DISCOVERING SPATIO-TEMPORAL PATTERNS IN MULTIVARIATE SPATIAL TIME SERIES

WU PAK KIT

PhD

The Hong Kong Polytechnic University

2019

This page is intentionally left blank.

The Hong Kong Polytechnic University

Department of Computing

Discovering Spatio-Temporal Patterns in Multivariate Spatial Time Series

WU Pak Kit

A thesis submitted in partial fulfilment of the requirements

for the degree of

Doctor of Philosophy

August 2018

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________ (Signed)

WU Pak Kit (Name of student)

ABSTRACT

A multivariate spatial time series (MSTS) consists of a collection of values by a set of geographical coordinates accompanied by a set of multivariate time series (MTS). An MTS is composed of a number of temporally interrelated variables monitored over a period of time at successive time instants spaced at uniform time intervals. MTS data are generated massively due to recent developments in sensor and satellite technologies, medical measurements, climate informatics, and bioinformatics. These large-scale data encode important information about complex relations among individual time series. Many of these MTS are spatio-temporal by nature in which they are collected together with spatial location information such as latitude and longitude. For example, climate data are from sensors located in different regions, each of which collects periodic readings of variables such as humidity, wind speed, temperature, and rainfall intensity. A computational technique that is able to discover interesting patterns in MSTS data can lead to many applications in diverse areas of research and be helpful to society as well as to the economy. MSTS can be represented as a set of MTSs each of which is associated with a spatial location. Conventional time series analysis methods which consider only the time domain are often adopted to analyze MTS, but the spatial and temporal relationships associated with the individual time series in MSTS are usually ignored, or treated separately, during the pattern discovery process. For this reason, new effective techniques are required. In this thesis, we proposed some such techniques, in particular, that can be used to address the problems of identifying interesting patterns in MSTS and the classification and clustering of them.

One of the classical examples of MTS is spatial trajectory data with *x* coordinate and *y* coordinate forming the different components of the MTS. In many cases, such data is also spatio-temporal as it may be associated with many spatio-temporal parameters such as velocity and direction etc. Mining spatial trajectories can have many applications in a variety of research areas. For example, in traffic data, finding patterns of driving behavior of moving objects can provide insight into many real applications such as auto insurance and vehicle safety checks. In this regard, we propose in this thesis a technique that can discover association patterns from the feature space characterizing the spatial trajectories. These discovered association patterns, treated as the driving behavior on the road, could be used for the classification of drivers. A classification algorithm has been developed based on the proposed technique to consider the variable length of multiple spatial trajectories of each driver to determine the class membership that exists between association patterns of these trajectories and the class. Integrated with an information theoretic measure, this classifier is able to predict and identify uniquely a driver based on driving patterns of unlabeled trajectories that are for or against a certain class membership. For performance evaluation, we have used it to solve problems in driver classification using their spatial trajectory data.

According to empirical studies on spatial data analysis, mining of MSTS should consider the spatial nature of the objects to be analyzed, their characteristics of the feature space and the uncertainty between the spatial units and their complex features. The proposed technique, to discover association patterns in MSTS, should consider both spatial and temporal information. This proposed technique not only can uncover the temporal and spatial association relationships of MSTS

but also can tackle supervised and unsupervised learning tasks. This technique incorporates an initial MTS pattern mining algorithm to detect temporal association relationships from frequent patterns in a set of MTSs for each location. We have developed an algorithm to detect co-occurrence of the discovered temporal patterns across locations by mining a transformed spatio-temporal pattern matrix (STPM) that characterizes the feature space to form spatio-temporal patterns. That is to say, if the frequency of co-occurrence of the respective temporal patterns in different spatial units is significantly higher, the co-occurrence of the temporal patterns across locations is the spatial association patterns of interest. To determine if the frequency of their co-occurrences is significantly higher, we apply a statistical significance test to measure how significantly the observed frequency of the co-occurrences deviates from its expected frequency. Furthermore, we effectively integrate this spatio-temporal pattern-mining algorithm for classification and clustering by an information theoretic measure. If the set of MSTS is labeled, the discovered patterns can be weighted to support or against a certain class membership for the construction of a classifier. If the set of MSTS is unlabeled, the discovered patterns in one location are compared against those discovered in the others so that, by taking the spatial contiguity between locations into consideration, MSTS that have similar discovered patterns and are closer to each other are grouped together into the same cluster. To evaluate the performance of the algorithms, we have tested them on both synthetic and real-world data sets. We have also applied them to tackle several practical problems in some case studies. Both experimental results and findings from practical case studies show the proposed techniques to be promising for MSTS analysis.

This page is intentionally left blank.

