant controls the trade-off between maximizing $\Pr(N|P)$ and deconstructing the potentials in the hidden field. Users can chose a value according to the correlation degree of the workflow patterns. In this work, we perform an exhaustive search in $[\frac{r}{2}, \frac{1+r}{2}]$ with step size 0.1 where r is the average non-weighted Frobenius inner products of the transition matrices.

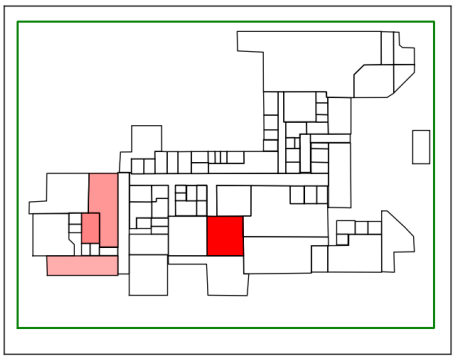
To measure the goodness-of-fit of the learned transition models, we compute the *average* log-*loss* with test location traces. For a test location trace $Tr = (L_1, L_2, \cdots, L_t)$ and its corresponding workflow sequence $Tr = (s_1, s_2, \cdots, s_\ell)$ transformed with work-

Building/Floor	State	Function
1/2nd	middle bottom	Post-anesthesia care unit
1/2nd	left bottom	Operating room
1/2nd	left top	Intensive care unit
1/2nd	right top	Patient care unit
1/2nd	right middle	Patient care unit
1/2nd	right bottom	Patient care unit
1/Basement	middle	Clinical Engineering
1/Basement	left middle	Storage
1/Basement	left bottom	Decontamination Service
2/1st	left top	Behavioral Health
2/1st	middle	Equipment Services
3/Basement	right top	Storage
3/Basement	middle	Nuclear Medicine
3/Basement	middle bottom	Hemodialysis

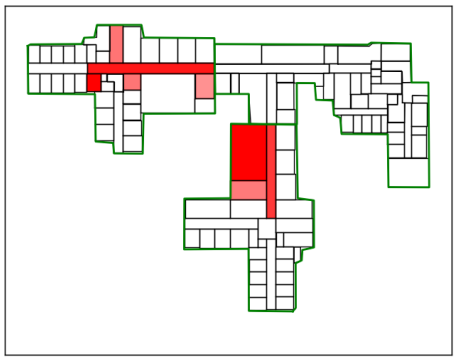
Table 3.3. Semantic summary of workflow states in Figure 3.4.



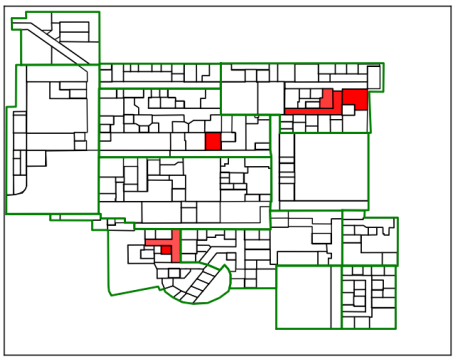
(a) Building 1/2nd Floor



(b) Building 1/Basement



(c) Building 2/1st Floor



(d) Building 3/Basement

Figure 3.4. Examples of workflow states.

flow states C, where $s_i = (c_i, d_i)$, the average log-loss is calculated as

$$L(Tr) = -\frac{1}{\ell} \log \Pr(Tr|P, q), \tag{3.8}$$

where $\Pr(Tr|P, q)$ is defined in Equation 3.1 and $\mu = (P, q)$ is learned from training data. According to Rissanen and Langdon [1979], smaller average log-loss implies better compression of the data when using the estimated parameters.

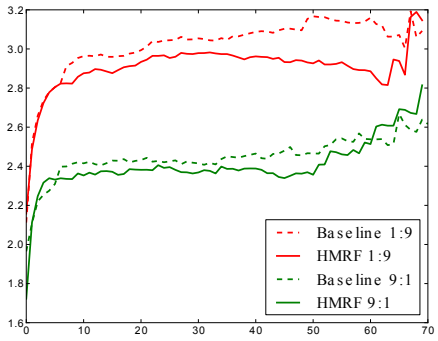
In the following, after constructing the workflow states C, we randomly partition the data into 10 subsets and compute the average log-loss in 10 rounds. In each

round, we use 9 of these subsets as training data to estimate the transition parameters $\mu = (P, q)$ and compute $L(Tr)$ for each Tr in the remaining test data. The solid green lines in Figure 3.5 show the average log-loss along the sequence length ℓ for different types of medical devices. We also show the performances with dashed green lines for the baseline models. The baseline model of each type straightforwardly estimates the transition parameters without considering the correlation among different types of moving objects. In the figure, we can see that our models based on HMRFs achieve lower information loss consistently for all types of moving objects.

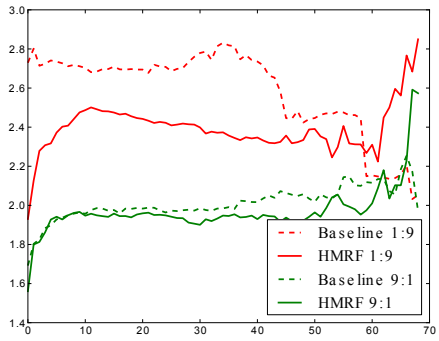
In practice, as we will discuss in subsection 3.6.5, sufficient observations might not be available to monitor the ongoing workflow patterns. In this case, our models can perform much better than baselines by holistic estimation. To see this, we repeat the above comparison by using only 1 subset as the training data. In Figure 3.5, as shown by the red lines, our models achieve much better performance. Our models not only achieve lower losses, but also produce robust results without rigid jumps.

3.6.4 The Simulation Performance

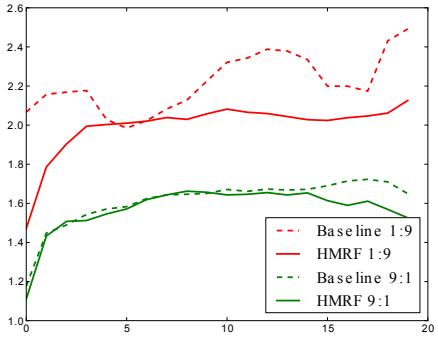
In addition to the goodness-of-fit in terms of information loss, a realistic model should also be able to simulate the modeled process without breaking important characteristics. In the research of time series data, one important characteristic is the auto-correlation function, ACF, which describes the serial dependence structure in the sequence models. In our workflow models, the state space C is categorical, thus we propose to investigate the χ^2 statistic to test the homogeneity. Specifically, for a set of categorical sequences $\{Tr = (c_1, \cdots, c_l)\}$ where $c_i \in C$ and $C = \{1, 2, \cdots, |C|\}$,



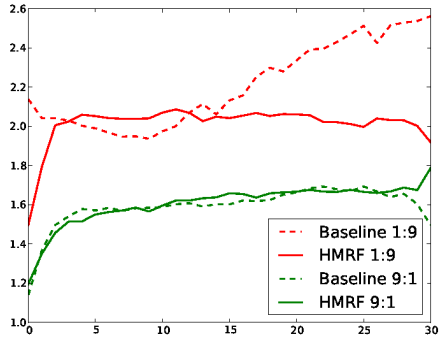
(a) Wheelchair



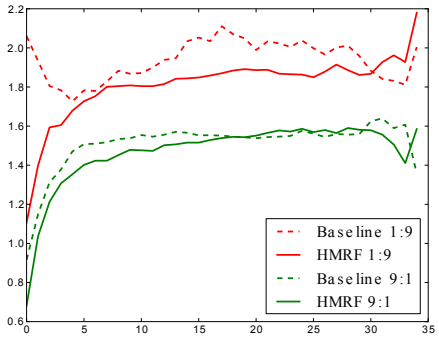
(b) Venodyne



(c) Feeding Pump

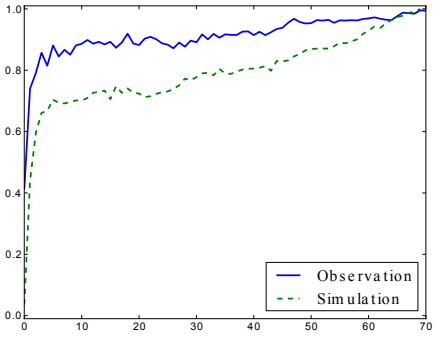


(d) PCA Pump

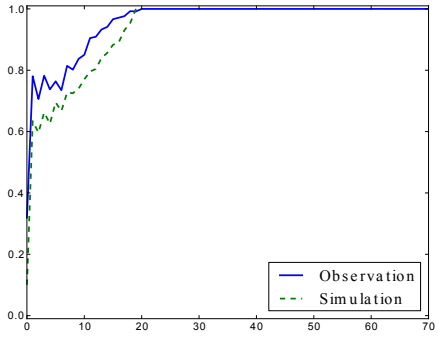


(e) ETCO2

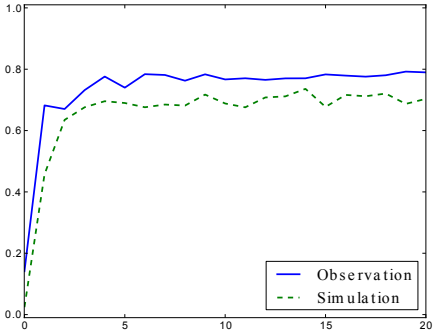
Figure 3.5. A comparison of the average log-loss. X-axis: the sequence length ℓ. Y-axis: the average log-loss.



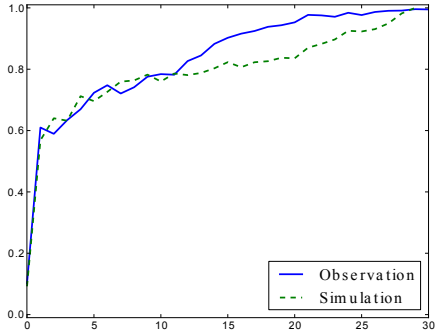
(a) Wheelchair



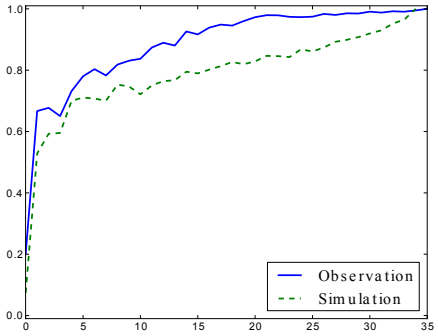
(b) Venodyne



(c) Feeding Pump



(d) PCA Pump



(e) ETCO2

Figure 3.6. A comparison of the AXF structure. X-axis: the time-lag h in AXF. Y-axis: the p-value of AXF.

the *auto* χ^2 *function*, AXF, is calculated as

$$\text{AXF}(h) = \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} \frac{(M_{ij} - E_{ij})^2}{E_{ij}}$$

where $M_{ij} = \sum_{Tr} \sum_{k=1}^{l-h} [c_k = i, c_{k+h} = j]$, $M_{i*} = \sum_{j=1}^{|C|} M_{ij}$, $M_{*j} = \sum_{i=1}^{|C|} M_{ij}$, $M = \sum_{Tr}(l-h)$ and $E_{ij} = \frac{M_{i*}M_{*j}}{M}$. The degree of freedom of the corresponding χ^2 distribution for AXF(h) is $(|C|-1)^2$. We can then calculate the p-value of the test of homogeneity with a null hypothesis that $(c_1, \cdots, c_{l-h})$ and $(c_h, \cdots, c_l)$ are independent. In Figure 3.6, we show the p-values of AXF(h) for the raw observations in solid lines. As can be seen, the serial dependence structures of the simulated processes shown in dashed lines are quite close to that of raw observations. Furthermore, we prefer the p-values that increase rapidly along h. This means the long-term dependence is little in the sequences and thus CTMC is a feasible model.

3.6.5 The Prototype Applications

The learned workflow model is valuable in that a range of practical problems can benefit from the results of modeling. In this work, we have implemented a management information system, HISflow, to exploit the discovered knowledge for healthcare operation and management. In the following we elaborate on the techniques used in our implementation.

Workflow Monitoring

When the mined workflow patterns are legislated as procedure codes, we can identify abnormal behavior from daily healthcare activities in a real-time manner. When such anomaly occurs, warnings and alerts can be activated by the management system.

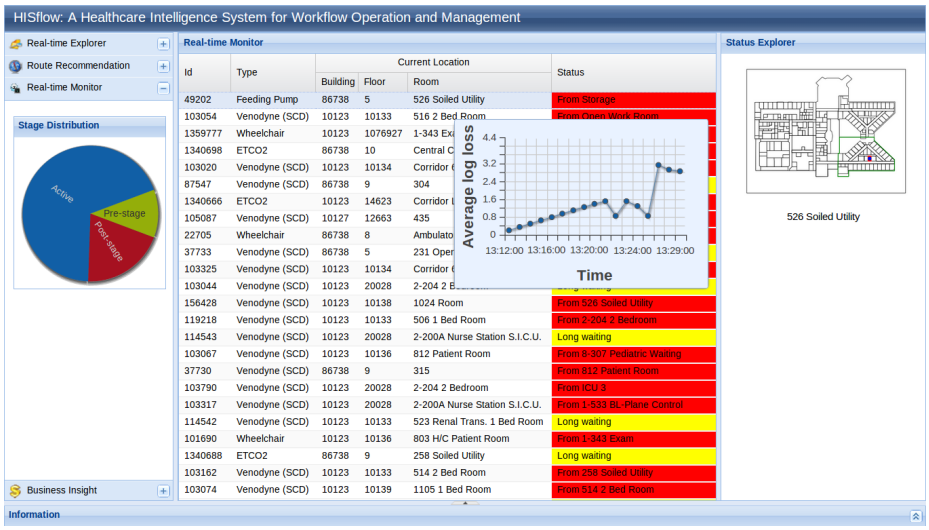


Figure 3.7. The screenshot of HISflow.

Such a system helps reduce the risks of faults or accidents of healthcare services. To develop this system, a simple approach is to rank the ongoing trajectories of all monitored medical devices based on average log-loss in Equation 3.8. In this way, the devices ranked in the top are worth more scrutiny. However, this ranking results might not be intuitive from the management perspective. In fact, it is vital to provide more insights into the cause for the higher log-loss trajectories. To this end, we further identify the recent transition $s_i = (c_i, d_i)$ in $Tr = (s_1, s_2, \cdots, s_l)$ which causes $L(Tr)$ to increase. With the identified s_i, we can then clarify whether the increase is caused by c_i or d_i via comparing $-\log P_{c_{i-1}c_i}$ and $-\log q_{c_i} + q_{c_i} d_i$. If $-\log P_{c_{i-1}c_i}$ is larger, the increase is caused by c_i because the transition coming from c_{i-1} is unlikely to end at c_i. Otherwise, it indicates that the device stays at c_i too long.

A screenshot of the real-time monitoring is shown in Figure 3.7.As can be seen, the average log-loss of the selected device ($Id = 49202$) increases linearly at the beginning, which means the device stayed at a workflow state c and $-\log q_c + q_c d$

increases linearly with waiting time d. As shown in red, when this device is moved from Storage Room (left middle spot in Figure 3.4b) to 526 Soiled Utility Room (blue spot in the floor map in Figure 3.7), it is ranked into the top due to the unlikely transition. We also show the related information in the grid and the location status on the floor map. Such an intuitive and interpretable real-time monitor system is valuable for the hospital managers to improve the quality of healthcare services.

Workflow Auditing

Our workflow models can also provide insights into auditing the efficiency of the healthcare services. Particularly, hospital managers would like to know how and how much the indoor spaces and the medical devices are being used. With our models, the utilization efficiency of medical devices and indoor space can be measured with sound statistics.

First, we can use waiting time parameter q in each state c to measure the utilization efficiency of the medical devices. Specifically, a small q indicates that the healthcare procedures taken in these states need a lot of time. For the workflow states corresponding to preprocessing maintenance stage (e.g., Storage rooms) and postprocessing stage (e.g., Soiled Utility Rooms, Decontamination Service Rooms), this small q means the processing procedures are not efficient. Second, to investigate the utilization efficiency of the indoor space, we would like to know the proportion of time the stochastic process spend at each state. In our CTMC model, this proportion at the ith state can be estimated by the stationary probability $r_i = \lim_{t\to\infty} Pr(S(t) = i) = \frac{\pi_i/q_i}{\sum_j \pi_j/q_j}$ where $\pi_j = \sum_i \pi_i P_{ij}$ and $\sum_j \pi_j = 1$.

Our system periodically calculates these statistics which allow hospital managers to audit the utilization efficiency of the medical devices and indoor space. By showing the trends over time and comparisons between different hospitals, one can get both intuitive understanding and quantitative measurement of the ongoing operation performance.

Workflow Compliance Inspection

The procedure of healthcare services is an operation process controlled by the science of medical treatment as well as industry legislation. Non-compliance practices lead to adverse affects such as litigation, prosecution and damage to brand reputation. To inspect the workflow compliance, we can calibrate the classical workflow sequences with respect to given workflow policies. In our stochastic model, the classical workflow sequences can be constructed by finding the most likely trajectories. Note that, in addition to providing statistic evidence on workflow compliance, the constructed classic workflow sequences are also helpful for understanding the ongoing workflow practices. Technically, given the initial state c, the final state c' and a time duration d, finding the most likely trajectory is an optimization problem to $\max_{Tr} \Pr(Tr|P, q)$, where $Tr = (s_1, \cdots, s_l)$, $s_i = (c_i, d_i)$, with constraints $c_1 = c, c_l = c'$ and $\sum_{i=1}^{l} d_i = d$. With parameters $\mu = (P, q)$ learned by our framework, this boundary value problem [Perkins, 2009] can be solved by dynamic programming algorithms.

3.7 Related Work

Workflow analysis conventionally relies on detailed workflow logs [Agrawal et al., 1998, Van der Aalst et al., 2004, Greco et al., 2005]. The workflow processes are

typically represented by activity graphs. Given the execution logs, which are lists of activity records, the workflow mining can be formalized as a graph mining problem by deeming the execution logs as walks on the activity graphs [Agrawal et al., 1998]. In practice, there might be discrepancies between the actual workflow processes and the processes perceived by the management. In this case, to discover a completely specified workflow design model, Van der Aalst et al. [2004] presented an algorithm to extract a process model from the workflow logs and represent it in terms of a Petri net. Instead of discovering the complete model, Greco et al. [2005] later formalized the problem of discovering the most frequent patterns of executions, i.e., the workflow substructures that have been scheduled more frequently and had lead to a desired final configuration. However, these methods rely on the workflow logs which are often recorded by people in the healthcare industry. Thus, the results may be distorted due to the missed activities and bias in the logs. These distorted results can be misleading for many operation and management tasks in hospitals, such as the inspection of workflow compliance. In comparison, as we discussed in section 3.1, in this work we propose a proactive approach to workflow modeling by mining the digital location traces of moving objects automatically recorded by RTLS in the hospitals, which requires minimum amount of human intervention. Our modeling results are helpful for a range of operation and management tasks in hospital environments.

In terms of methodology, another category of related work is the modeling and prediction of human activities. For instance, Yin et al. [2005] proposed stochastic process models to predict the goals of indoor human activities. Furthermore, for multiple-goal recognition, Chai and Yang [2005] proposed a two-level architecture for

behavior modeling and Hu and Yang [2008] developed a dynamic Bayesian model where skip-chain conditional random fields were used for modeling interleaving goals. However, these approaches are supervised and require sufficient training data. Also, the activities considered in the above papers are not within the hospital environments, where specialized workflow and activities happen. In the hospital environments, the desired knowledge to be discovered is different from and more complicated than that within other environments, such as school or company.

In terms of analytics of location traces, trajectory pattern mining is also related to this work. For instance, Giannotti et al. [2007a] introduced trajectory patterns as frequent behaviors in terms of both time and space, where the frequent trajectory patterns are computed based on the given thresholds. In [Li et al., 2010], methods were proposed to discover the periodic patterns from spatio-temporal data, where a periodic pattern is defined as a regular activity which periodically happens at certain locations. Also, Yang and Hu [2006] proposed methods to discover sequential patterns from imprecise trajectories of moving objects. However, these methods developed for outdoor space are not designed for the purpose of workflow modeling in the indoor hospital environments, and more importantly, the mined frequent patterns cannot provide a parsimonious description of healthcare activities in hospitals and support the applications we have considered.

Finally, the last category of related work is the detection of area-of-interest with trajectory data. For instance, Liu et al. [2010b] proposed a non-density-based approach, called mobility-based clustering, to identify the hotspots of moving vehicles in an urban area. The key idea is that sample objects are used as "sensors" to per-

ceive the vehicle crowdedness in nearby areas using their instant mobility, rather than the "object representatives". Moreover, Zheng et al. [2009a] proposed a stay point concept and identified hotspots from human moving trajectory. One location was considered as a hotspot if a lot of moving objects stay nearby over a thresholded time period. In addition, Giannotti et al. [2007a] used the neighborhood function to model Regions-of-Interest. Basically, they partitioned the spatial space into grids and quantified the interest of each grid with the density and the direction information of each grid. As we have discussed, even some methods mentioned above are very successful for analyzing outdoor location traces, most of them are not applicable to our hospital environments because of the unique characteristics of the indoor space.

3.8 Summary

In this work, we exploited the location traces of medical devices for modeling the healthcare workflow patterns in the hospital environment. Specifically, we developed a stochastic process-based framework, which provides parsimonious descriptions for long location traces. This framework provides new opportunities to understand the logistics of a large hospital in a concise manner. From the application perspective, the discovered knowledge, such as workflow states, transition patterns, and co-transiting relationships, can be integrated for use in the management information system we developed. With this system, we showed that valuable intelligent applications for healthcare operation and management can be enabled to manage, evaluate and optimize the healthcare services.

CHAPTER 4

A STOCHASTIC MODEL FOR CONTEXT-AWARE ANOMALY DETECTION IN INDOOR LOCATION TRACES

4.1 Introduction

Advances in mobile and sensor based technologies have allowed us to collect and process massive amounts of location traces across many different mobile applications. If properly analyzed, this data can be a source of rich intelligence for providing real-time decision making in various applications. It has been shown that the analysis of trajectory data can help to identify object movement patterns and understand moving object activities. Indeed, there are extensive work on the analysis of outdoor location traces, such as GPS traces Ge et al. [2011b,c]. However, the study of indoor location traces is relatively scattered due to the less availability of large-scale context-rich indoor location traces.

In this chapter, we study the indoor location data in the context of a hospital environment, where each medical device has been attached by a sensor. A real-time wireless location system has been deployed to locate and track each medical device. With this location system, it is possible to track the utilization of medical devices for better asset management. However, the indoor scenarios are quite different from the outdoor scenarios. The quality of indoor location traces captured by wireless

location systems depends on the density of device deployment as well as the underlying wireless localization techniques. Also, the quality of indoor location traces can also be affected by many other issues related to wireless devices, such as the battery issues and the defective sensor devices. Therefore, some fundamental assumptions for outdoor scenarios do not hold for indoor scenarios, and many techniques developed in the outdoor scenarios may not be suitable for the analysis of indoor location traces. For example, the widely used similarity measurements for outdoor trajectories are based on geometry distance or the overlapping degree of the trajectories, while these two measures are not much meaningful for indoor location traces.

To address the above challenges, we provide a focused study of identifying anomaly events of medical devices from their location traces. Indeed, one purpose of tracking all the medical devices in a hospital is to prevent the medical devices from being stolen. The location traces of all the medical devices provide the opportunity to detect the locations of missing devices. Along this line, we propose a stochastic model for context-aware anomaly detection in indoor location traces. The motivation for the context-aware framework comes from the observation that the missing events usually have a clustering effect and happen in some special spots in the hospital. These special spots are semantically meaningful and are usually close to the exits of building, elevators, or windows.

However, not all missing events are necessarily the stolen events of medical devices. For instance, when the wireless signal has been blocked in an area, the devices in this area may not be tracked for a long time. Although the system will indicate that there are missing events in this area, the devices will be tracked again in the system

if the communication channels have been cleared. Also, there are other issues which can lead to missing events, such as the battery problem and the sensor device change. Therefore, the missing events may happen in different contexts. In our context-aware framework, we first identify the hotspots which have high-level abnormal events in the indoor environment. The identified hotspots can then be used as the context for the nearby trajectories. With this context-aware model, we could identify several categories of abnormal events, such as missing events due to the blocking signal, the defective sensor devices, and the devices being stolen. In this stochastic model, we consider the hospital work flow and model the movements of medical devices as the transitions in finite state machines. In this way, we can estimate the stochastic properties of the hotspots and the transition patterns from the historical indoor location traces. The missing events can then be detected by measuring the abnormality of transition patterns comparing with the majority location traces.

Before we can accurately estimate the abnormality, however, we need to first address the uncertainty of the recorded location traces, which is caused by the relatively poor quality of the indoor sensor networks. To this end, after carefully studying data characteristics of recorded indoor location traces, we propose an iterative algorithm to address this uncertainty. Indeed, this algorithm is developed to effectively recover the missing location records which are critical for the abnormality estimation. Finally, we have conducted extensive experiments with real-world indoor location traces collected in a hospital environment. In experiments, we show the anomaly events captured by the context-aware anomaly detection model. With the benchmark from domain experts, we also show the effectiveness of our method in terms of detection accuracy.

4.2 Preliminaries and Problem Formulation

In this section, we first introduce the location traces of medical devices. Then, we formulate the problem of anomaly event detection. Next, we investigate the unique characteristics of these location traces in the hospital environment, where the moving activities of these devices are constrained by the routine work flow in the hospital. These unique data characteristics are important for the development of customized missing event detection techniques.

4.2.1 Data Preprocessing

We have collected a large amount of location traces (trajectories) of medical devices by real-time location systems in a number of hospitals. The location trace of each medical device can be denoted as a sequence:

$$Tr = (L_1, L_2, \cdots, L_l) \tag{4.1}$$

where $L_i (1 \leq i \leq l)$ represents the ith location in the sequence of length l. L_i contains the specific coordinate and the corresponding time stamp when this location is recorded, i.e., $L_i = (start_i, end_i; x_i, y_i, z_i)$ where $start_i$ and end_i are the start time and the end time of L_i. In other words, during the time window from $start_i$ to end_i, the corresponding device locates at the coordinate (x_i, y_i, z_i) in the 3-dimensional indoor space.

However, some wireless communication can be interrupted by environmental factors, such as metal objects. This communication interruption often leads to some errors or noise in the collected data. Indeed, a coordinate in the location traces lo-

calized by the sensor networks might not indicate the exact geometry position, but a small area surrounding the coordinate. In addition, it may not be very meaningful to directly use the coordinates for such indoor location traces of medical devices. For example, for two recorded coordinates which are close to each other on the same floor, although the geometry distance is very short, the real moving distance from one coordinate to another one may be very long when there is a wall between these two coordinates. To deal with these specific indoor challenges, we use the normalized location traces instead of the raw location traces. Specifically, we project each raw coordinate to a semantic location of the building, such as a room in the hospital, based on the given floor map. Each hallway is also treated as a room and some long hallways have been segmented into several small rooms. Then, L_i can be transformed to the following term:

$$L_i = (start_i, end_i, r_i) \tag{4.2}$$

where r_i is the room containing the coordinate (x_i, y_i, z_i). After such a projection, two neighboring coordinates within the raw location traces may be mapped into the same room. In other words, r_i for L_i may be the same as r_{i+1} for L_{i+1}. In practice, we merge these consecutive repeating records to one union record. Specifically, if $i < j$ and $r_i = r_{i+1} = \cdots = r_j$, we replace the subsequence $(L_i, L_{i+1}, \cdots, L_j)$ with only one record $L_i^* = (start_i, end_j, r_i)$.

Now, each raw location record L_i is transformed to a transition state, and Tr is transformed to a transition sequence among the rooms. This data preprocessing can drastically reduce the computational cost of anomaly detection because we signifi-

cantly reduce the number of records in the data. Also, this preprocessing step can greatly smooth out the noise and alleviates the impact of errors on anomaly detection.

To facilitate the following discussion, we use $r \in Tr$ to indicate that the location trace Tr passes the room r. A subsequence $(r_i, r_{i+1}, \cdots, r_j)$ of Tr will be denoted as $(r_i, r_{i+1}, \cdots, r_j) \subset Tr$. To estimate the transition probability in Equation 4.15, we let $\mathrm{NTr}(r_i r_j)$ count the transitions from r_i to r_j in the collected traces, and

$$\mathrm{NTr}(*r_j) = \sum_{r_i} \mathrm{NTr}(r_i r_j), \tag{4.3}$$

$$\mathrm{NTr}(r_i *) = \sum_{r_j} \mathrm{NTr}(r_i r_j). \tag{4.4}$$

In Section 4.5, we also use $\mathrm{NTr}(r_i r_j r_k)$ as the number of transitions passing through the 3 consecutive states r_i, r_j, r_k.

In addition to the location traces, we also have access to the map data. Although the underlying indoor environment is complicated, we focus on one important information; that is, we extract the neighboring relationship among the rooms. The first one is based on geometric distance.

Definition 6 *For room r_0, we denote*

$$GN(r_0) = \{r \mid \min_{L \in r, L_0 \in r_0} d(L, L_0) < d_0\}$$

as the geometric neighboring rooms of r_0, where $d(L, L_0)$ is the geometric distance between two locations L, L_0. Two rooms r_0, r are said to be geometric neighbors if the minimal distance between some point in one room and some point in the other room is less than a threshold d_0.

Note that $r \in \mathrm{GN}(r_0)$ if and only if $r_0 \in \mathrm{GN}(r)$. Strictly speaking, Definition 6 only applies to rooms on the same floor. For rooms on different floors, we treat them as not neighboring, although there might be common neighbors such as the stairs and elevators. Thus, we can also use different thresholds d_0 for different floors. For example, we can use a bigger threshold for a floor with bigger rooms without losing accuracy. For the hospital we study, we use $d_0 = 5$ meters regarding the building structure. The second type of neighboring relationship is based on connectivity.

Definition 7 *For room r_0, we denote*

$$TN(r_0) = \{r \mid \exists\ Tr, (rr_0) \in Tr\}$$

as the transition neighboring rooms of r_0. One room r is said to be transition neighbor of another room r_0 if one can transit from r to r_0 without passing through other rooms.

Note that each hallway is also treated as a room and some long hallways have been segmented into small rooms. Thus, two geometric neighboring rooms facing to the same hallway may not be transition neighbors, because one must pass through the hallway to transit from one to the other. On the contrary, for r to be in $\mathrm{TN}(r_0)$, it must hold that $r \in \mathrm{GN}(r_0)$ if the transition sequence is recorded correctly. The reason is that it is not likely to transit from one room to another room far away without passing through other rooms. We will study this issue further in Section 4.5.

4.2.2 The Abnormal Event Detection Problem

In the deployed sensor networks, a medical device with a remote sensor communicates with the networks at every network configuration time. However, some medical de-

vices may be inactive, such as disconnected from the networks, for a longer period, and some of them may never communicate with the networks again. Formally, suppose the network configuration time is D, which means a remote sensor communicates with the networks for every time period D, an inactive event of a medical device is recorded when this medical device does not communicate with the networks after a time period D. For the location traces of a medical device, $Tr = (L_1, L_2, \cdots, L_i, \cdots L_l)$ and $L_i = (start_i, end_i, r_i)$, if $d_i = start_{i+1} - end_i > D$, we can define

$$I = (d_i, r_i) \tag{4.5}$$

as an inactive event. Especially, if the device is stolen, then there is no the last location record, we let $d_i = start_{i+1} = \infty$. For a given trajectory Tr of a device, we denote its inactive event set by

$$E(Tr) = \{I = (d, r) | r = r_i \in Tr, d = start_{i+1} - end_i > D\}. \tag{4.6}$$

For those devices with such inactive events never communicating with the networks again, it probably means these devices have been stolen out of the hospital. In this chapter, *we aim to detect abnormal location traces, which might lead to missing (stolen) events.*

4.2.3 Data Characteristics

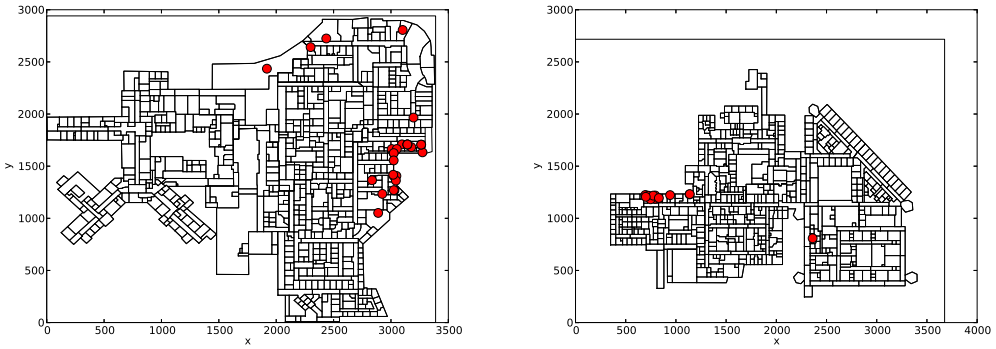
The Clustering Effect

Before proposing the method for anomaly detection, we first investigate the characteristics of these indoor location traces of medical devices. One important characteristic is the clustering effect for the spatial distribution of all the missing events. For exam-

ple, Figure 4.1 shows the locations of the missing events in a floor map. These missing events are identified by domain experts, which will be treated as the benchmark in this chapter. As can be seen in Figure 4.1 (a), there is a strong clustering effect for the last locations of missing medical devices in the first floor of the hospital. Indeed, there is a big cluster which is actually near exits and the elevators of the building. Another smaller cluster is near the top-right corner in the map. In this corner, there is no exit or elevator nearby, but this is on the first floor of the building and there are several windows in this corner. The situation in Figure 4.1 (b) is similar, where the cluster in the left corner is close to elevators and stairs.