PUBLICATIONS ARISING FROM THE THESIS

Wu, G. P., & Chan, K. C. (2018, March). *Clustering driving trip trajectory data based on pattern discovery techniques*. Paper presented at 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China. doi:10.1109/ICBDA.2018.8367726

Wu, G. P., & Chan, K. C. (2018). *Discovery of spatio-temporal patterns in multivariate spatial time series*. Manuscript submitted for publication.

Wu, G. P., & Chan, K. C. (2018, March). *Mining spatio-temporal patterns in multivariate spatial time series*. Paper presented at 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China. doi:10.1109/ICBDA.2018.8367650

Wu, G. P., & Chan, K. C. (2017, December). *Privacy-preserving trajectory classification of driving trip data based on pattern discovery techniques*. Paper presented at 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA. doi:10.1109/BigData.2017.8258383

Wu, G. P., Chen, Y. C., Zhu, W. Y., & Chan, K. C. (2013, December). *An Intelligent System for Effective Mobile Application Advertising*. Paper presented at 2013 Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taiwan.

This page is intentionally left blank.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Keith C. C. Chan for the continuous support of my PhD study and related research, for his patience, encouragement and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

I would also like to thank my committee members, Prof. Yiu Ming Cheung, Department of Computer Science, Hong Kong Baptist University, Prof. Wei Ding, Department of Computer Science, University of Massachusetts Boston and Dr. Maggie Wenjie Li, Department of Computing, The Hong Kong Polytechnic University, for their insightful comments and encouragement but also for the hard questions benefiting me to widen my research from various perspectives.

I thank my fellow labmates for stimulating the research discussions and for all the fun we have had in the last six years. Also, I thank my friends for accepting nothing less than excellence from me. In particular, I am grateful to Prof. Andrew K. C. Wong, Department of Systems Design Engineering, University of Waterloo, Canada for enlightening me at first glance of research.

I deeply thank my parents for unconditional trust, timely encouragement, and endless patience. It was their love that raised me up again when I got weary. My sister, parents in law, sister in law and brother in law have also been generous with their love and affection. Last but not the least, I thank with love to my wife and little son, being my best friend, great companion, loved, supported, encouraged, entertained, and helped me get through this tough period in the most positive way. There is no way I would have been able to finish this without them.

This page is intentionally left blank.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A_j^p	The *p*th value of *j*th attribute
A_j	The *j*th attribute
C	Set of classes
$Cluster$	Set of clusters
C_x	Cluster label
C_y	Assigned cluster label
$\|C\|$	Number of class labels
c_i	The *i*th class label
cnt_{C^i}	Count of records in *i*th cluster
cnt_{C_x,C_y}	Count of records with cluster label C_x in assigned cluster C_y
D	Set of trips in a moving object database
d_{P_n}	Adjusted residual of P_n
d_P	Adjusted residual of P
$\mathrm{d}'(i)$	Aggregate pattern statistical significance of region i
$\widetilde{d'(\iota)}$	Normalized aggregate pattern statistical significance of region i
E	Set of attributed hyperedges
E_P	Expected occurrences of spatial or temporal pattern $P \in \{TP, SP\}$
e_{P_n}	Expected occurrences of pattern P_n

e_{pk} .... Expected occurrences when $A_j = A_j^p$ and $A_{j'} = A_{j'}^k$

ε .... Alphabet set ε for SAX

FP .... Frequent pattern

$G(x, y)$ .... Function to retrieve the region for a given longitude x and latitude y

g .... Geographic coordinate containing longitude x and latitude y

$H(V, E)$ .... Attributed hypergraph

$I(\,)$ .... Mutual information

L .... Locations in the study area

$|L|$ .... Number of locations in the study area

$|L_P|$..................... Observed frequency of occurrences of pattern $L_P, P \in \{TP, SP\}$

L_C .... A new occurrence list created for each cluster of patterns during the operation of temporal association pattern discovery

LS .... Level of SP

LT .... Level of TP

L_{SP} .... Set of occurrences of SP in all MTS in different regions

L_{TP} .... Set of occurrences of TP in the MTS

$MSTS$ .... Multivariate spatial time series

MTS .... Multivariate time series

max_{LS} .... Maximum level of spatial association SP

max_{LT} .... Maximum level of temporal association TP

min_o .... Minimum number of occurrences for frequent patterns

o_{P_n} .... Observed occurrences of pattern P_n

o_{pk} .... Total number of occurrences when $A_j = A_j^p$ and $A_{j'} = A_{j'}^k$

$order_{max}$ .... User input parameter for maximum order of pattern search operation