Actually, this clustering effect is caused naturally by the building structure and network construction. First, the building structure may lead to the clustering effect of missing events. For example, the locations of missing or theft events are usually close to exists and windows directly connected to outside at the first floor of a building, because it is usually difficult to steal a device out of a building directly from a higher floor or a room without windows. The theft suspects may hide the devices near the elevator or the exit and then wait for a good time to steal the device out. When these stolen trials happen, the location systems will lose the traces of these devices for an abnormally long period. If the attempt fails eventually, we can also observe an abnormally long period of being inactive for the corresponding device.

Second, the network construction may also lead to this kind of clustering effect. For instance, when there is no network coverage at a specific location, the devices will not be able to communicate with the system at this location and abnormal events might be identified. Although this kind of missing events is not necessarily related



(a) 1st Floor (b) 2nd Floor

Figure 4.1. The spatial distribution of missing events identified by domain experts.

to theft events, detecting this kind of clusters can also help to improve the network quality and the medical asset management. In fact, a thief who is familiar with these locations with poor network coverage may intentionally pass these locations to steal a device without being tracked or captured.

Moving Patterns

The moving pattern of medical devices is highly influenced by the hospital work flow. For example, when an infusion pump is needed by a patient and there is no pump available nearby, the nurse will take one from the storage rooms. After the pump is used by the patient, it must be transported to one of the clean utility rooms. After it is cleaned or sterilized, the nurse will put it back to one of the storage rooms. There might also be the registration process when the medical devices are brought into and taken out of the storage rooms. Due to these work flows, the transition sequences of medical devices will show some frequently repeated patterns. Although we do not have the utilization information for each room, the abnormal transition sequence can be identified by our context-aware stochastic model.

The Uncertainty in Transition Sequences

For the high noisy indoor environment, a common issue is that some transition states may not be captured by the system. For instance, let us consider the underlying real transition sequence $r_i, r_{i+1}, r_{i+2}, \cdots$. Due to the noisy environment, very often, only the states $r_i, r_{i+2}, \cdots$ are recorded and the state of r_{i+1} is not captured by the system. The consequence of this issue is critical for mining the real transition pattern and the subsequent anomaly detection practice. Thus, we propose an iterative algorithm to deal with these difficulties. Specifically, we first investigate the nature of such issues using the building map. We show that it is possible to recover the lost states for recorded transition sequences by estimating the uncertainty for a finite set of candidates to be interpolated in between.

4.3 A Density-based Clustering Algorithm for Hotspot Detection

As we discussed in Section 4.2.3, it is critical to first identify hotspot as context for anomaly detection. In this section, we propose a density based clustering algorithm to group all the inactive events and detect hotspots with high levels of abnormality. Formally, each cluster is a hotspot of inactive events weighted by anomaly degree.

Inspired by Ester et al. [1996], we provide a DBSCAN-style algorithm to identify the hotspots. However, even we adopt the framework of the DBSCAN algorithm, we have to develop some new methods to address the challenges caused by the unique characteristics of the indoor location traces. First, while DBSCAN uses the geometry distance to define the neighborhood of an object, it is almost meaningless to define the neighborhood of one inactive event based on the geometry distance. Second,

DBSCAN estimates the density by simply counting the number of points within a neighborhood and uses the density to identify the core points. However, for the inactive event data, we have more information associated with each inactive event, such as the duration of being inactive. To address these challenges, let us first introduce how to define the neighborhood of an inactive event.

Definition 8 *The neighborhood of an inactive event $I_0 = (d_0, r_0)$ is a set of inactive events located in the rooms that are involved in transitions to r_0, denoted by*

$$N(I_0) = \{I = (d, r) \in E \mid r \in TN(r_0)\}.$$

Thus, instead of identifying the neighborhood of an inactive event based on a distance threshold as used in DBSCAN, we search the neighborhood of an inactive event by directly querying all the transition history from the data. In this way we can avoid the use of the parameters, such as the distance threshold, which DBSCAN is very sensitive to.

After identifying the neighborhood of each inactive event, we propose a *weighted density* to measure the density of individual neighborhood and identify core inactive events. Intuitively, more weight should be assigned to the event with a longer inactive duration, which is more likely abnormal with respect to the network configuration. To this end, we use a stochastic process model (Figure 4.2) to estimate the weight based on the inactive duration. Specifically, the state of a device can be either active or inactive at a moment. And we denote the state at time t as $S(t)$, which is 1 when it is active, and 0 otherwise. Moreover, if the memory less property is assumed in the appearance pattern, such a stochastic process becomes the Continuous-Time Markov

chain. For the Continuous-Time Markov chain, the amount of time T_i that the process stays in state 1 (0) before making a transition into state 0 (1) follows an exponential distribution as $Pr(T_i > d) = e^{-\lambda_i d}$ where $i = 0, 1$ and λ_i is the exponential rate in distribution of T_i. With this assumption, we can weight each inactive event by the transition probability as

$$w(I) = Pr(T_0 \leq d) = 1 - e^{-\lambda_0 d}. \tag{4.7}$$

Then, we have the following definition of the *weighted density*.

Definition 9 *The weighted density of a neighborhood of an inactive event I_0 is defined as:*

$$P(I_0) = \frac{\sum_{I \in N(I_0)} w(I)}{|N(I_0)|}. \tag{4.8}$$

The exponential rates λ_i for $i = 0, 1$ are the reciprocal of the expectation of the waiting time which can be estimated via the mean waiting time from the historical data. In particular, when estimating λ_0 for event $I_0 = (d_0, r_0)$, we use the following formula

$$\lambda_0 = \frac{|N(I_0)|}{\sum_{I=(d,r) \in N(I_0)} d}.$$

Finally, we compare this estimated *weighted density* with a user-specified threshold θ to decide if an inactive event is a core inactive event or not. Specifically, the inactive event I is a core one if $P(I) > \theta$. Similar to DBSCAN, we can also identify the border inactive events and noisy ones according to the definition of the *weighted density*. A detailed algorithm is described in Algorithm 1. Note that the noise events will be automatically detected and omitted, thus the union of clustering results is only a subset of all inactive events.

Algorithm 1 A Clustering Algorithm for Finding Hotspots

Construct the set of inactive events $E = \cup_{Tr} E(Tr)$.
Estimate λ_0 for each $I = (d, r) \in E$.
for Each event $I \in E$ **do**
  if $P(I) > \theta$ **then**
    Label I as core point;
  else
    if $\exists I_0, P(I_0) > \theta, I \in N(I_0)$ **then**
      Label I as border point;
    else
      Eliminate I from E as noise point.
    end if
  end if
end for
for Each pair of core points: I, I_0 **do**
  if $I \in N(I_0)$ and $I_0 \in N(I)$ **then**
    Put an edge between I, I_0.
  end if
end for
for Each group of connected core points C **do**
  Label C as a cluster.
end for
for Each border point I **do**
  Assign I to one of the clusters of its associated core points.
end for

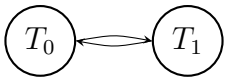


Figure 4.2. The states of the appearance process. State 0 is inactive and state 1 is active. T_i is the waiting time in state i.

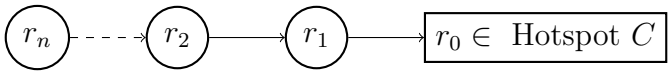


Figure 4.3. The transition process $r_{n:0}$.

4.4 A Stochastic Model for Anomaly Detection

Here, we introduce a probabilistic method for measuring the degree of anomaly based on the detected hotspots.

The behaviors of stealing the devices out of the building usually leave abnormal transition sequences (location traces) compared with the majority of transition sequences of most devices. This is especially true when suspects try to avoid the security guards when they steal the medical devices out of the building. Also, these abnormal location traces are usually related to those hotspots identified in the previous section. Thus, we define the abnormality measurement of the location traces with the identified hotspots as the context of the local transition sequences. Specifically, for each hotspot C, the set of local transitions ending within the cluster can be divided into two disjoint sets: O contains transitions ending with an inactive event and $\bar{O}$ contains transitions ending without any inactive event. The goal here is to determine if a specific transition sequence Tr ending within C belongs to O or $\bar{O}$.

Given a transition sequence Tr ending within the cluster C, we choose the tran-

sitions happened most recently to determine its subsequent event. Suppose that we have n transitions in the recent period t before the sequence ends within C. We denote the sequence as $Tr = (r_n, r_{n-1}, \cdots, r_1, r_0)$ where $r_0 \in C$ as shown in Figure 4.3. To simplify the following discussion, we use $r_{i:j}$ to denote the sequence $(r_i, \cdots, r_j)$. Now we define the abnormality of Tr as

$$o(Tr) = \frac{Pr(O|r_{n:0})}{Pr(\bar{O}|r_{n:0})} \tag{4.9}$$

which is the ratio between the probability that it ends with an inactive event and the probability that it ends normally. By applying the Bayes' Rule, it can be rewritten as

$$o(Tr) = \frac{Pr(r_{n:0}|O)Pr(O)}{Pr(r_{n:0}|\bar{O})Pr(\bar{O})}, \tag{4.10}$$

where the evidence term $Pr(r_{n:0})$ is reduced.

To compute this abnormality measurement, the first step is to estimate the ratio of priors $\frac{Pr(O)}{Pr(\bar{O})}$. By Algorithm 1, each hotspot is identified with the location traces which have inactive events near the hotspot. Thus, a simple estimation of the abnormality of C can be computed as $o(C) = \frac{|O|}{|\bar{O}|}$. However, the inactive events in O with different inactive duration have different weights defined by Equation 4.7. To take the weights into account, we estimate the ratio of priors as follows:

$$\frac{Pr(O)}{Pr(\bar{O})} = \frac{\sum_{I \in O} w(I)}{|\bar{O}|}. \tag{4.11}$$

Furthermore, by using the Chain Rule, the likelihood can be expanded. For the

set O, we have

$$Pr(r_{n:0}|O) = Pr(r_n|r_{n-1:0}, O)Pr(r_{n-1}|r_{n-2:0}, O)$$
$$\cdots Pr(r_1|r_0, O)Pr(r_0|O)$$

If expanded in this way, the last term $Pr(r_0|O)$ is able to be estimated from the historical data.

However, the estimation of these conditional transition probabilities are computationally too expensive. Instead, we propose to use the simpler N-gram model; that is, the transition r_i is independent of transitions r_{i-j} if $j \geq N$. We use the Bigram model ($N = 2$) as follows

$$Pr(r_{n:0}|O) = Pr(r_n|r_{n-1}, O)Pr(r_{n-1}|r_{n-2}, O)$$
$$\cdots Pr(r_1|r_0, O)Pr(r_0|O). \tag{4.12}$$

Similarly, for the set of active events, we have

$$Pr(r_{n:0}|\bar{O}) = Pr(r_n|r_{n-1}, \bar{O})Pr(r_{n-1}|r_{n-2}, \bar{O})$$
$$\cdots Pr(r_1|r_0, \bar{O})Pr(r_0|\bar{O}) \tag{4.13}$$

Finally, we have the anomaly degree measurement as the following:

$$o(r_{1:n}) = \prod_{i=1}^{n} \frac{Pr(r_i|r_{i-1}, O)}{Pr(r_i|r_{i-1}, \bar{O})} \times \frac{Pr(r_0|O)}{Pr(r_0|\bar{O})} \times \frac{\sum_{I \in O} w(I)}{|\bar{O}|}. \tag{4.14}$$

We will show the effectiveness of this Bigram model later in Section 4.6.

4.5 Transition Probability Estimation in Noisy Environment

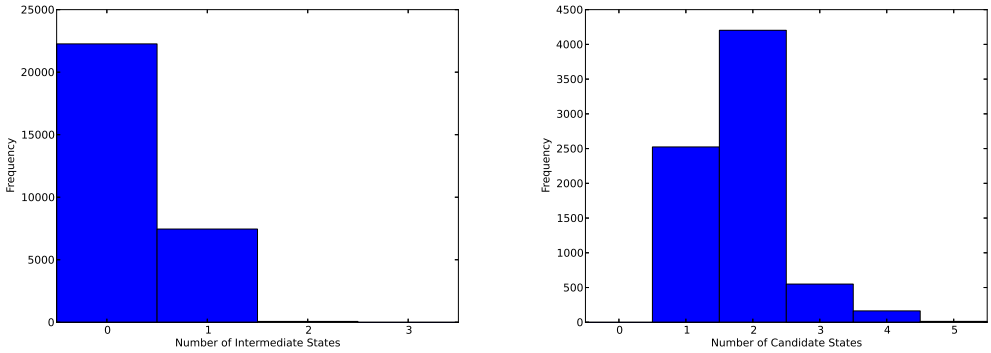
Now we elaborate on the estimation of $Pr(r_i|r_j)$, which is the probability of a transition starting from state r_i given that it ends in state r_j. Empirically, $Pr(r_i|r_j)$ can

be estimated as

$$Pr(r_i|r_j) = \frac{\mathrm{NTr}(r_i r_j)}{\mathrm{NTr}(*r_j)}, \tag{4.15}$$

where $\mathrm{NTr}(r_i r_j), \mathrm{NTr}(*r_j)$ are defined in Section 4.2. Nevertheless, as we discussed in Section 4.2.3, the issue of uncertainty in transition sequences is critical for the estimation in Equation 4.15 and the model in Equation 4.14.

To address this problem, we empirically interpolate possible missing states between each recorded transition, and simultaneously estimate transition probability in an iterative way. Specifically, for the recorded transition from r_i to r_j where $r_i \notin \mathrm{TN}(r_j)$, we need to recover the possible lost transition states between r_i and r_j. Ideally, we should consider all possible transition sequences $(r_i r_1^* r_2^* \cdots r_k^* r_j)$ from r_i to r_j, where $r_i \in \mathrm{TN}(r_1^*)$, and $r_k^* \in \mathrm{TN}(r_j)$. Also $r_t^* \in \mathrm{TN}(r_{t+1}^*)$ for $t = 1, 2, \cdots, k-1$, where k is the length of missing sequence. However, this leads to an intractable problem due to extreme complexity. Fortunately, in many indoor wireless network systems, it is not common that a long sequence of states are lost in most transition sequences. Thus, we can significantly reduce the complexity by limiting the length of the lost sequence k less than a threshold. To determine a practical threshold, Figure 4.4a shows the histogram of number of intermediated states of recorded trajectories near one hotspot. One can see that most of the transitions (about 74.7 percent) are from and to states which are transition neighbors. For these transitions, there was no lost record between the neighboring transition states. We have 7454 (about 25 percent) recorded transitions, for which we need 1 intermediate state because they are from and to states which are not geometric neighboring (thus also not transition



(a) Number of Intermediate States (b) Number of Candidate States

Figure 4.4. Two Histograms of Transitions.

neighboring). Our task is to recover the lost intermediate state for each transition $(r_i r_j)$. We have few (70, about 0.2 percent) transitions, where it is impossible to recover the sequence with only 1 intermediate state. Since transitions with multiple consecutive missing states are really rare, it is sufficient to use the threshold of 1. In other words, we assume only one state may be missed between two consecutive states of a sequence in this chapter.

For a recorded transition $(r_i r_j)$, where r_i and r_j are not geometric neighboring, we need to find candidate intermediate states r^* such that $r^* \in \text{TN}(r_j)$ and $r_i \in \text{TN}(r^*)$. If there exists only one such candidate intermediate state, r^*, we are quite confident that the real transition sequence is $(r_i r^* r_j)$ which can be confirmed by domain experts with experiments. But there may be more than one candidate intermediate states between r_i and r_j. Figure 4.4b shows the distribution of number of candidate intermediate states for the 7454 transitions. For the first category, where we have only 1 candidate intermediate state for each transition, we interpolate that state r^* between r_i and r_j. For the remaining categories, we have more than 1 (and no more

than 5) candidate intermediate states, where there is uncertainty on which of them should be interpolated.

Let G denote the set of recoverable transitions. We recover the lost state r^* for the transitions $(r_i r_j) \in G$ with uncertainty defined as

$$\mathrm{P}^{r^*}_{r_i r_j} = \frac{\mathrm{NTr}(r_i r^* r_j)}{\sum_{r \in \mathrm{TN}(r_j), r_i \in \mathrm{TN}(r)} \mathrm{NTr}(r_i r r_j)}. \tag{4.16}$$

Via Equation 4.16 we obtain a probability for each candidate intermediate state for a transition. In other words, we consider each candidate intermediate state as the missing point with a probability.

After the interpolation in such a probabilistic way, each trajectory Tr with transition $(r_i r_j) \in Tr$ now can be reconstructed by replacing $(r_i r_j)$ with $(r_i r^* r_j)$ and a corresponding weight $\mathrm{P}^{r^*}_{r_i r_j}$. Here the uncertainty weights will be used during counting the number of transitions in the weighted trajectories. Each transition in one weighted trajectory will be counted as the weight instead of 1 as we did in the beginning.

However, after this reconstruction, the estimation in 4.16 will be changed due to overlap between lost states. For example, suppose we have trajectories with transition sequence $(r_1 r_3 r_4)$ and $r_2 \in \mathrm{TN}(r_3)$ and $r_1 \in \mathrm{TN}(r_2)$. After the reconstruction process, some of these trajectories will be recovered as $(r_1 r_2 r_3 r_4)$. Thus, one consequence is that the count of $\mathrm{NTr}(r_2 r_3 r_4)$ and the uncertainty weight $\mathrm{P}^{r_3}_{r_2 r_4}$ will be increased. Therefore, we need to iterate the two steps, computing the uncertainty weights and reconstructing trajectories, till convergence. Specifically, an iterative algorithm is shown in Algorithm 2. Generally, this algorithm converges quickly for the following reasons. First, consider the example above again, if the increase of $\mathrm{P}^{r_3}_{r_2 r_4}$ will not

Figure 4.5. A type of infusion pump.

affect $P^{r_2}_{r_1 r_3}$, then these two uncertainty weights will not be changed again because of these two segments. On the contrary, if the increase of $P^{r_3}_{r_2 r_4}$ does affect $P^{r_2}_{r_1 r_3}$, $P^{r_2}_{r_1 r_3}$ must be increased too. Thus, they will be changed monotonously. We omit more detailed analysis due to space limit.

4.6 Experimental Results

In this section, we evaluate the performances of the proposed context-aware anomaly detection method in indoor location traces. Specifically, we demonstrate: 1) the effectiveness of the proposed model for identifying interesting anomaly events. 2) a performance comparison between the proposed method and the baseline methods.

4.6.1 The Experimental Setup

Experimental Data. We use a real-world data set for the evaluation of the proposed methods. This data set is collected from a hospital. Each medical device operated in the hospital has an attached sensor and tracked by the location system. The location traces of each medical device is recorded in an accumulative way. Specifically, the network tracks every device periodically. When the location of a tracked device changes, a location change record will be generated and recorded in the database.

Algorithm 2 Algorithm to estimate the uncertainty for intermediate state

Input: Historical trajectories $\{Tr\}$.

Output: Recovered trajectories $\{Tr^*\}$.

Extract the recoverable transitions into set G.
repeat
  for Each transition $(r_i r_j) \in G$ **do**
    for Each r^* where $r^* \in \mathrm{TN}(r_j)$ and $r_i \cap \mathrm{TN}(r^*)$ **do**
      Compute the uncertainty weight $\mathrm{P}^{r^*}_{r_i r_j}$ by Equation 4.16.
    end for
  end for
  for Each Tr **do**
    for Each transition $(r_i r_j) \in G$ **do**
      for Each $r^* \in N(r_i) \cap N(r_j)$ **do**
        Recover $(r_i r_j)$ as $(r_i r^* r_j)$ with weight $\mathrm{P}^{r^*}_{r_i r_j}$ for Tr.
      end for
    end for
  end for
until Converge

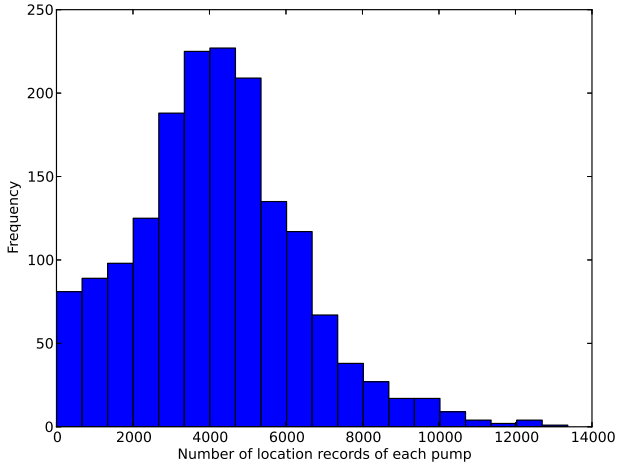


Figure 4.6. The histogram of the number of location records of each infusion pump.

The medical device we studied in this chapter is infusion pump, which is the most common device in the hospital. The size of one pump is about $9 \times 5 \times 5$ inches and the weight is about 15 pounds. An infusion pump is shown in Figure 4.5. Given the size and weight, it is easy to take it out of the hospital without the notice of security guards. Since each infusion pump is worth several thousand dollars, it is very important to detect the missing events of infusion pumps.

In detail, we have collected the location traces of 1680 infusion pumps in the hospital. The duration is from January 2011 to August 2011. For these infusion pumps, the total number of location records is 7038246. The distribution of the number of location records of each pump is shown in Figure 4.6. The sensor network deployed in this hospital is configured to communicate with the remote sensor of a device with a time gap no longer than $D = 30$ minutes: If a device is in motion, the network will communicate with the senor and a new location of this device will be recorded every minute; If the device is not in motion, the network will communicate

with the sensor every 30 minutes. Based on the definition of inactive events in Section 4.2.2, we obtained 26752 inactive events, which were used to discover the hotspots.

Algorithm 3 A Context-Aware Anomaly Detection Algorithm

1: Use Algorithm 1 to identify hotspots $\{C\}$ from $\{Tr\}$.
2: For each hotspot C, divide the location transitions ending within C into two disjoint sets: O containing transitions ending with an inactive event and $\bar{O}$ containing transitions ending without inactive event.
3: Use Algorithm 2 to recover the lost transition states for transitions in O and $\bar{O}$, separately.
4: Use transition sequences in O to estimate $Pr(r_i|r_j, O)$ with Equation 4.15.
5: Use transition sequences in $\bar{O}$ to estimate $Pr(r_i|r_j, \bar{O})$ with Equation 4.15.
6: Use Equation 4.14 to estimate the anomaly degree of $r_{n:0}$.

The proposed overall detection method. Our overall method is outlined in Algorithm 3. Specifically, there are three components. First, we identify the hotspots as the context of transition sequences. Then, before measuring the anomaly degree, we recover the lost transition states for trajectories of each context. Finally, we use the N-gram model defined in Equation 4.14 to distinguish the normal trajectories from the abnormal trajectories.

To demonstrate the effectiveness of our approach, we also proposed the following baseline methods:

The baseline method 1 **via Hotspots.** As a baseline method, we can estimate the anomaly degree of a transition sequence Tr ending within the core point c based

on its density in the hotspot C. Specifically, as discussed in Section 4.3, we define

$$o(Tr) = P(c) = \frac{\sum_{I \in N(c)} w(I)}{|N(c)|}. \tag{4.17}$$

The baseline method 2 without addressing the uncertainty issues. Another baseline method we compared is similar with Algorithm 3, except that the step 3 is omitted. In other words, we measure the anomaly degree of trajectories without addressing the uncertainty issues in the recorded sequences.

Experimental Tools and Parameters. The parameter θ, i.e., the minimum mean weight of the inactive events to construct a hotspot, for our clustering algorithm is empirically set as $\theta = 0.85$. Another important parameter is the time lag t. We use $t = 1$ day, because it is natural to assume that the abnormal events only happen in a short period of the same day in most cases.

Accuracy Measurement. We use three popular accuracy measurements to evaluate the experimental results. With the detected abnormal trajectory set A by our methods and the manually-labeled abnormal trajectory set B by domain experts, the precision, recall and F-measure can be calculated by:

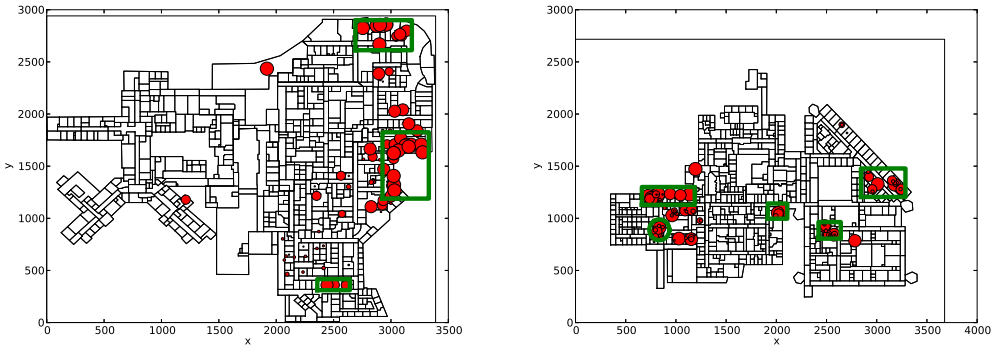
$$Precision = \frac{|A \cap B|}{|A|}, \tag{4.18}$$

$$Recall = \frac{|A \cap B|}{|B|}, \tag{4.19}$$

$$F - measure = 2\frac{Precision \times Recall}{Precision + Recall}. \tag{4.20}$$

In this experiment, we compute these measurements for the top-k abnormal trajectories with different k values.

Human labeled benchmark. We have obtained 67 abnormal missing events



(a) 1st Floor (b) 2nd Floor

Figure 4.7. The spatial distribution of weighted inactive events and hotspots. Each red point is an inactive event with the radius proportional to its weight. Each green box is a hotspot.

which are manually labeled by the domain experts as the benchmark. The spatial distribution of these events are illustrated in Figure 4.1.

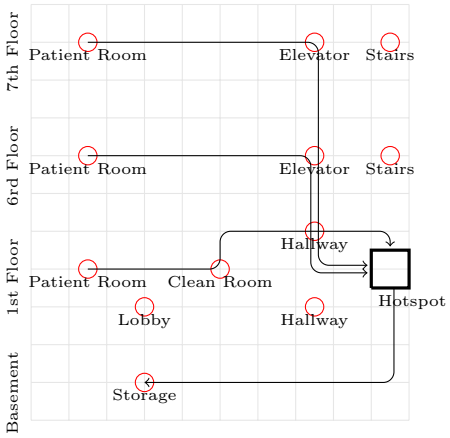
4.6.2 Anomaly Events

As shown in Figures 4.1 and 4.7, the proposed clustering algorithm can successfully discover almost all the locations which contain missing events labeled by domain experts. We identified 7 hotspots on two floors of a building. A semantic summary of these hotspots is provided in Table 4.1. Particularly, with the weighted density clustering algorithm, we can effectively eliminate noise events. For example, in the circle shown in Figure 4.7b, the number of events with small weight is much higher than that of those with large weight. Thus, the average weight is less than the specified criterion. Indeed, many of these events actually end within the cluster above the circle.

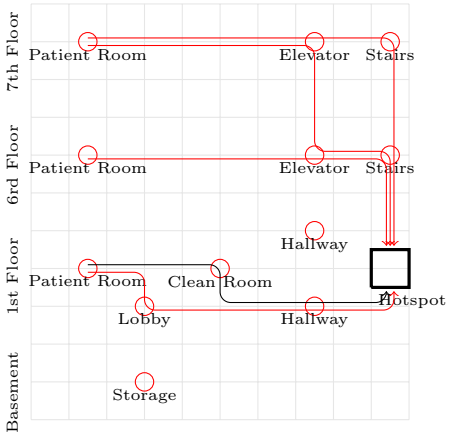
Floor	Hotspot	Summary
1st Floor	top	Patient room area
1st Floor	middle	Near exit and elevator
1st Floor	bottom	Patient room area
2nd Floor	left	Storage, near exit and stairs
2nd Floor	middle left	Near stairs
2nd Floor	middle right	Near stairs
2nd Floor	right	Patient room area

Table 4.1. A semantic summary of the identified hotspots.

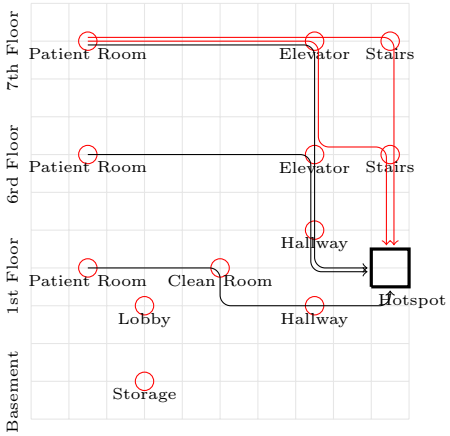
With the developed anomaly degree measurement, we can rank all location traces according to their anomaly degree from high to low. Figure 4.8 shows top-5 abnormal location traces of the detected missing events. The red color indicates that the corresponding detected event is confirmed in the benchmark. In Figure 4.8a, we plot the examples of normal location traces. In Figure 4.8b, we show top-5 abnormal location traces based on our methods. To make a comparison, we show top-5 abnormal location traces in Figure 4.8c based on the baseline methods. By comparing the results in Figures 4.8b and 4.8c, we can see the proposed method performs better than the baseline methods. Specifically, the abnormal events detected by our method went through rare transition patterns. In contrast, the baseline methods cannot differentiate events according to their historical transitions or have difficulty when measuring the anomaly degree accurately, and thus have a higher rate of false alarms.



(a) Normal Transitions

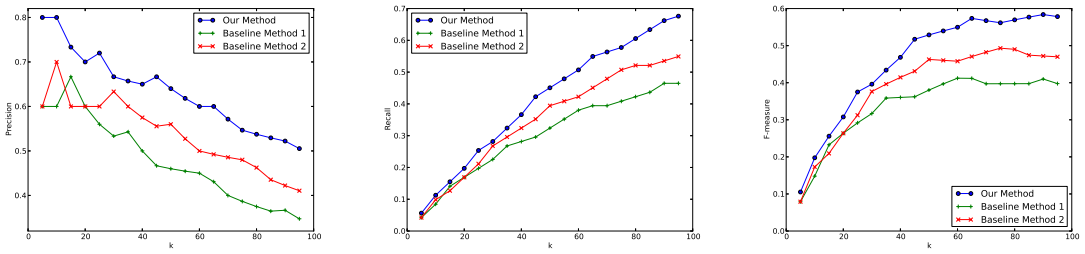


(b) Our Method



(c) The Baseline Methods

Figure 4.8. An illustration of detected abnormal events. The red sequences are confirmed in benchmark data.



(a) Top k precision (b) Top k recall (c) Top k f-measure

Figure 4.9. The accuracy comparison of the proposed method and the baseline methods. The blue lines on the top are of the proposed method. The other lines on the bottom are of the baseline methods.

4.6.3 A Performance Comparison

In Figure 4.9, we evaluate the performances of our method and the baseline methods with respect to the precision, recall and F-measure defined in Equation 4.18, 4.19 and 4.20. We identified top k abnormal events by these methods, where k ranges to 100 with a step of 5. We can see that our method outperforms the baseline methods in terms of both precision and recall consistently. Moreover, the results of our method are more stable than those of the baseline methods. For example, in Figure 4.9a, we can see there is dramatic change at the beginning of the precision curve for the baseline methods. This is not a surprise because the baseline methods cannot distinguish the different transitions ending within the same core point or have difficulty when measuring the anomaly degree accurately. In contrast, our method integrates the information from both the hotspots and the transition pattern with uncertainty addressed, thus leads to more stable results.

4.7 Related Work

Related work can be grouped into four categories. The first category includes the work related to trajectory outlier detection, which is most relevant to the main theme of this chapter. In the second category, we introduce some more general studies of trajectory data, such as trajectory clustering and trajectory pattern mining. The third category includes the relevant studies for finding area of interest from trajectory data. Finally, we introduce recent studies on activity recognition in the indoor scenarios.

The first category includes the work on trajectory outlier detection, which is highly related to this work. For instance, Lee et al. Lee et al. [2008] proposed a two-phase trajectory partition strategy for detecting trajectory outliers. This work has exploited both distance and density information for outlier detection. In Bu et al. [2009], an outlier detection framework was proposed for monitoring anomalies over continuous trajectory streams. The key idea is to build local clusters upon trajectory streams and detect anomalies by a cluster join mechanism. Li et al. Li et al. [2009] proposed a method for detecting temporal trajectory outliers with an emphasis on historical similarity trends among data points. Outliers could be determined if they have drastic changes from the historical trend and this drastic change is only observed in the membership of neighborhoods. In Ge et al. [2011c], Ge et al. proposed to quantify the outlier score in two ways and combined these two types of outlier scores together based on the Dempster-Shafer Theory. An incremental semi-supervised learning method was developed in R.R.Sillito and R.B.Fisher [2008] for trajectory outlier detection. This work is along the line of a learning approach and requires the training data. Finally,

Ge et al. Ge et al. [2010] proposed an evolving outlier detection method to detect the outlier trajectories in an evolving way. However, most methods above were developed to detect the outliers from outdoor trajectories, and not suitable for the analysis of specific indoor trajectories.

The second category includes the work on more general analysis of trajectory data, such as trajectory clustering and trajectory pattern mining. For instance, Giannotti et al.Giannotti et al. [2007b] introduced trajectory patterns as the concise descriptions of frequent behaviors in terms of both time and space. Also, a trajectory clustering algorithm was proposed in Lee et al. [2007]. This clustering algorithm first partitions the trajectories according to the Minimum Description Length (MDL) principle and then clustered the trajectory segments using a line-segment clustering algorithm. In Jeung et al. [2008], a filter-refinement approach was developed for discovering convoys in trajectory databases. Moreover, people have various interests in developing similarity and distance measures for trajectories Chen et al. [2005], Vlachos et al. [2002]. Finally, in Wang et al. [2008], Wang et al. proposed to co-cluster trajectories and semantic regions with the Dual Hierarchical Dirichlet Process (Dual-HDP) model, by treating trajectories as documents and positions as words.