P .... A pattern P is a short sequence of consecutive characters

PC_n .... Pattern candidate set

P^{LT} .... Last block of TP

P_n .... Pattern of order n

ρ^{LT} .... Occurrence of P^{LT}

r_j .... The j^{th} region label

S .... Sequence of characters

sig .... Significance threshold value

$Sim(X_i, X_j)$ .... Similarity measure of objects X_i and X_j

SP .... Spatial association pattern

ST .... Statistical significance test

$STPM$ .... Spatio-temporal pattern matrix

T .... Trip as known as trajectory

TP .... Temporal association pattern

TS .... Time series

t_d .... Maximum time delay for temporal association TP

V .... Set of attributed vertices

V_P .... Maximum likelihood estimate of the variance of $\frac{|L_P|-E_P}{\sqrt{E_P}}, P \in \{TP, SP\}$

V_{pk} .... Maximum likelihood estimate of the variance of $\frac{o_{pk}-e_{pk}}{\sqrt{e_{pk}}}$

v_{P_n} .... Maximum likelihood estimate of the variance of z_{P_n}

W .... Adjacency matrix assigning weights to neighbors of regions

W$_{i,j}$ .... Indication of region r_i and r_j sharing a common border or not

$W^i(T|p^i)$ .... Weight of evidence measure provided by a pattern p^i for or against the classification of a trajectory T into class c_{i}

$W^i(X|p^i)$ .... Weight of evidence measure provided by a pattern p^i for or against the classification of an object X into class c_{i}

$W^i(T_u)$ .... Total weight of evidence provided by each of individual patterns is a measure for T_u to be classified into class c$_{\mathrm{i}}$

$W^i(X_u)$ .... Total weight of evidence provided by each of individual patterns is a measure for X_u to be classified into class c$_{\mathrm{i}}$

X .... Object as known as row vector

z_{P_n} .... Standardized residual of P_n

1 INTRODUCTION

Multivariate spatial time series (**MSTS**) data consist of a collection of values by a set of geographical coordinates accompanied by a set of multivariate time series (**MTS**). An MTS is composed of a number of temporally interrelated variables monitored over a period of time at successive time instants spaced at uniform time intervals. MTS data are generated massively due to recent developments in sensor and satellite technologies, medical measurements, climate informatics, and bioinformatics. These large-scale data encode important information about complex relations among individual time series. Many recent works on multivariate time series (MTS) pattern discovery focus mainly on extracting temporal association patterns and features (Zhuang, Li & Wong, 2014; Zhou & Chan, 2015; MacEachren, Wachowicz, Edsall, Haug & Masters, 1997). Many of these MTS are spatio-temporal by nature in which they are collected together with spatial location information such as latitude and longitude. For example, climate data are from sensors located in different regions, each of which collects periodic readings of variables such as humidity, wind speed, temperature, and rainfall intensity. A computational technique that is able to discover interesting patterns in MSTS data can lead to many applications in diverse areas of research and be helpful to society as well as to the economy. MSTS can be represented as a set of MTSs each of which is associated with a spatial location. Conventional time series analysis methods which consider only the time domain are often adopted to analyze MTS (Rosén & Yuan, 2001; Yang & Shahabi, 2004; Singhal & Seborg, 2005; Yoon, Yang & Shahabi, 2005; Owsley, Atlas & Bernard, 1997; Coppi, D'Urso & Giordani, 2010; Zuur, Fryer, Jolliffe, Dekker & Beukema, 2003) but the spatial and temporal relationships associated with the individual time series in MSTS are usually ignored, or treated separately, during the pattern discovery process. For this reason, new effective techniques are required. In this thesis, we will propose some such techniques, in particular, that can be used to address the problems of identifying interesting patterns in MSTS and the classification and clustering of them.

One of the classical examples of MTS is spatial trajectory data with *x* coordinate and *y* coordinate forming the different components of the MTS. In many cases, such data is also spatio-temporal as it may be associated with many spatio-temporal parameters such as velocity and direction etc. Mining spatial trajectories can have many applications in a variety of research areas (Zheng, 2015). For example, in traffic data, finding patterns of driving behavior of moving objects can provide insight into many real applications such as auto insurance and vehicle safety checks. In this regard, we start with an attempt to propose in this thesis a technique that can discover association patterns from the feature space characterizing the spatial trajectories. These discovered association patterns, treated as the driving behavior on the road, should be able for the classification of drivers.

According to empirical studies on spatial data analysis (Coppi, D'Urso & Giordani, 2010), mining of MSTS should consider the spatial nature of the objects to be analyzed, their characteristics of the feature space and the uncertainty between the spatial units and their complex features. In this regard, the proposed technique should discover association patterns in MSTS using both spatial and temporal information. This proposed technique not only should uncover the temporal and spatial association relationships of MSTS, but also should tackle supervised and unsupervised learning tasks. We attempt to combine both temporal and spatial information in the MSTS data for the analysis and address the challenges that exist in the literature. Hence, this proposed technique will incorporate an initial MTS pattern-mining algorithm to detect temporal association relationships from frequent patterns in a set of MTSs for each location. Then, to detect co-occurrence of the discovered temporal patterns across locations, we will mine a transformed spatio-temporal pattern matrix (**STPM**) that characterizes the feature space to form spatio-temporal patterns. Furthermore, we will investigate how this spatio-temporal pattern-mining algorithm can be integrated for classification and clustering by an information theoretic measure.