The third category includes the work on the detection of area of interest with trajectory data. For instance, Liu et al. Liu et al. [2010a] proposed a non-density-based approach, called mobility-based clustering, to identify the hotspots of moving vehicles in an urban area. The key idea is that sample objects are used as "sensors" to perceive the vehicle crowdedness in nearby areas using their instant mobility, rather than the "object representatives". Moreover, Zheng et al. proposed a stay point concept and

identified hotspots from human moving trajectory Zheng et al. [2009b]. One location was considered as a hotspot if a lot of moving objects stay nearby over a certain time period controlled by a time threshold. In addition, Giannotti et al.Giannotti et al. [2007b] used the neighborhood function to model Regions-of-Interest. Basically, they partitioned the spatial space into grids and quantified the interest of each grid with the density and the direction information of each grid. Even some methods above are very successful for analyzing outdoor location traces, most of them are not applicable to the proposed indoor hospital environments because of the unique characteristics of these indoor location traces.

Finally, for the location based pattern mining research in indoor scenarios, there exist some studies focused on the activity recognition. For instance, in Chai and Yang [2005], Chai et al. exploited a dynamic Bayesian model to recognize the activity goals of the tracked object in a controlled experiment environment. Focused on the segmentation of the sensor signal sequences, in Yin et al. [2005], Yin et al. modeled the transition sequence as a Markov process conditioned on goals. However, most of the studies in this direction focused on recognizing activity goals for single or a few tracked objects who travel with the sensor devices intensively. Moreover, these studies are usually deployed in the school environment. The activities are not constrained by routine work flow patterns which are typical in the hospital environment. Indeed, in this chapter, our work is based on the real-world hospital scenario and has a focus on the missing event detection using indoor location traces of all the medical devices. This is a very unique problem setting, while the traditional methods for activity recognition in indoor environments are not designed for this purpose.

4.8 Summary

In this chapter, we provided a pilot study of detecting anomaly events of medical devices in indoor hospital environments, where all the medical devices have sensors attached and a real-time localization system can locate the position of any medical device. Since the activities of these medical devices have been constrained by the work flow in the hospital, there are a lot of context information which is available for the anomaly analysis. Therefore, we proposed a context-aware stochastic model for anomaly detection in indoor location traces. Along this line, we first identified the hotspots, the places with high-level anomaly activities. These hotspots can then be used as the context for the passing trajectories. This context-aware model could help us to identify different categories of anomaly events, such as the missing events due to the blocking signal, the defective sensor devices, and the devices being stolen. The key idea of this stochastic model is to exploit the hospital work flow and model the movements of medical devices as the transitions in finite state machines. As a result, the stochastic properties of the hotspots and the transition patterns can be estimated by the analysis of historical location traces. Finally, we have performed extensive experiments on real-world indoor location data in a hospital environment. The results clearly showed the different anomaly activities captured by the context-aware anomaly detection model. Also, we have demonstrated the effectiveness of missing event detection in terms of the detection accuracy.

CHAPTER 5

POPULARITY MODELING FOR APP RECOMMENDATION SERVICES

5.1 Introduction

With the rapid development of mobile App industry, the number of mobile Apps available has exploded over the past few years. For example, as of the end of April 2013, there are more than 1.6 million Apps at Apple's App store and Google Play. To facilitate the adoption of mobile Apps and learn the user experience with mobile Apps, many App stores enable the periodical (such as daily) App chart rankings and allow users to post ratings and comments for their Apps. Indeed, such popularity information plays an important role in App recommendation services Böhmer et al. [2013], Shi and Ali [2012], Yan and Chen [2011], and opens a venue for mobile App understanding, trend analysis and other App recommendation services Lim et al. [2010], Wu et al. [2012].

While people have developed some specific approaches to explore the popularity information of mobile Apps for some particular tasks Yan and Chen [2011], Zhu et al. [2012], the use of popularity information for App recommendation services is still fragmented and under-researched. Indeed, there are two major challenges along this line. First, the popularity information of mobile Apps often varies frequently and has the instinct of sequence dependence. For example, although the daily rank-

ings of different mobile Apps may be different, it is impossible that an App with a high ranking will be ranked very low in the following day due to the momentum of popularity. Second, the popularity information is heterogeneous, but contains latent semantics and relationships. For example, although *Ranking=1* and *Rating=5* are totally different observations, they may both indicate some sort of popularity.

To this end, in this work, we propose a sequential approach based on Hidden Markov Model (HMM) to model the heterogeneous popularity information of mobile Apps. Particularly, the objective of our approach is to provide the comprehensive modeling of popularity information towards App recommendation services. Along this line, we first propose a Popularity based HMM (PHMM) model by extending the HMM with heterogeneous popularity information of mobile Apps, including chart rankings, user ratings and review topics extracted from comments. The popularity information can be modeled in terms of different transitions of different popularity states, which indicate the latent semantics and relationships of popularity observations. Then, to efficiently train our PHMM model, we introduce a bipartite based method to pre-cluster various popularity observations. The pre-cluster results can be leveraged for choosing parameters and the initial values of our PHMM model. Although many applications may benefit from the results of our PHMM model, in this work, we focus on demonstrating several novel App recommendation services enabled by our PHMM model, including App recommendation, rating and comment spam detection, and ranking fraud detection. Finally, to validate the proposed model, we carry out extensive experiments on a real-world data set collected from Apple's App Store. The experimental results clearly demonstrate both the effectiveness and

efficiency of our approach.

5.2 Overview

We first introduce the popularity observations of mobile Apps, and then present the problem statement of popularity modeling and the overview of our model.

5.2.1 Preliminaries of Popularity Observations

In this work, we focus on three kinds of important popularity information, namely *chart rankings*, *users ratings* and *review comments*. We can collect periodical observations for each type of popularity information.

- **Chart Rankings.** Most of the App stores launch the chart rankings of Apps, which are usually updated periodically, e.g., daily. Therefore, each mobile App a has many historical ranking observations which can be denoted as a time series, $\mathcal{P}_a = \{p_1^a, \cdots, p_n^a\}$, where $p_i^a \in \{1, ..., K_p\}$ is the ranking position of a at time stamp t_i. Note that, the smaller value p_i^a is, the higher ranking the App a has. Intuitively, the higher ranking indicates the higher popularity.

- **User Ratings.** After an App is published, it can be rated by any user who downloaded it. Indeed, the user rating is one of the most important features of App popularity. Particularly, each rating can be categorized into K_r discrete rating levels, e.g., 1 to 5, which represent users' different preferences for Apps. Therefore, the rating observations of an App a can also be denoted as a time series, $\mathcal{R}_a = \{r_1^a, \cdots, r_n^a\}$, where $r_i^a \in \{1, \cdots, K_r\}$ is the user rating posted at time stamp t_i.

- **Review Comments.** Besides ratings, users can also put their comments on each App. Each comment reflects a user's personal perception for a particular App. Similar as user ratings, we can denote all comments of an App a as a time series, $\mathcal{C}_a = \{c_1^a, \cdots, c_n^a\}$, where c_i^a is the comment posted at time stamp t_i. Note that, the number of comments is always the same as that of ratings for a single App.

Indeed, all popularity observations on the above are important for mobile App recommendation. However, different from the ranking and rating observations, it is hard to directly leverage comments as observations for modeling App popularity. Therefore, in this work we propose to leverage topic modeling Blei et al. [2003] to extract the latent semantics of user comments as popularity observations. The intuitive motivation is mapping each comment onto a specific *review topic*, which is easy to understand and exploit for opinion analysis. Specifically, in this work we adopt the widely used Latent Dirichlet Allocation (LDA) model Blei et al. [2003] for learning latent semantic topics. To be more specific, the historical comments of a mobile App a, i.e., $\mathcal{C}_a$, is assumed to be generated as follows. First, before generating $\mathcal{C}_a$, K_z prior conditional distributions of words given latent topics $\{\phi_z\}$ are generated from a prior Dirichlet distribution β. Second, a prior latent topic distribution θ_a is generated from a prior Dirichlet distribution α for each mobile App a. Then, for generating the j-th word in C_a denoted as $w_{a,j}$, the model firstly generates a latent topic z from θ_a and then generates $w_{a,j}$ from ϕ_z. Particularly, Figure 5.1 shows the graphic representation of the LDA model, where M is the number of mobile Apps, N is the number of all

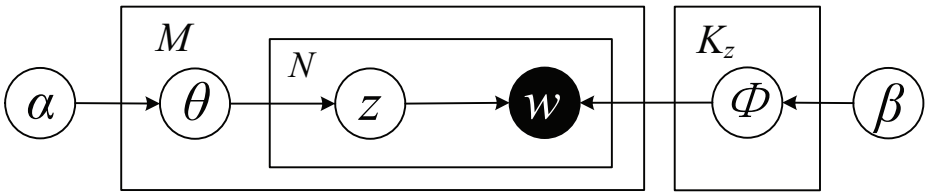


Figure 5.1. The graphic representation of LDA model.

unique words in comments, and K_z is the predefined number of latent topics. The training process of LDA model is to learn proper latent variables $\theta = \{P(z|\mathcal{C}_a)\}$ and $\phi = \{P(w|z)\}$ for maximizing the posterior distribution of comment observations, i.e., $P(\mathcal{C}_a|\alpha, \beta, \theta, \phi)$. In this work, we use a Markov chain Monte Carlo method named Gibbs sampling Griffiths and Steyvers [2004] for training LDA model. Then we can map each comment c_i^a onto a review topic z_i^a by

$$z_i^a = \arg\max_z P(z|c_i^a) \propto \arg\max_z \Big(\prod_{w \in c_i^a} P(w|z)P(z) \Big), \tag{5.1}$$

where both $P(w|z)$ and $P(z)$ can be learned via training LDA model. Therefore, we can obtain the time series of comments as $\mathcal{Z}_a = \{z_1^a, \cdots, z_n^a\}$, where $z_i^a \in \{1, \cdots, K_z\}$

In reality, the time stamps of ranking observations and user rating (comment) observation are usually not identical. For example, for a particular App, there may be multiple ratings and comments from multiple users per day, while there may be only one ranking observation per day. Therefore, we first aggregate the ranking, and rating (comment) observations together to get the integrated observations, which share the same time stamp with the ranking observations. Moreover, in practice we may need to model App popularity with different time granularity, such as daily or weekly. The aggregation can allow us to get the integrated observations with

different time interval/granularity. After getting the integrated observations with the same time stamps, we can represent the heterogeneous observations of popularity for a mobile App a as a sequence as $\mathcal{O}_a = \{o_1^a, \cdots, o_n^a\}$, where $o_i^a = \{\mathcal{P}_i^a, \mathcal{R}_i^a, \mathcal{Z}_i^a\}$ contains the observations of ranking, rating and review topic during time interval t_i.

5.2.2 Problem Statement

We define the problem of popularity modeling for mobile Apps as follows:

Definition 10 (Problem Statement) *Given a set of mobile Apps A, where each App $a \in A$ has a sequence of historical popularity observations $\mathcal{O}_a = \{o_1^a, \cdots, o_n^a\}$. The problem of popularity modeling is to learn a model $\mathcal{M}$ from all the observation sequences $\{O_a | a \in A\}$, which can be used for predicting the popularity observations for each mobile App in the future.*

However, it is not a trivial problem to model mobile App popularity. First, the popularity information of mobile Apps often varies frequently and has the instinct of sequence dependence. Second, the popularity information is heterogeneous but contains latent semantics and relationships. To solve those challenges, we propose a novel approach for popularity modeling based on Hidden Markov Model (HMM), which is widely used for modeling sequential observations. Specifically, we assume that there are multiple latent *Popularity States* of mobile Apps, such as *very popular*, *popular*, and *out-of-popular*, and different kinds of popularity observations appear for one App at the same time because they all belong to the same popularity state. Moreover, the varying of popularity observations is due to the transitions of different popularity states. Figure 5.2 shows the graphic representation of our Popularity based

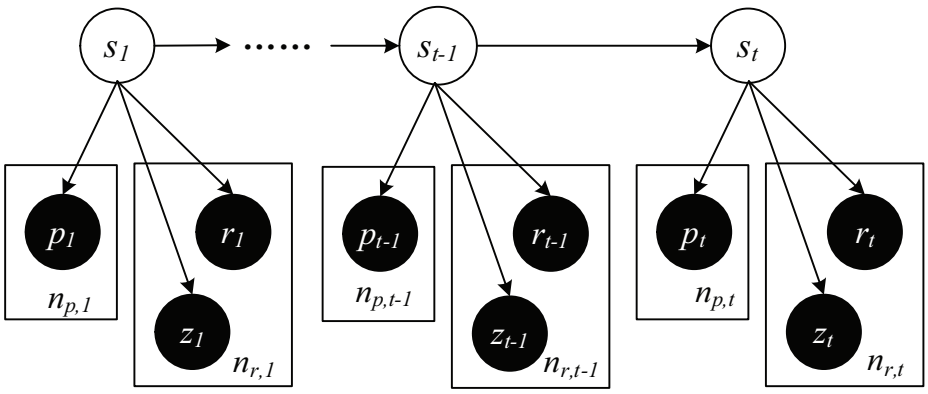


Figure 5.2. The graphical structure of PHMM model.

HMM (PHMM) model. In this figure, each observation o_i is generated by the latent state s_i, and o_i contains $n_{p,i}$ rankings, $n_{r,i}$ ratings and review topics. Particularly, here we adopt the widely used first order Morkov assumption in our model. In other words, the probability distribution of each popularity state s_i is independent of the previous states $s_1, \cdots, s_{i-2}$, given the immediately previous state s_{i-1}, i.e., $P(s_i|s_1, \cdots, s_{i-1}) = P(s_i|s_{i-1})$.

Indeed, with this PHMM model for modeling App popularity, there are two main problems to be resolved. First, *how to train the PHMM model with respect to different kinds of popularity observations?* Second, *how to choose the proper number of latent popularity states for PHMM?* In the following section, we will present our solutions for both problems.

5.3 App Popularity Modeling

Now we present the details of App popularity modeling by the PHMM model.

5.3.1 Training PHMM Model

Given a set of popularity states $\mathcal{S} = \{s_1, \cdots, s_{K_s}\}$, a set of App rankings $\mathcal{P} = \{p_1, \cdots, p_{K_p}\}$, a set of user ratings $\mathcal{R} = \{r_1, \cdots, r_{K_r}\}$, and a set of review topics $\mathcal{Z} = \{z_1, \cdots, z_{K_z}\}$, the PHMM can be represented by a model containing three different probability distributions:

- The transition probability distribution $\Delta = \{P(s_i|s_j)\}$, where $s_i, s_j \in \mathcal{S}$ are the latent popularity states.
- The initial state distribution $\Psi = \{P(s_i)\}$, where $P(s_i)$ is the probability that popularity state s_i occurs as the first element of a state sequence.
- The emission probability distribution $\Lambda = \{P(\mathcal{P}, \mathcal{R}, \mathcal{Z}|s_i)\}$, where $P(\mathcal{P}, \mathcal{R}, \mathcal{Z}|s_i)$ is the joint probability that popularity observations $\mathcal{P}$, $\mathcal{R}$ and $\mathcal{Z}$ are generated by state s_i.

As the common setting of HMM, here we assume that ranking, rating and review topics are conditionally independent given the popularity state, i.e., $P(\mathcal{P}, \mathcal{R}, \mathcal{Z}|s_i) \equiv \prod_{p\in\mathcal{P}} P(p|s_i) \prod_{r\in\mathcal{R}} P(r|s_i) \prod_{z\in\mathcal{Z}} P(z|s_i)$. Therefore, we can denote the emission probability distribution Λ by the triple $(\Lambda_p, \Lambda_r, \Lambda_z) \equiv (\{P(p|s_i)\}, \{P(r|s_i)\}, \{P(z|s_i)\})$, which stratifies $\sum_p P(p|s_i) = \sum_r P(r|s_i) = \sum_z P(z|s_i) = 1$.