The rest of this chapter is organized as follows. Section 1.1 introduces the motivations behind the introduction of pattern-mining problems in spatial trajectory data and multivariate spatial time series data. Section 1.2 introduces the objectives of this thesis that will lead to the research and development of the proposed approaches to tackle the spatio-temporal pattern-mining problems by the application of pattern discovery techniques. Section 1.3 presents the organization of this thesis.

1.1 Motivations

Spatio-temporal data come from a database that manages both space and time information. This database captures the spatial and temporal aspects of data. Many real-world applications contain and generate a vast amount of digital spatio-temporal information constantly so there is a great need to reveal new insights which previously remain hidden from the data in spatio-temporal nature such that useful information could be well extracted, effectively structured and further arranged for analysis including but not limited to clustering, classification, visualization, and interpretation. A number of approaches have been developed to tackle pattern-mining problems in this area of research. Most of them, such as PCA-based approach (Rosén & Yuan, 2001; Singhal & Seborg, 2005; Yoon, Yang & Shahabi, 2005), SVD-based approach (Yang & Shahabi, 2004), HMM-based approach (Owsley, Atlas & Bernard, 1997), Fuzzy *c*-means based approach (Coppi, D'Urso & Giordani, 2010), EM-based approach (Zuur, Fryer, Jolliffe, Dekker & Beukema, 2003), may find patterns between different multivariate time series data sets or within a single data set, focusing on time domain. However, for a set of multivariate spatial time series data, it may be interested in both finding temporal patterns within a multivariate time series and finding those across spaces, focusing on time as well as space domain. Moreover, conducting further correlation analysis over such spatio-temporal patterns might be able to unveil more useful knowledge.

The aforementioned methods have been used to obtain useful analytical knowledge, to some extent, to build models for classification and prediction tasks effectively but, however, it still poses some challenges that motivate us to develop the methodology in this thesis to tackle them as follows.

First, classical spatial analysis studies entities using their topological, geometric or geographic properties. Most algorithms performing the task of discovering relationship such as autocorrelation from spatial units emphasizes more on some predefined topological properties while analysis on both spatial and temporal information associated with multivariate spatial time series are not many. As such, important temporal dependence across multiple spatial units that might be meaningful cannot be discovered.

Second, traditionally, analytical techniques favor the spatial definition of objects as points. Some important attribute information, such as discriminative features hidden in the spatial trajectories and related temporal associations hidden in multiple spatial time series, may not be fully utilized for many pattern mining algorithms. Therefore, some interesting spatio-temporal patterns that possess significant co-occurrences may not be identified effectively.

Third, conventional approaches capture spatial and temporal dependency to provide information on spatio-temporal relationships in variable level. For example, the method proposed by Shumway (2014) detects clusters by segmenting a distance matrix that measures the pairwise distance between two sets of multivariate time series data. The entries in the distance matrix are obtained by comparing the binary difference of two distance measures. The strength of attribute values that might determine the class and cluster membership may not be fully utilized by this kind of methods. Without attribute-value level information, this may degrade the quality of the classification and clustering model.

1.2 Objectives

The objectives of this thesis are motivated by the aforementioned practical needs derived from the real-world application and are particularly listed as follows.

I. To discover statistically significant association patterns from i) a spatial trajectory database and ii) a multivariate spatial time series database: For a large spatial trajectory database, different trajectories may contain common and/or different features. These features can be formed association patterns. For a large multivariate spatial time series database, different time series in different spatial locations may also contain temporal and spatial dependencies. These spatio-temporal dependencies can also be formed association patterns. Each pattern of these types could represent certain characteristics of real-world events. Whether spatial trajectories or multivariate time series are labeled, or unlabeled, statistically significant association patterns should be discovered first in order for further pattern analysis.

II. To transform i) a spatial trajectory database and ii) a multivariate spatial time series database based on the discovered patterns into a relational database for further analysis: once the spatio-temporal features are extracted and discretized, the original i) spatial trajectory database and ii) multivariate spatial time series database will be processed based on the extracted characteristics to construct the transformed database similar to a relational database with discrete attributes.

III. To apply pattern discovery approach on a transformed database: Once i) the spatial trajectory data and ii) the multivariate spatial time series data from the spatio-temporal databases are transformed into discrete interval events, an effective pattern discovery method on categorical data could be applied to discover significant association patterns. The patterns discovered in the general form of a subset of discrete data attributes will then become explicit and will be available for the cluster and classification analysis.

1.3 Organization of this Thesis

This thesis interpolates material from four papers published by the author (Wu, Chen, Zhu & Chan, 2013; Wu & Chan, 2017, 2018a, 2018c) and one paper submitted to the journal (Wu & Chan, 2018b). Some material from each of these papers has also been incorporated into this introductory Chapter and Chapter 2. Meanwhile, Chapter 3 is based on the references of Wu and Chan (2017, 2018a, 2018b, 2018c). Chapter 4 uses material from references of Wu and Chan (2017, 2018a). Finally, Chapter 5 is based on the reference of Wu and Chan (2018b, 2018c). The thesis consists of six chapters and is organized as follows.

Chapter 1 introduces the motivation, objective and organization of this work.

Chapter 2 introduces the related knowledge, including the background to spatial trajectory data, trajectory classification and clustering, multivariate spatial time series and pattern discovery through a literature review.

Chapter 3 presents the problem definitions of mining patterns in spatial trajectory data and multivariate spatial time series data. The overview of the proposed mining approach is also given. This approach is composed of a collection of techniques, including a trajectory classification and clustering algorithm and an unsupervised pattern discovery algorithm for multivariate spatial time series data.

Chapter 4 proposes a new approach to classify and cluster spatial trajectory data based on the discovered association pattern. The proposed approach begins with data transformation based on the extracted features to convert the original spatial trajectory data to a transformed feature matrix. Instead of mining directly from the original spatial trajectory data, we mine association patterns from the transformed feature matrix. From the patterns discovered in the transformed feature matrix, these patterns are further used for training a classifier for classification and prediction if the original spatial trajectory data is labeled or for clustering if the data is unlabeled. To evaluate the effectiveness, the proposed approach is first applied on a synthetic data set and then on a number of real-world data sets including GPS tracks data set of physical exercises, a human location history data set and a driver telematics data set. The experimental results show that the proposed approach is effective and efficient in achieving a good accuracy in the prediction of the class labels of the spatial trajectory data based on the transformed set of attributes and can produce meaningful clustering results.

Chapter 5 defines the problem of supervised and unsupervised pattern discovery for multivariate spatial time series data and introduces a methodology for solving it. The proposed method, utilizing a statistical significance measure to detect associations and optimizing some information measures, such as mutual information and weight of evidence, between attributes of the transformed data matrix, groups spatial locations into groups/clusters. By applying the proposed algorithm to the multivariate spatial time series database, meaningful patterns that capture associations in a single and/or multiple time series across spaces are discovered. The matrix representation for the transformed MSTS stores important spatio-temporal pattern information to reveal the statistical significance, available for further clustering and classification. To evaluate the performance, we applied these proposed techniques and algorithms on a synthetic data set and several real-world data sets. The experimental results of the data mining tasks prove that they are capable of revealing interesting spatio-temporal patterns and building very accurate and insightful classification and clustering models.

Chapter 6 concludes the thesis, which summarizes its contributions, and proposes further work.

This page is intentionally left blank.

2 BACKGROUND AND RELATED WORK

Spatio-temporal data are generated massively due to advances in location-based service and mobile computing (Zheng, 2015). To discover patterns in spatio-temporal data, several algorithms have been proposed. In this thesis, we are particularly interested to tackle the data mining problems associated with spatial trajectory data and multivariate spatial time series data. In the literature, some problems and computational algorithms are related to the research interests in this thesis. These algorithms can be categorized according to the specific problems and properties that are considered, the specific techniques that are applied, and the specific applications of the algorithms. In this section, the state-of-the-art algorithms related to problems of discovering patterns in spatial trajectory data and multivariate spatial time series data are introduced sequentially.

2.1 Overview of the Era of Big Spatio-Temporal Data

Applying big data methods on spatio-temporal data is a relatively new topic, and a growing number of related applications have proved the potential application value of the huge data sets provided by millions of operating devices including smartphones, medical devices, telematics devices, space telescopes, environment sensors, etc. The emerging spatio-temporal data, as well as their important value for a number of fields (e.g. weather forecast, epidemic analysis, mobility analysis, social media analysis, etc.), have motivated a lot of researchers to develop powerful and scalable systems for processing and analyzing them. Nevertheless, due to its nature, spatio-temporal data sets are always very large and hard to analyze, making it a challenge to develop an efficient method for representing and mining general spatio-temporal data.

Vehicle telematics data is one of the important sources of spatial trajectory data. In the field of vehicle or automobile insurance, telematics data mining has been gradually showing its value. It greatly enhances the idea and applicability of usage-based insurance (UBI), which customizes the insurance plan for individual drivers according to driving features. The way telematics data are applied is that the data returned by a telematics device on the driver’s vehicle get analyzed through some analytical models. These models will either analyze the locations and distance covered by the vehicle or decide the driver’s driving style by studying the data, conclude with a driving pattern for the specific driver, and generate a customized insurance plan according to the property of the discovered patterns. This innovative approach to insurance premium pricing is getting more and more welcomed by most of the car insurers, making it the most widely realized application of big data on telematics devices. Baecke and Bocca

(2017) in their recent study prove the value of telematics-based data in the risk selection process of an insurance company. They compared the performance of three models in this context: a logistic regression, random forests and artificial neural networks model. Their research illustrates the importance of industry knowledge in the variable creation process.