Therefore, given a set of training sequences of popularity observations $\mathcal{X} = \{\mathcal{O}_1, \cdots, \mathcal{O}_N\}$, the task of training PHMM is to learn the set of parameters $\Theta = (\Psi, \Delta, \Lambda_p, \Lambda_r, \Lambda_z)$. Specifically, we denote the length of sequence $\mathcal{O}_n$ as L_n and the j-th observation $o_j \in \mathcal{O}_n$ as an triple $(\mathcal{P}_{n,j}, \mathcal{R}_{n,j}, \mathcal{Z}_{n,j})$. Moreover, we let $p_{n,j,k}$, $r_{n,j,k}$,

and $z_{n,j,k}$ denote the k-th ranking, rating and review topic in $\mathcal{P}_{n,j}$, $\mathcal{R}_{n,j}$, and $\mathcal{Z}_{n,j}$. Therefore, we can use the Maximum Likelihood Estimation (MLE) to compute the optimal parameters Θ^* by

$$\Theta^* = \arg\max_{\Theta} \log P(\mathcal{X}|\Theta) = \arg\max_{\Theta} \sum_{n} \log P(\mathcal{O}_n|\Theta). \tag{5.2}$$

Here, we denote all the possible state sequences of an observation sequence $\mathcal{O}_n$ as a set $\Omega_n = \{S_1^n, \cdots, S_M^n\}$, and define a variable $\mathcal{Y}_n \in \Omega_n$, where each S_m^n is a state sequence with length L_n. Moreover, we denote the j-th state in S_m^n as $s_{m,j}^n$. Accordingly, we can define $\mathcal{Y} = \{\mathcal{Y}_1, \cdots, \mathcal{Y}_N\}$ as the set of all possible state sequences. Then we can rewrite the likelihood $\log P(\mathcal{O}_n|\Theta)$ in Equation 5.2 as

$$\log P(\mathcal{O}_n|\Theta) = \log \sum_{m} P(\mathcal{O}_n, S_m^n|\Theta), \tag{5.3}$$

where the joint distribution can be written as

$$\begin{aligned} P(\mathcal{O}_n, S_m^n|\Theta) &= P(\mathcal{O}_n|S_m^n, \Theta) P(S_m^n|\Theta) \\ = \left(\prod_{j=1}^{L_n} \prod_{k} P(p_{n,j,k}|s_{m,j}^n) \prod_{i} P(r_{n,j,i}|s_{m,j}^n) P(z_{n,j,i}|s_{m,j}^n) \right) \\ \times \left(P(s_{m,1}^n) \prod_{j=2}^{L_n} P(s_{m,j}^n|s_{m,j-1}^n)) \right). \end{aligned} \tag{5.4}$$

Indeed, directly optimizing the above likelihood function is not a trivial problem. In this work, we propose to exploit the Expectation Maximization (EM) algorithm to iteratively estimate the parameters.

Specifically, at the *E-Step*, we have

$$\begin{aligned} Q(\Theta, \Theta^{(i-1)}) &= \mathbf{E}\Big[\log P(\mathcal{X}, \mathcal{Y}|\Theta); \mathcal{X}, \Theta^{(i-1)} \Big] \\ &= \sum_{n,m} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \log P(\mathcal{O}_n, S_m^n|\Theta), \end{aligned} \tag{5.5}$$

where $\Theta^{(i-1)}$ is the set of model parameters estimated in the last round of EM iteration. Particularly, we can estimate $P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)})$ by

$$P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \propto P(\mathcal{O}_n, S_m^n|\Theta^{(i-1)}). \tag{5.6}$$

At the *M-step*, we maximize $Q(\Theta, \Theta^{(i-1)})$ iteratively until it converges by estimating the model parameters as follows,

$$P(s_i) = \frac{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)})\delta(s_{m,1}^n = s_i)}{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)})}, \tag{5.7}$$

$$P(p|s_i) = \frac{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \sum_j \delta(s_{m,j}^n = s_i \wedge p \in \mathcal{P}_{n,j})}{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \sum_j \delta(s_{m,j}^n = s_i) N_{\mathcal{P}_{n,j}}}, \tag{5.8}$$

$$P(r|s_i) = \frac{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \sum_j \delta(s_{m,j}^n = s_i \wedge r \in \mathcal{R}_{n,j})}{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \sum_j \delta(s_{m,j}^n = s_i) N_{\mathcal{R}_{n,j}}}, \tag{5.9}$$

$$P(z|s_i) = \frac{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \sum_j \delta(s_{m,j}^n = s_i \wedge z \in \mathcal{Z}_{n,j})}{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)}) \sum_j \delta(s_{m,j}^n = s_i) N_{\mathcal{Z}_{n,j}}}, \tag{5.10}$$

$$P(s_i|s_j) = \frac{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)})\delta(\exists t\ s_{m,t-1}^n = s_j \wedge s_{m,t}^n = s_i)}{\sum_{m,n} P(S_m^n|\mathcal{O}_n, \Theta^{(i-1)})\delta(\exists t\ s_{m,t-1}^n = s_j)}, \tag{5.11}$$

where $\delta(x) = 1$ if $x = True$, and 0 otherwise; $N_{\mathcal{P}_{n,j}}$, $N_{\mathcal{R}_{n,j}}$, and $N_{\mathcal{Z}_{n,j}}$ are the number of unique ranking, rating, topic observations in $\mathcal{P}_{n,j}$, $\mathcal{R}_{n,j}$, and $\mathcal{Z}_{n,j}$. Furthermore, the above equations can be efficiently computed by the Forward-Backward algorithm Rabiner [1989].

5.3.2 Choosing the Number of Popularity States

Another problem of training PHMM model is how to choose the proper number of latent popular states. Indeed, a common used approach for estimating the latent state of HMMs is to leverage domain knowledge or some existing algorithms to pre-cluster the observations Cao et al. [2009a]. In our problem, the popularity observations contain $K_p + K_r + K_z$ elements, i.e., K_p unique rankings, K_r unique ratings, and K_z unique topics. Intuitively, these observations are heterogeneous and contain internal

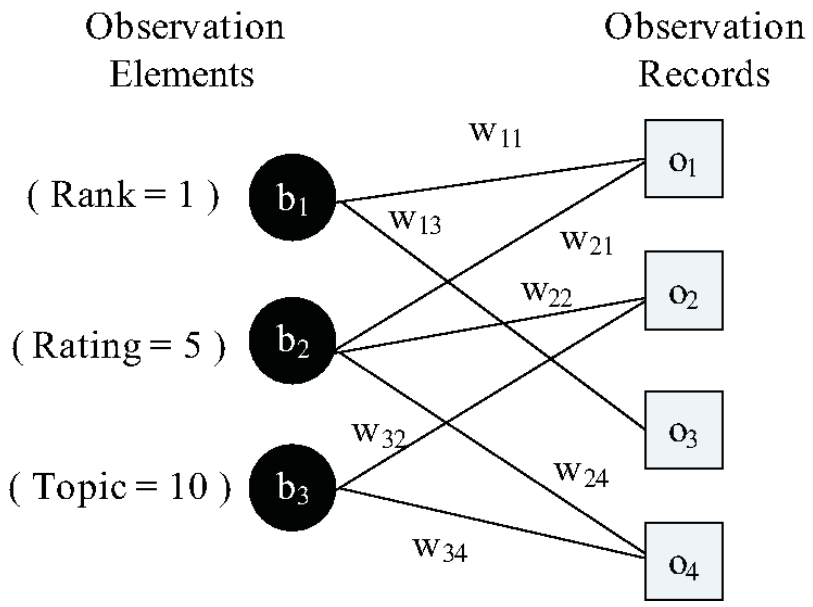


Figure 5.3. An example of E-R bipartite graph.

relationships. Thus how to cluster such information is an open question. To solve this problem, in this work we propose a novel clustering method based on the Element-Record (E-R) bipartite graph. Specifically, the E-R bipartite graph can be denoted as $G = \{V, E, W\}$ where $V = \{V^b, V^o\}$. $V^b = \{b_1, \cdots, b_K\}$ denotes the set of unique observation elements, i.e., $K = K_p + K_r + K_z$, and $V^o = \{o_1, \cdots, o_M\}$ denotes the set of all observation records from sequence set $\mathcal{X}$. Edge set is $E = \{e_{ij}\}$, where e_{ij} denotes the observation element b_i has appeared in record o_j. Edge weight set is $W = \{w_{ij}\}$, where each w_{ij} represents the normalized frequency of the appearance of b_i in o_j. For example, if an observation element $b_i = (Rating = 5)$ has appeared n_i times in o_j and there are totally n_j ratings in o_j, the weight would be $w_{ij} = \frac{n_i}{n_j}$. Figure 5.3 shows an example of the E-R bipartite graph.

Therefore, given an E-R bipartite graph, we can denote each unique observation element as a normalized vector $\overrightarrow{b_i} = dim[M]$, where M is the number of all unique ob-

servation records, $dim[j] = \frac{w_{ij}}{\sum_k w_{ik}}$ is the normalized dimension of vector. Accordingly, we can estimate the similarity between two popularity observations by calculating the Cosine distance between their vectors. After that, many existing algorithms can be leveraged for estimating the number of clusters, such as density based clustering algorithms. In this work, we utilize a clustering algorithm proposed in Cao et al. [2008], which is robust for high dimensional data and only needs a parameter to indicate the minimum average mutual similarity S_{min} for the data points in each cluster. The average mutual similarity for a cluster C is calculated as $S_C = \frac{2\times\sum_{1\leq i<j\leq|C|} Sim(b_i,b_j)}{|C|\times(|C|-1)}$, where $|C|$ indicates the number of observation elements in C and $Sim(b_i, b_j)$ is the similarity between the i-th and j-th observation elements in C.

The results of pre-clustering may not be the true popularity states learned by PHMM. However, we believe the pre-clustering can provide positive guidance for estimating the number of latent states due to the intrinsic relationships between popularity observations. Furthermore, the results of pre-clustering can also be used for assigning initial values of EM algorithm. Actually, the basic EM algorithm implemented by randomly assigning initial values for model parameters Θ, which may lead to more training iterations and unexpected local optimal results. Particularly, if we treat each popularity cluster C_i as the latent state s_i, we can estimate the initial values of parameters Θ as follows. First, we define the prior distribution of observation element b_i $(b = p, r, z)$ as $P(b_i)$, which can be computed by the MLE method. Specifically, $P(b_i) = \frac{N_{b_i}}{\sum_k N_{b_k}}$, where N_{b_k} is the appearance frequency of b_k in all observation records. Second, for each observation element b_i, we can first compute the probability $P(s_j|b_i)$ by the normalized similarity between observation vector $\overrightarrow{b_i}$ and

cluster C_j, i.e., $P(s_j|b_i) = \frac{Sim(\overrightarrow{b_i},\overrightarrow{C_j})}{\sum_k Sim(\overrightarrow{b_i},\overrightarrow{C_k})}$, where $Sim(*,*)$ is the Cosine distance and $\overrightarrow{C_j} = Norm(\frac{\sum_{b\in C_i}\overrightarrow{b}}{|C_i|})$ is the normalized centroid of cluster. Actually, here we assume that an observation has higher probability of belonging to a nearer cluster. Therefore, we have following estimations:

- The initial state distribution can be computed by the summation $P^0(s_i) = \sum_{b=p,r,z}\sum_k P(s_i|b_k)P(b_k)$.
- The emission probability can be computed according to the Bayes rule, i.e., $P^0(b_i|s_j) = \frac{P(s_j|b_i)P(b_i)}{P^0(s_j)}$.
- The transition distribution $P^0(s_i|s_j)$ can be computed by the normalized similarity between cluster C_i and C_j, i.e., $P^0(s_i|s_j) = \frac{Sim(\overrightarrow{C_i},\overrightarrow{C_j})}{\sum_k Sim(\overrightarrow{C_k},\overrightarrow{C_j})}$.

Indeed, our experimental results clearly validate that using pre-clustering for assigning initial values of EM algorithm can accelerate the training process and enhance the model fitting of PHMM.

5.4 PHMM Model Application

There are many applications which can be derived from our PHMM model. But, in this work, we focus on demonstrating several novel recommendation related applications motivated by our PHMM model, including trend based mobile App recommendation, rating and comment spam detection, as well as ranking fraud detection for mobile Apps.

Particularly, our PHMM model can be learned from different time and App granularity. For example, as introduced in Section 5.2, we can use the daily, weekly

or monthly observations from one or more Apps for modeling popularity. Different model granularity may generate different popularity patterns and lead to different applications. Particularly, in this work we mainly focus on learning PHMM model from all mobile Apps, which can be used for capturing the common popularity patterns and relationships of mobile Apps.

Assume that we have observed a popularity sequence $\mathcal{O}_a = \{o_1^a, \cdots, o_t^a\}$ from a mobile App a. Thus we can estimate the latent states for the i-th observation o_i^a $(0 \leq i \leq t)$ by

$$P(s|o_i^a, \Theta) \propto P(o_i^a|s, \Theta)P(s|\Theta) = \prod_{p,r,z \in o_i^a} P(p|s) \times P(r|s)P(z|s) \sum_{s'} P(s|s')P(s'|o_{i-1}^a, \Theta), \tag{5.12}$$

which can be computed effectively by Forward-Backward algorithm or Viterbi algorithm Viterbi [1967]. Similarly, we can predict the $(t+1)$-st popularity state for App a by $P(s^{(t+1)} = s|\mathcal{O}_a, \Theta) = \sum_{s'} P(s|s')P(s'|o_t^a, \Theta)$. Based on the above, we can conduct the following three App recommendation related applications.

• **Trend based Mobile App Recommendation.** The existing mobile App recommender systems usually recommend Apps which were popular in the past. This is not proper in practice because the popularity information is always varying frequently, and mobile users tend to follow the future popularity trend of Apps. Therefore, in this work we propose a trend based App recommendation approach by leveraging our PHMM model. Specifically, given a t-length observation sequence of mobile App

a, i.e., $\mathcal{O}_a = \{o_1^a, \cdots, o_t^a\}$, we can predict the possible rankings and ratings of a at next time stamp $t+1$ by $P(p^{(t+1)} = p|\mathcal{O}_a, \Theta) = \sum_s P(p|s)P(s^{(t+1)} = s|\mathcal{O}_a, \Theta)$, and $P(r^{(t+1)} = r|\mathcal{O}_a, \Theta) = \sum_s P(r|s)P(s^{(t+1)} = s|\mathcal{O}_a, \Theta)$, where $s^{(t+1)}$ is the $(t+1)$-st popularity state of $\mathcal{O}_a$. Furthermore, we can compute the ranking and rating expectations of App a at time stamp $t+1$ by $p_a^* = \sum_p p \times P(p^{(t+1)} = p|\mathcal{O}_a, \Theta)$ and $r_a^* = \sum_r r \times P(r^{(t+1)} = r|\mathcal{O}_a, \Theta)$. Similarly, we can rank all mobile Apps with respect to their ranking and rating expectations, and obtain two ranked list Υ_{Rank} and Υ_{Rate}. Then, we can calculate the final popularity score of each mobile App by Borda's ranking fusion method:

$$P\_Score(a) = \alpha \times \frac{1}{RK_{Rank}(a)} + (1-\alpha) \times \frac{1}{RK_{Rate}(a)}, \tag{5.13}$$

where α is the fusion parameter; $RK_{Rank}(a)$ and $RK_{Rate}(a)$ is the ranking of a in ranked list Υ_{Rank} and Υ_{Rate}. Particularly, if $\alpha = 0$, the final rank is only based on the rating trend, which is similar to the ranked list Υ_{Rate}. If $\alpha = 1$, the final rank is only based on the ranking trend, which is similar to the ranked list Υ_{Rank}. The score $P\_Score(a)$ indicates the popularity trend in the future, thus can be used for recommending Apps.

- **Rating and Comment Spam Detection.** User ratings and comments are the important information in mobile App market. The App store provider and the developers of Apps rely on the ratings and comments of users a lot to get helpful feedback from various users. However, some of the shady users may post deceptive ratings and comments with the purpose of inflating or deflating corresponding mobile Apps. Many efforts have been made in the literatures for detecting such rating and

comments spams Lim et al. [2010], Wu et al. [2012], Xie et al. [2012]. However, few of them took the sequence characteristics of App popularity into consideration. In this work, we propose a novel approach based on PHMM model for detecting rating and comment spams. Specifically, given a t-length observation sequence of mobile App a, i.e., $\mathcal{O}_a = \{o_1^a, \cdots, o_t^a\}$, we can first leverage the first $(t-1)$-length sequence to predict the possible t-th popularity states of a by Equation 5.12. Then, we can calculate the likelihood of the observations of rating and review topic at time stamp t by $\log P(\mathcal{R}_t^a|\Theta) = \log \sum_s P(\mathcal{R}_t^a, s^t = s|\Theta)$, and $\log P(\mathcal{Z}_t^a|\Theta) = \log \sum_s P(\mathcal{Z}_t^a, s^t = s|\Theta)$, which can be estimated by the similar way of Equation 5.4. Then, if the likelihood is less than the predefined thresholds τ_r, and τ_z, we believe that there are rating or comment spams in a at time stamp t.

- **Ranking Fraud Detection.** The ranking fraud of mobile Apps refers to fraudulent or deceptive activities which have a purpose of bumping up the rankings of Apps during a specific time period. Detecting such ranking fraud is very important for the healthy development of mobile App industry, especially for building mobile App recommender systems. Different from rating and comment spam, the ranking fraud always happens during some specific time periods. It is due to that people who try to manipulate the App rankings always have some specific expectations of ranking, such as top 25 for one month. Moreover, some of the normal promotion means, such as "Free App a Day", may also result in the abnormal ranking observations. Therefore, to detect the ranking fraud for mobile Apps, we should check the observation sequence during a time period but not at only one time stamp. To be specific, we can first define a sliding window with length T, and segment the popularity records of mobile Apps

Table 5.1. Statistics of the experimental data.

	Data Statistics
App Num.	9,784
Ranking Num.	285,900
Avg. Ranking Num.	29.22
Rating/Comment Num.	14,912,459
Avg. Rating/Comment Num.	1,524.17

a by several T-length observation sequences $\{\mathcal{O}_1^a, \cdots, \mathcal{O}_n^a\}$. Then, for each sequence $\mathcal{O}_i^a$ we will calculate its anomaly score by the average log-loss of ranking observations:

$$\mathcal{L}(\mathcal{O}_i^a) = -\frac{1}{T}\log P(\mathcal{P}_{\mathcal{O}_i^a}|\Theta) = -\frac{1}{T}\log \sum_m P(\mathcal{P}_{\mathcal{O}_i^a}, S_m^T|\Theta), \tag{5.14}$$

where $\mathcal{P}_{\mathcal{O}_i^a} = \{\mathcal{P}_{i,1}^a, \cdots, \mathcal{P}_{i,T}^a\}$ is the sequence of all ranking observations in $\mathcal{O}_i^a$, and each S_m^T is a state sequence with length T and the equation can be estimated in the similar way as Equation 5.4. Finally, if the anomaly score $\mathcal{L}(\mathcal{O}_i^a)$ is larger than a predefined threshold τ_p, we believe that the ranking fraud happens during the time period of $\mathcal{O}_i^a$.

5.5 Experimental Results

In this section, we evaluate the performance of our PHMM model by using a real-world App data set.

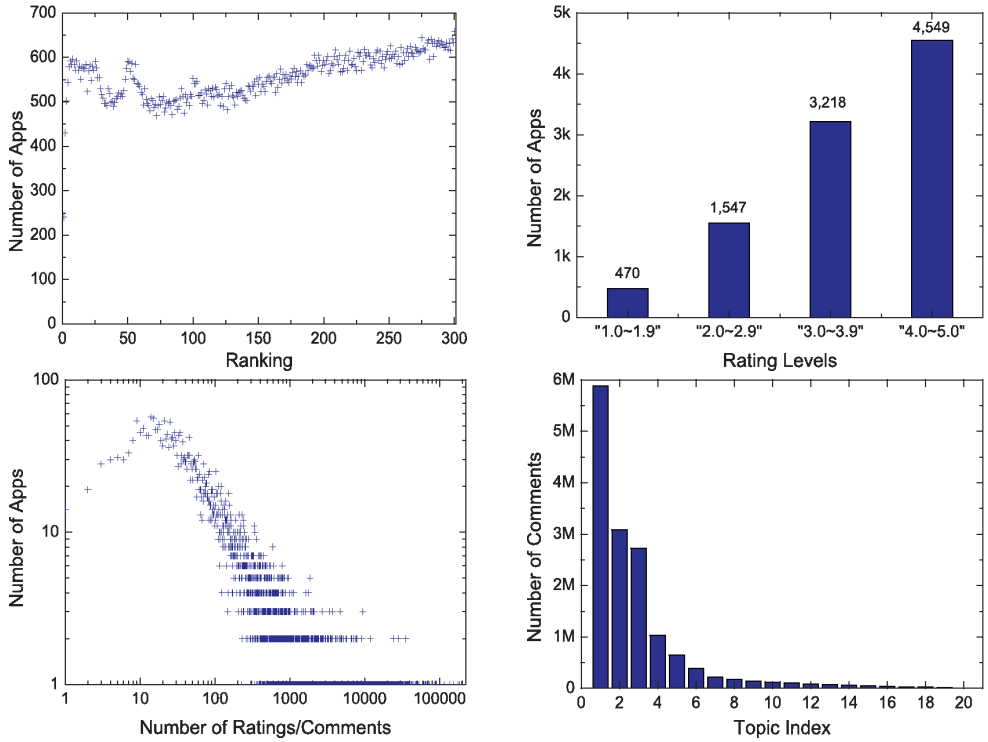


Figure 5.4. The distribution of the number of Apps w.r.t (a) different rankings, (b) different rating levels, (c) different number of ratings/comments, (d) the distribution of the number of comments w.r.t different topics.

5.5.1 The Experimental Data

The experimental data set was collected from the "Top Free 300" leaderboard of Apple's App Store (U.S.) from February 2, 2010 to September 17, 2012 [1] . The data set contains the daily chart rankings, which were collected at 11:00PM (PST) daily, all user ratings and review comments of top 300 free Apps during the period. Specifically, Table 5.1 shows some statistics of the data.

Figures 5.4 (a), (b) and (c) show the distributions of the number of Apps with respect to different rankings, different rating levels and different number of ratings/comments. Although the distributions of popularity observations are not even.

[1] **This data set will be made publicly available soon**.

Furthermore, we use the LDA model to extract review topics as introduced in Section 5.2. Particularly, we first remove all stop words (e.g., "of", "the") and normalize verbs and adjectives (e.g., "*plays* $\rightarrow$ *play*", "*better* $\rightarrow$ *good*") of each comment by the Stop-Words Remover [2] and the Porter Stemmer [3] . Then, the number of latent topic K_z is set to 20 according to the perplexity based estimation approach Zhu et al. [2012]. Two parameters α and β for training LDA model are set to be $50/K$ and 0.1 according to Heinrich [2008]. Figures 5.4 (d) shows the distribution of the number of comments w.r.t different topics. From the figure we can observe that only a few topics are frequently mentioned in comments.

5.5.2 The Performance of Training PHMM

Here, we demonstrate the performance of training PHMM model. Particularly, we treat the data of each day as an observation record. Therefore, for each observation record of a specific App, there are one chart ranking, several user ratings and review comments (topics). Note that, for each App a, we get the first (or last) record when it appears in the top 300 leaderboard for the first (or last) time during the time period (i.e., from February 2, 2010 to September 17, 2012). We treat the days that are the first and last time of App a to appear in the top 300 leardboard or have ratings/comments as the start and end records of observation sequence $\mathcal{O}_a$.

We first use the algorithm introduce in Cao et al. [2008] for pre-clustering popularity observations from all Apps, where the parameter S_{min} is empirically set as 0.5. After that, there are totally 13 clusters used for assigning initial values of PHMM

[2] http://www.lextek.com/manuals/onix/index.html

[3] http://www.ling.gu.se/~lager/mogul/porter-stemmer

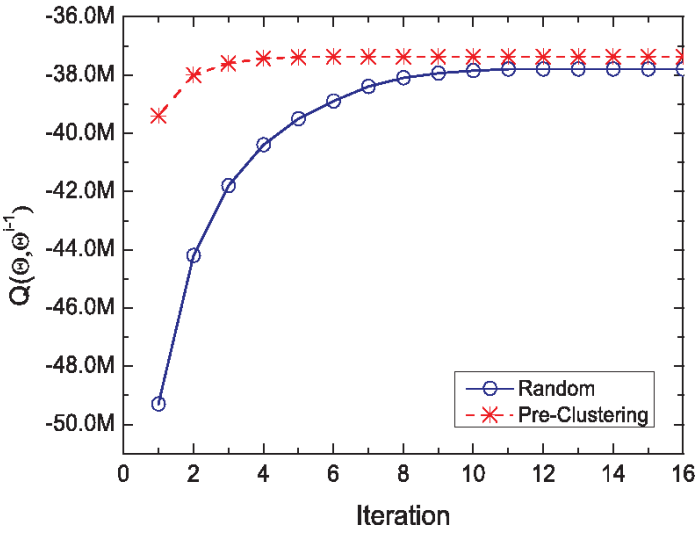


Figure 5.5. The value of the $Q(\Theta, \Theta^{(i-1)})$ of PHMM model with different initial value assignment.

parameters. Figure 5.5 shows the the value of the $Q(\Theta, \Theta^{(i-1)})$ of PHMM model with initial value assignment of model parameters randomly, and initial value assignment of model parameters by pre-clustering in each iteration. Particularly, both experiments are conducted five times and the figure shows the average values. We can observe that the pre-clustering approach can accelerate the training process of PHMM. Moreover, the PHMM model with initial value assignment by the pre-clustering can achieve the better fitting, which indicates the higher likelihood of PHMM model.

To further demonstrate the performance of our PHMM model, we show two examples of identified latent popularity states in Table 5.2 and Table 5.3. Note that, we manually transfer each review topic to semantic description, and due to the limited space we only show top 2 ranking, rating and topic observations which are most probable to appear in each state. In these tables, we can see that the latent popularity states are meaningful. For example, the states s_1 and s_6 may indicate the Apps are

Table 5.2. The PHMM state s_1.

Ranking=5
Ranking=15
Rating=5
Rating=4
Topic="Funny Apps"
Topic="Good Design"

Table 5.3. The PHMM state s_6.

Ranking=253
Ranking=148
Rating=3
Rating=4
Topic="Boring"
Topic="Old-Fashioned"

very popular and *out-of-popular* respectively.

Indeed, all the applications of PHMM model introduced in Section 5.4 are based on the prediction of popularity observations. Therefore, in this subsection we validate the effectiveness of PHMM model by evaluating its performance of predicting rankings, rankings and topics. To reduce the uncertainty of splitting the data into training and test data, in the experiments we utilize a five-fold cross validation to evaluate PHMM model. To be specific, we first randomly divide all observation sequences of mobile Apps into five equal parts, and then use each part as the test data while using other four parts as the training data in five test rounds. Particularly, for each test sequence we randomly select top T observation records for fitting model, and use the $(T+1)$-st observation record as ground truth for predicting probable rankings, ratings and topics. Moreover, in our experiments we find that the pre-clustering results of each training data set are very similar, thus we set the number of popularity states as 13 for all five PHMM models. To the best of our knowledge, there is no existing work of App popularity modeling has been reported. Thus, we develop two baselines for

evaluating our PHMM model, which are static and sequential approaches respectively.

The first baseline **CPP** stands for pre-Clustering based Popularity Prediction, which is a static approach for predicting popularity observations. Specifically, given a T-length observation sequence $\mathcal{O} = \{o_1, \cdots, o_T\}$, we predict the popularity observation $b^{(T+1)}$ $(b = p, r, z)$ by the probability $P(b^{(T+1)} = b|\mathcal{O}) = \sum_i P(b^{(t+1)} = b, C_i|\mathcal{O}) \propto \sum_i P(b^{(T+1)} = b|C_i)P(C_i|\mathcal{O})$, where C_i is the i-th observation cluster. $P(b^{(T+1)} = b|C_i)$ can be estimated as introduced in Section 5.3, and $P(C_i|\mathcal{O})$ can be computed by $P(C_i|\mathcal{O}) \propto P(C_i) \prod_{j=1}^{T} \prod_{k,m} P(p_{j,k}|C_i)$

$P(r_{j,m}|C_i)P(z_{j,m}|C_i)$, where $p_{j,k}$, $r_{j,m}$, and $z_{j,m}$ denote the k-th ranking observation, m-th rating and topic observations in observation record $o_j \in \mathcal{O}$.

The second baseline **MPP** stands for Markov chain based Popularity Prediction, which is a sequential approach with first order Markov assumption. Specifically, given a T-length observation sequence $\mathcal{O} = \{o_1, \cdots, o_T\}$, we predict the popularity observation $b^{(T+1)}$ $(b = p, r, z)$ by the probability $P(b^{(T+1)} = b|\mathcal{O}) = P(b^{(t+1)} = b|o_T)$. We have the probability $P(b^{(t+1)} = b|o_T) \propto P(b^{(T+1)} = b) \prod_k P(p_{T,k}|b^{(T+1)} = b) \prod_m P(r_{T,m}|b^{(T+1)} = b)P(z_{T,m}|b^{(T+1)} = b)$, where probabilities $P(b^{(T+1)} = b)$ and $P(b'|b^{(T+1)} = b)$ $(b' = p, r, z)$ can be computed by the MLE method. Specifically, we have $P(b^{(T+1)} = b) = \frac{N_b}{\sum_{b'} N_{b'}}$, and $P(b'|b^{(T+1)} = b) = \frac{N^o_{b',b}}{N^o_b}$, where N_b is the appearance frequency of b in all observation records. N^o_b is the number of observation records in training data that contain b, and $N^o_{b',b}$ is the number of observation records in training data that contain b, and the last record as b'.

First, we compare the performance of ranking and rating prediction by each approach. Indeed, both ranking and rating are numerical observations, thus we expect

the prediction results should be close to the ground truth values of observations. Particularly, in our data set, there are one ranking observation and several rating observations in each observation record. Therefore, we can use the ranking observation and the average rating as ground truth values for evaluation. Specifically, we evaluate each approach by calculating the Root Mean Square Error (RMSE) with the predicted results and ground truth values for all test sequences. Take ranking as an example, we define $RSME = \sqrt{\frac{\sum_{\mathcal{O}_i}(p_i^* - b_i^{\Delta})^2}{N}}$, where $\mathcal{O}_i$ is the i-th test sequence with length T_i, $p_i^* = \arg\max_p P(p^{(T_i+1)} = p|\mathcal{O}_i)$, p_i^{Δ} is the ground truth ranking in the $(T_i + 1)$-st observation record, and N is the number of test sequences. Moreover, we can calculate the RMSE of rating prediction in a similar way. The smaller RMSE value, the better performance of ranking and rating prediction.

Second, we evaluate the performance of predicting topics by each approach. Different from ranking and rating, review topic is categorical observation. Therefore, we propose to exploit the popular metric Normalized Discounted Cumulative Gain (NDCG) for evaluation. Specifically, in the ground truth observation records, there are several review topics, thus we define the ground truth relevance of each unique topic z, i.e., $Rel(z)$, as its normalized appearance frequency in the record. Also, each approach can predicate a ranked list, i.e., Υ_{PR}, of topics for each test sequence $\mathcal{O}_i$ with respect to the posterior probability $P(z^{(T_i+1)} = z|\mathcal{O}_i)$. After that, we can calculate the discounted cumulative gain (DCG) of each approach by $DCG = \sum_{i=1}^{K_z} \frac{2^{Rel(z_i)}-1}{\log_2(1+i)}$, where $K_z = 20$ is the number of topics, z_i is the i-th topic in Υ_{PR}, $Rel(z_i)$ is the ground truth relevance. The NDCG is the DCG normalized by the IDCG, which is the DCG value of the ideal ranking list of the returned results and $NDCG = \frac{DCG}{IDCG}$.

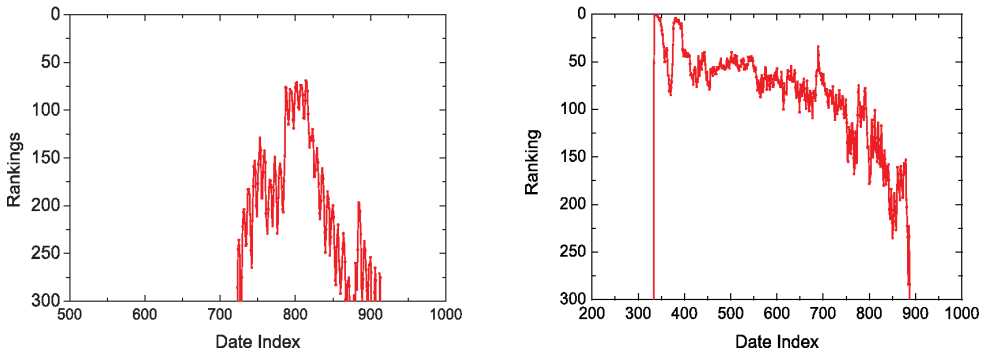


Figure 5.6. The demonstration of the ranking records of two different Apps.

Finally, we calculate the average NDCG for all test cases. Indeed, $NDCG$ indicates how well the rank order of topics is by each approach. The larger NDCG value, the better performance of topic prediction.

5.5.3 A Case Study of Ranking Fraud Detection

As introduced in Section 5.4, our PHMM model can be used for detecting ranking fraud for mobile Apps. Here, we study the performance of ranking fraud detection based on the prior knowledge from existing reports. Specifically, as reported by IBTimes cas, there are eight free Apps which might involve the ranking fraud. In this work, we use seven of them in our data set (*Tiny Pets*, *Social Girl*, *Fluff Friends*, *Crime City*, *VIP Poker*, *Sweet Shop*, *Top Girl*) for evaluation. Particularly, instead of using sliding widow to segment observation sequences, we directly calculate the anomaly score with respect to all popularity observations of each sequence. When we rank all Apps in our data set with respect their ranking anomaly scores, we find that all above seven suspicious Apps are ranked in top 5%, which indicates our PHMM model can find these suspicious Apps with high rankings. Furthermore, Figure 5.6 (a), (b) show the ranking records of the highest-ranked (i.e., most suspicious) Apps in

our data set and the seven suspicious Apps. From these figures, we can find that the Apps that contain several impulsive ranking patterns have high ranking positions. In contrast, the ranking behaviors of the normal Apps may be completely different. For example, Figure 5.6 (c), (d) show the ranking records of the lowest-ranked (i.e., most normal) App in our data set and a popular App "Angry Birds: Season-Free", both of which have the clear popularity trends. In fact, once a normal App is ranked high in the leaderboard, it often owns a lot of honest fans and may attract more and more users to download. Thus, the popularity will not vary dramatically in a short time.

5.6 Related Work

Generally speaking, the related works of this study can be grouped into two categories.