Applications also include the topic of crime control and reduction. As technology advances, the even greater focus has been placed on the clear identification of crime hotspots together with the strategy of crime reduction or detection so as to combat organized crime, as a result of the growing role of intelligence-led policing. For a number of places, crime hotspots are getting critical to policing strategy, as they allow police force to focus on the areas of highest priority. Crime hotspot identification and analysis could greatly benefit from mining criminal data, which are represented as multivariate spatial time series in this thesis and by nature are spatio-temporal, resulting in lifting the efficiency of the police department (Ratcliffe, 2014).

Movement study and prediction are the new perspectives one could benefit from mining spatial trajectory data coming from medical devices, smartphones, and accessories (wristbands, watches etc.) or tracker system installed on wildlife. The development of wearable smart devices has become a hot topic. By making use of the collected data, the application concerns movement study and prediction, enhancing the research on many medical and zoology topics. For example, researchers have claimed that mining data from smart wearable devices could boost the study of human motor patterns (Bonato, Mork, Sherrill & Westgaard, 2003). In recent years, Liew, Wah, Shuja and Daghighi (2015) have surveyed some emerging application areas for personal data mining using smartphones and wearable devices with an extensive review on recent literature and a detailed taxonomy in terms of data generation, design choices, application models and algorithms. Mardonova and Choi (2018) have reviewed the latest trend in wearable device technology that includes the classification of wearable devices with some examples of their utilization in various industrial fields as well as the features of sensors used in wearable devices.

Spatio-temporal data mining could also be applied in the field of urban planning and building smart road systems. Combining the traffic information with discovered patterns in mining spatial trajectory data from vehicles could result in a number of interesting findings, which may include a better understanding of traffic blockages, the behavioral patterns of a driver with respect to traffic status, etc. Results drawn from related studies could also provide guidance in planning and build a more intelligent and efficient traffic system. There are already some existing research studies in this area. For instance, an existing US patent uses vehicle telematics data to provide traffic forecasts and driving guidance for freeway drivers (U.S. Patent No. 6,401,027, 2002).

2.1.1 Data collection motivation.

Spatio-temporal data collection has experienced explosive growth in recent years. Advances in instrumentation and computation have boosted the number of electronic devices that record petabytes of data into various databases, while also bringing up a big but promising challenge of mining these data efficiently. As stated in the survey conducted by

Eldawy and Mokbel (2015), while an explosive growth of spatio-temporal data has been happening during recent years, a lack of efficient systems to extract meaningful patterns from the spatio-temporal data set is hampering the growing need of managing and analyzing such data. As mentioned in previous sections, recognized patterns of spatio-temporal data could be applied to a variety of fields, including movement study and prediction, vehicle insurance, and many others.

A good representative of spatio-temporal data, which is being analyzed in this thesis, is spatial trajectory data and multivariate spatial time series data. Benefited from the high penetration rate of GPS devices installed on vehicles as well as the advancement of remote sensing technologies and location recording accuracy and frequency, together with the popularity of personal tracking devices such as smart wristbands and smart sensors, the quantity and quality of moving objects data and sensor data has been growing significantly over the years, unlocking potential value in mining these types of data.

With more and more cloud-based telematics devices emerging, an enormous amount of data is being stored into cloud databases every second. The newly opened market for personal trackers has also enriched the sources of moving objects' data. Users' data are recorded first inside the device, then synchronized through wireless mediums and uploaded to the database. Type and amount of data collected may vary largely among producers, but in general that basic data types, such as certain coordinates at different timestamps, should always be present.

If proper data representations and corresponding mining algorithms are developed, analysts can get rather useful patterns from the mining results. For example, in the case of mining driver telematics data, questions such as each driver’s driving habit could be answered, which might have a great impact on the vehicle insurance industry.

2.2 Knowledge Discovery and Data Mining in Related Area

According to Fayyad, Piatetsky-Shapiro, and Smyth (1996), knowledge discovery in databases (KDD) or data mining can be defined as the nontrivial extraction process of implicit, previously unknown, and potentially useful information from data. This implicit, previously unknown, and potentially useful information which is referred to as knowledge is hidden in the databases and is usually in the form of relationships among data items. These relationships can be in the form of functional, or partial functional dependencies. Their discovery analysis and characterization may involve the use of various techniques. The process of applying KDD in a general situation consists of the following phases according to Han, Pei, and Kamber (2011):

1. Understanding the Application domain: This includes the understanding of the relevant prior knowledge and the goals of the application.
2. Extracting the target data set: This includes the selection of a data set or focusing on a subset of variables.
3. Data preprocessing and transformation: This phase improves the quality of the actual data for data mining. It also increases the efficiency of data mining by reducing the computational effort for mining the preprocessed data. Data preprocessing involves data cleaning, data transformation, data integration and data reduction or compression. Data cleaning consists of some basic operations such as normalization, noise removal, handling of

missing data, reducing redundancy etc. Data integration includes integrating multiple and heterogeneous data sets from different data sources. Data reduction finds useful features to represent the data by means of dimensionality reduction, feature selection, discretization etc.

4. Data mining: This phase constitutes one or more of the following functions including classification and prediction, association analysis, cluster analysis etc.
5. Pattern interpretation and evaluation: This phase includes interpreting the discovered patterns and the possible visualization of them. Visualization is important in that it increases understandability from the perspective of humans. The mined patterns can be evaluated automatically or semi-automatically to identify the interestingness or usefulness of them.
6. Using discovered knowledge: This phase incorporates the discovered knowledge into the expert system and actions can be taken based on this knowledge.

KDD or data mining techniques have been practically applied in a wide spectrum of areas that benefits from discovering patterns over the data sets. In the following subsections, some categories of common data mining algorithms related to this thesis will be discussed.

2.2.1 Association, classification, and clustering.

In the below, we will introduce the three common classes of data mining tasks. Association analysis mines or generates rules from the data. Association rule mining refers to discovering associations among different attributes (Cheung et al. 1996; Agrawal & Srikant, 1994). It tries to describe the relationship among data items. A population application of association rules mining is the analysis of supermarket transaction data, helping the planning of marketing strategies. Popular algorithms include AIS (Agrawal, Imielinski & Swami, 1992), SETM (Houtsma & Swami, 1993), and Apriori (Agrawal & Srikant, 1994).

Classification analysis classifies a data item into one of several predefined categorical classes. Based on the predefined classes in the training objects, the general approach involves a systematic search for minimal descriptions, which can distinguish between members of different classes. In machine learning terminology, this is called supervised learning (Tou & Gonzalez, 1974), i.e. learning is done with explicit training examples. Popular algorithms include *k*-nearest neighbor (*k*-NN) (Dasarathy, 1991), decision-tree generators (ID3 by (Quinlan, 1987), C4.5 by (Quinlan, 1993), CART by (Breiman et al., 2017)), neural networks (Aleksander & Morton, 1990; Beale & Jackson, 1990) and genetic algorithms (Davis ,1991; Holland & Goldberg, 1989; Holland, 1987).

Cluster analysis maps a data item into one of several clusters, where clusters are natural groupings of data items based on distance measure (as known as similarity measure). In general, the resulting clusters should exhibit high within-cluster homogeneity and high between-cluster heterogeneity. Clustering is dependent on the distance measure to be applied. In machine learning terminology, this is a type of unsupervised learning (Tou & Gonzalez, 1974), i.e. learning is done without explicit training examples. Commonly, clustering algorithms can be classified into two broad categories: (1) hierarchical and (2) non-hierarchical. Hierarchical clustering involves the construction of a hierarchy or tree structure. Popular hierarchical clustering algorithms include agglomerative (Milligan, 1980), Chameleon (Karypis, Han & Kumar, 1999), DIANA (Kaufman & Rousseeuw, 1990), AGNES (Kaufman & Rousseeuw, 1990) and BIRCH (Zhang, Ramakrishnan & Livny, 1996). Non-hierarchical clustering does not involve the construction of the tree structure while it first selects a cluster center or seed and then all objects or data points within a pre-specified threshold distance are included in the resulting cluster. Popular non-hierarchical clustering algorithm includes *k*-means (Forgy, 1965; MacQueen, 1967), CLARA (Kaufman & Rousseeuw, 2009), CLARANS (Ng & Han, 2002), CLIQUE (Agrawal et al., 1998) and SOM (Kohonen, 2012).