The first category is about the mobile App recommendation and other related services. For example, Yan *et al.* Yan and Chen [2011] developed a mobile App recommender system, named Appjoy, which is based on user's App usage records to build a preference matrix instead of using explicit user ratings. Also, to solve the sparsity problem of App usage records, Shi *et al.* Shi and Ali [2012] studied several recommendation models and proposed a content based collaborative filtering model, named Eigenapp, for recommending Apps in their Web site Getjar. Karatzoglou *et al.* Karatzoglou et al. [2012] proposed a novel context-aware collaborative filtering algorithm based on tensor factorization for mobile App recommendation, which named Djinn model. Indeed, detecting the rating and comment spam is also an important application of recommender systems. For example, Lim *et al.* Lim et al. [2010] have identified several representative behaviors of review spammers and model these be-

haviors to detect the spammers. Wu *et al.* Wu et al. [2012] have studied the problem of detecting hybrid shilling attacks on rating data based on the semi-supervised learning algorithm. Xie *et al.* Xie et al. [2012] have studied the problem of singleton review spam detection. Specifically, they solved this problem by detecting the co-anomaly patterns in multiple review comments based time series. Although most of the previous studies leveraged the popularity information in their applications, none of them can comprehensively model the popularity observations. To this end, in this work we proposed a PHMM model for popularity modeling of mobile Apps, which can be exploited for most of above applications.

The another category of related works is about the HMM models, which have been widely used in various research domains. For example, Rabiner has propose a comprehensive tutorial of HMM and its applications in speech recognition. Maiorana *et al.* Maiorana et al. [2010] have applied the HMM into biometrics with application of online signature recognition. Yamanishi *et al.* Yamanishi and Maruyama [2005] proposed to leverage HMM for network failure detection by estimating the anomaly sequences of system logs. Different with above works, in this work, we introduce a novel application, namely popularity modeling for mobile Apps, by extending the HMM model with multiple popularity observations. To our best knowledge, this is the first comprehensive study of modeling popularity for Apps.

5.7 Summary

In this work, we presented a sequential approach for modeling the popularity information of mobile Apps. Along this line, we first proposed a Popularity based HMM

(PHMM) model by learning the sequences of heterogeneous popularity observations from mobile Apps. Then, we introduced a bipartite based method to pre-cluster the popularity observations, which can efficiently learn the parameters and initial values of the PHMM model. A unique perspective of our approach is that it can capture the sequence dependence and the latent semantics of multiple popularity observations. Furthermore, we also demonstrated several novel App recommendation services enabled by the PHMM model, including trend based App recommendation, rating and comment spam detection, and ranking fraud detection. Finally, the extensive experiments on a real-world data set collected from the Apple App Store clearly showed the efficiency and effectiveness of our approach.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Major Results

This dissertation presents a suite of techniques for sequential pattern analysis. Particularly, we focus on identifying the right pattern granularity for both sequential pattern mining and modelling. First, in Chapter 2, we contend that the right pattern granularity for sequential pattern mining is hidden due to the curse of cardinality. To proactively reduce the cardinality based on the temporal content in the sequences, we develop the temporal skeletonization. We use graph-based algorithms in a novel way to analyze the correlation structures of the high-dimensional symbolic features. Second, in Chapter 3, for statistical modelling of sequential data, we show that the right data granularity is critical to improve both the interpretability and the goodness-of-fit of the learned models. Focused on the workflow modelling with the indoor location traces of the moving objects in hospitals, we identify the modelling granularity with the activity density based clustering algorithm. We also exploit the correlations between multiple types of medical devices to jointly estimate the workflow models. Third, in Chapter 4, we show another strategy to improve the modelling performance with complicated pattern granularity. The assumption is that, although the right granularity is unknown for the overall data, the local scenario can be much

clearer. Focused on identifying the abnormal location traces with missing events of the medical devices, we first use clustering algorithms to locate the 'hotspots' of the missing events, then use stochastic models to quantify the anomaly degree of the nearby location traces. Finally, in Chapter 5, we develop the PHMM method to analyze the product adoption in the digit markets. The goal is again to improve both the interpretability and the goodness-of-fit of the stochastic model. Motivated by the data characteristic, we introduce a bipartite based method to pre-cluster the raw observations and use the clustering solution to initialize the non-convex unsupervised HMM estimation and guide its convergence.

6.2 Future Research Directions

The general future research align with this dissertation is the *Temporal and Structural Analysis of Sequential Data*. As we have shown in earlier chapters, by introducing the powerful graph-based algorithms to the analysis of sequences and particularly the temporal correlations, we open up new possibilities to explore, quantify, and visualize the symbolic sequential data. In the future, it is worthy to generalize the temporal graph to accommodate the real-valued multivariate time series. The generalized temporal graph should be able to capture the correlations with practical temporal decays, and naturally substitute the conventional covariance analysis. In terms of application, the generalized temporal graph can be used for dynamic system monitoring, such as disease diagnosis with fMRI (functional magnetic resonance imaging) data and modelling the correlated trending of stock prices. In the literature, this is an active field where different formulations are being proposed to compute the tem-

poral correlations with better robustness and interpretability. It will be interesting to investigate the complementariness between our idea and the existing approaches.

As another direction to further the temporal and structural analysis of sequential data, it is meaningful to exploit the personalized temporal graphs. For instance, a temporal graph can be constructed with the sequential observations for each individual. Then a set of graphs can be mined and modelled for downstream analytical tasks. In this way, the personalized temporal graph can be deemed a new knowledge representation of the sequential data. One challenge along this line is how to learn informative features based on the graph-based sequence representation.

Moreover, on sequential pattern modelling, we have been improving the model interpretability and goodness-of-fit by constructing semantically meaningful states for better model granularity and pre-clustering the temporally correlated observations for model initialization. In the future, to cope with different applications, it is interesting to generalize these works with an unified probabilistic framework which simultaneously identifies the pattern granularity levels and estimates the model parameters.

REFERENCE

http://www.ibtimes.com/apple-threatens-crackdown-biggest-app-store-ranking-fraud-406764.

Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

Rakesh Agrawal, Dimitrios Gunopulos, and Frank Leymann. Mining process models from workflow logs. *Advances in Database TechnologyEDBT'98*, pages 467–483, 1998.

Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435. ACM, 2002.

A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM, 2003.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Lantent dirichlet allocation. In *Journal of Machine Learning Research*, pages 993–1022, 2003.

Matthias Böhmer, Lyubomir Ganev, and Antonio Krüger. Appfunnel: a framework for usage-centric evaluation of recommender systems that suggest mobile applications. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, IUI '13, pages 267–276, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1965-2. doi: 10.1145/2449396.2449431. URL `http://doi.acm.org/10.1145/2449396.2449431`.

Byron Boots and Geoffrey J Gordon. An online spectral learning algorithm for partially observable nonlinear dynamical systems. In *AAAI*, 2011.

Yingyi Bu, Lei Chen, Ada Wai-Chee Fu, and Dawei Liu. Efficient anomaly monitoring over moving object trajectory streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 159–168, Paris,France, 2009.

Alexandre Bureau, Stephen Shiboski, and James P Hughes. Applications of continuous time hidden markov models to the study of misclassified disease outcomes. *Statistics in Medicine*, 22(3):441–462, 2003.

Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 875–883, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401995. URL `http://doi.acm.org/10.1145/1401890.1401995`.

Huanhuan Cao, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 191–200, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526736. URL `http://doi.acm.org/10.1145/1526709.1526736`.

Huanhuan Cao, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proceedings of the 18th international conference on World wide web*, pages 191–200. ACM, 2009b.

X. Chai and Q. Yang. Multiple-goal recognition from low-level signals. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Lei Chen, M. Tamer Ozsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pages 491 – 502, IL, USA, 2005.

Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, 1995.

Don Coppersmith, Lisa Fleischer, and Atri Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 776–782. ACM, 2006.

Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.

M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. AAAI Press, 1996.

Aphrodite Galata, Neil Johnson, and David Hogg. Learning variable-length markov models of behavior. *Computer Vision and Image Understanding*, 81(3): 398–413, 2001.

Y. Ge, H. Xiong, C. Liu, and Z.H. Zhou. A taxi driving fraud detection system. In *Proceedings of the 11th IEEE International Conference on Data Mining*, pages 181–190. IEEE, 2011a.

Yong Ge, Hui Xiong, Zhi-hua Zhou, Hasan Ozdemir, Jannite Yu, and Kuo Chu Lee. Top-eye: Top-k evolving trajectory outlier detection. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1733–1736. ACM, 2010.

Yong Ge, Chuanren Liu, Hui Xiong, and Jian Chen. A taxi business intelligence system. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 735–738. ACM, 2011b.

Yong Ge, Hui Xiong, Chuanren Liu, and Zhi hua Zhou. A taxi driving fraud detection system. In *Proceedings of the IEEE International Conference on Data Mining*, pages 181–190, 2011c.

Fosca Giannotti, Mirco Nanni, and Dino Pedreschi. Efficient mining of temporally annotated sequences. In *In Proc. SDM06*. Citeseer, 2006.

Fosca Giannotti, Mirco Nanni, Dino Pedreschi, and Fabio Pinelli. Trajectory pattern mining. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339, 2007a.

Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330 – 339, California, USA, 2007b.

Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339. ACM, 2007c.

Aristides Gionis, Heikki Mannila, Kai Puolamäki, and Antti Ukkonen. Algorithms for discovering bucket orders from data. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 561–566. ACM, 2006.

Gianluigi Greco, Antonella Guzzo, Giuseppe Manco, and Domenico Sacca. Mining and reasoning on workflows. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):519–534, 2005.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *In Proc. of National Academy of Science of the USA*, pages 5228–5235, 2004.

Jiawei Han and Yongjian Fu. Mining multiple-level association rules in large databases. *Knowledge and Data Engineering, IEEE Transactions on*, 11(5):798–805, 1999.

Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and MC Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *proceedings of the 17th international conference on data engineering*, pages 215–224, 2001.

Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15 (1):55–86, 2007.

G. Heinrich. Paramter stimaion for text analysis. *Technical report, University of Lipzig*, 2008.

Derek Hao Hu and Qiang Yang. Cigar: concurrent and interleaving goal and activity recognition. In *Proceedings of the 23rd national conference on Artificial intelligence*, pages 1363–1368, 2008.

Hoyoung Jeung, Man Lung Liu, Xiaofang Zhou, Christian S. Jensen, and Heng Tao Shen. Discovery of convoys in trajectory databases. In *Proceedings of the VLDB Endowment*, pages 1068–1080, 2008.

Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. Climbing the app wall: enabling mobile app discovery through context-aware recommendations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 2527–2530, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398683. URL `http://doi.acm.org/10.1145/2396761.2398683`.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

J-G Lee, Jiawei Han, Xiaolei Li, and Hong Cheng. Mining discriminative patterns for classifying trajectories on road networks. *Knowledge and Data Engineering, IEEE Transactions on*, 23(5):713–726, 2011.

Jae-Gil Lee, Jiawei Han, and Whang Kyu-Young. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pages 593–604, Beijing, China, 2007.

Jae-Gil Lee, Jiawei Han, and Xiaolei Li. Trajectory outlier detection: A partition-and-detect framework. In *Proceedings of the 24th International Conference on Data Engineering*, pages 140–149, Cancun, Mexico, 2008.

Xiaolei Li, Zhenhui Li, , Jiawei Han, and Jae-Gil Lee. Temporal outlier detection in vehicle traffic data. In *Proceedings of the 25th International Conference on Data Engineering*, pages 1319–1322, Shanghai, China, 2009.

Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1099–1108. ACM, 2010.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 939–948, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871557. URL `http://doi.acm.org/10.1145/1871437.1871557`.

Chuanren Liu, Hui Xiong, Yong Ge, Wei Geng, and Matt Perkins. A stochastic model for context-aware anomaly detection in indoor location traces. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 449–458. IEEE, 2012.

Chuanren Liu, Yong Ge, Hui Xiong, Keli Xiao, Wei Geng, and Matt Perkins. Proactive workflow modeling by stochastic processes with application to healthcare operation and management. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1593–1602. ACM, 2014a.

Chuanren Liu, Kai Zhang, Hui Xiong, Geoff Jiang, and Qiang Yang. Temporal skeletonization on sequential data: patterns, categorization, and visualization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1336–1345. ACM, 2014b.

Siyuan Liu, Yunhuai Liu, Lionel M. Ni, Jianping Fan, and Minglu Li. Towards mobility-based clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 919–928, 2010a.

Siyuan Liu, Yunhuai Liu, Lionel M. Ni, Jianping Fan, and Minglu Li. Towards mobility-based clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 919–928. ACM, 2010b.

David Lo, Hong Cheng, et al. Mining closed discriminative dyadic sequential patterns. In *Proceedings of the 14th International Conference on Extending Database Technology*, pages 21–32. ACM, 2011.

E. Maiorana, P. Campisi, J. Fierrez, J. Ortega-Garcia, and A. Neri. Cancelable templates for sequence-based biometrics with application to on-line sig-

nature recognition. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(3):525–538, 2010. ISSN 1083-4427. doi: 10.1109/TSMCA.2010.2041653.

N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D.W. Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 236–245. ACM, 2004.

Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2: 849–856, 2002.

Lane MD Owsley, Les E Atlas, and Gary D Bernard. Automatic clustering of vector time-series for manufacturing machine monitoring. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 4, pages 3393–3396. IEEE, 1997.

Jian Pei, Guozhu Dong, Wei Zou, and Jiawei Han. On computing condensed frequent pattern bases. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 378–385. IEEE, 2002.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1424–1440, 2004.

T.J. Perkins. Maximum likelihood trajectories for continuous-time markov chains. *Advances in Neural Information Processing Systems*, 22:1437–1445, 2009.

L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. ISSN 0018-9219. doi: 10.1109/5.18626.

J. Rissanen and G.G. Langdon. Arithmetic coding. *IBM Journal of research and development*, 23(2):149–162, 1979.

R.R.Sillito and R.B.Fisher. Semi-supervised learning for anomalous trajectory detection. In *Proceedings of the 19th British Machine Vision Conference*, pages 1035–1044, Leeds, UK, 2008.

Frans Schalekamp and Anke van Zuylen. Rank aggregation: Together we're strong. In *ALENEX*, pages 38–51. SIAM, 2009.

Kent Shi and Kamal Ali. Getjar mobile application recommendations with very sparse datasets. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 204–212, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339563. URL `http://doi.acm.org/10.1145/2339530.2339563`.

Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. In *VLDB*, volume 95, pages 407–419, 1995.

Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, volume 1057 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 1996.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

Wil Van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *Knowledge and Data Engineering, IEEE Transactions on*, 16(9):1128–1142, 2004.

A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2): 260–269, 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1054010.

Michail Vlachos, Dimitrios Gunopoulos, and George Kollios. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering*, pages 673–684, CA, US, 2002.

X. Wang, K.T. Ma, Gee-Wah Ng, and W. Eric L. Grimson. Trajectory analysis and semantic region modeling using nonparametric bayesian model. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Patter Recognition (CVPR)*, pages 1–8, AK, USA, 2008.

Zhiang Wu, Junjie Wu, Jie Cao, and Dacheng Tao. Hysad: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 985–993, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339684. URL http://doi.acm.org/10.1145/2339530.2339684.

Sihong Xie, Guan Wang, Shuyang Lin, and Philip S. Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 823–831, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339662. URL http://doi.acm.org/10.1145/2339530.2339662.

Dong Xin, Jiawei Han, Xifeng Yan, and Hong Cheng. Mining compressed frequent-pattern sets. In *Proceedings of the 31st international conference on Very large data bases*, pages 709–720. VLDB Endowment, 2005.

Kenji Yamanishi and Yuko Maruyama. Dynamic syslog mining for network failure monitoring. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 499–508, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X. doi: 10.1145/1081870.1081927. URL http://doi.acm.org/10.1145/1081870.1081927.

Bo Yan and Guanling Chen. Appjoy: personalized mobile application discovery. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys '11, pages 113–126, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0643-0. doi: 10.1145/1999995.2000007. URL http://doi.acm.org/10.1145/1999995.2000007.

Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):40–51, 2007.

Jiong Yang and Meng Hu. Trajpattern: Mining sequential patterns from imprecise trajectories of mobile objects. In *EDBT'06*, pages 664–681, 2006.

J. Yin, D. Shen, Q. Yang, and Z.N. Li. Activity recognition through goal-based segmentation. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 28. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Junfu Yin, Zhigang Zheng, and Longbing Cao. Uspan: an efficient algorithm for mining high utility sequential patterns. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 660–668. ACM, 2012.

Mohammed J Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.

Y. Zheng, L. Zhang, X. Xie, and W.Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009a.

Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the International Conference on World Wide Web*, pages 791–800, 2009b.

Hengshu Zhu, Huanhuan Cao, Enhong Chen, Hui Xiong, and Jilei Tian. Exploiting enriched contextual information for mobile app classification. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1617–1621. ACM, 2012.

Hengshu Zhu, Chuanren Liu, Yong Ge, Hui Xiong, and Eenhong Chen. Popularity modeling for mobile apps: A sequential approach. *IEEE transactions on cybernetics*, 2014.