2.2.2 Discretization of continuous data.

In data mining and machine learning, data discretization techniques can be used to reduce the number of unique values for a given continuous attribute by dividing the range of the attribute into intervals (Han & Kamber, 2001). "Discretization is a technique to partition continuous attributes into a finite set of adjacent intervals in order to generate attributes with a small number of distinct values" (Tsai, Lee, & Yang, 2008, p. 715). In short, a continuous attribute (a.k.a variable) can be discretized into a finite number of discrete intervals (Kurgan & Cios, 2004). Interval labels can be applied to replace actual value. There are several reasons to perform discretization as a data preprocessing step for data analysis. The obvious reason is it reduces and simplifies the original data, leading to a concise, easy-to-use, and knowledge-level representation of the data and mining results. In data mining algorithms, many of the effective ones have been developed to handle categorical attributes such as AQ (Kaufman & Michalski, 1999; Michalski, Mozetic, Hong, & Lavrac, 1986), CLIP (Cios & Kurgan, 2002; Cios & Kurgan,

2004) and CN2 (Clark & Niblett, 1989), while others can deal also with continuous attributes but have better performance on categorical attributes (Wu et al., 2006). Since continuous data can be discretized into a finite set of discrete intervals, discretization can be performed before the learning process (Chan, Ching, & Wong, 1992). A good discretization algorithm can produce a concise summarization of continuous attributes but also facilitate learning faster and more accurately (Liu, Hussain, Tan, & Dash, 2002). For many real-world data sets, attributes may be in a combination of discrete and continuous types. Back in the late 1980s, there was no common and fully integrated approach to inductive learning (IL) which can handle both mixed-mode continuous and discrete data simultaneously (Wong & Chiu, 1987). Ching, Wong, and Chan (1995) have proposed a class attribute dependent discretization (CADD) method to partition continuous data attributes. Basically, two important decisions must be made for discretization. Firstly, the number of discrete intervals or bins must be selected. Secondly, the width of the intervals must be determined. Their method can automatically determine the most preferred number of intervals to make the first decision and seeks to maximize the mutual dependence between the discrete intervals and class attribute to make the second decision. Later, optimal class dependent discretization (OCDD) method applies a dynamic programming technique to search for global optimum discretization scheme to efficiently partition the continuous attributes in a supervised setting (Liu, Wong, & Wang, 2004). Our previous work mixed-mode attribute clustering algorithm (MACA) and a fuzzy version of it (FMACA) was proposed to extend CADD and OCDD for maximizing the interdependence among attributes to break down attributes into attribute clusters for further processing such as discretization and classification (Wong et al., 2010; Wu, Chan, & Wong, 2011). The discretization procedure in MACA separately applies on each attribute cluster to locally optimize the partitioning by treating the most representative attribute, which is referred to as mode and is with the highest total interdependency to all the other member attributes, in the attribute cluster as the class. Therefore, using the mode, it drives the discretization of other continuous attributes in the attribute cluster using class attribute dependent discretization method, i.e. OCDD. From the experiments, there are some circumstances that the class attribute is not always the mode. In this regard, if class attribute is absent in the data set, MACA can still operate the clustering of attributes and then discretize the continuous attributes by the modes of attribute clusters. This shows MACA is capable to cope with both supervised and unsupervised situations.

According to Liu et al. (2002), the discretization algorithms can be classified into five axes: supervised versus unsupervised, static versus dynamic, global versus local, top-down (splitting) versus bottom-up (merging), and direct versus incremental. Out of the five axes, Tsai, Lee, Yang (2008) summarize them as follows.

1. Supervised methods discretize attributes with the consideration of class information, while unsupervised methods do not.
2. Dynamic methods consider the interdependence among the attributes and discretize continuous attributes when a classifier is being built. On the contrary, the static methods consider attributes in an isolated way and the discretization is completed prior to the learning task.

3. Global methods, which use total instances to generate the discretization scheme, are usually associated with static methods. On the contrary, local methods are usually associated with dynamic approaches in which only parts of instances are used for discretization.
4. Bottom-up methods start with the complete list of all continuous values of the attribute as cut-points and remove some of them by merging intervals in each step. Top-down methods start with an empty list of cut-points and add new ones in each step.
5. Direct methods, such as Equal Width and Equal Frequency (Chiu, Wong, & Cheung, 1991), require users to decide on the number of intervals *k* and then discretize the continuous attributes into *k* intervals simultaneously. On the other hand, incremental methods begin with a simple discretization scheme and pass through a refinement process although some of them may require a stopping criterion to terminate the discretization. (p. 715)

A more detailed discussion about the five axes mentioned above can be found in the paper of Liu et al. (2002). In this section, the discussion of discretization algorithms will follow the axis of top-down versus bottom-up.

Class-Attribute Contingency Coefficient (CACC) by Tsai, Lee, and Yang (2008) is one of the latest top-down discretization algorithms. The main contribution of it is that it can generate a good discretization scheme and its discretization scheme can lead to the improvement of classifier accuracy like that of C5.0. The quality of a discretization scheme can be measured by Class-Attribute Interdependence Redundancy (CAIR) proposed by Ching, Wong, and Chan (1995). According to Tsai, Lee, and Yang (2008), the general goals of a discretization to achieve include: 1) a high-quality discretization scheme to help users understand the data easily, 2) the scheme should lead to the improvement of accuracy and the efficiency of a learning algorithm which is the training time and the number of rules generated to reach the classification accuracy, and 3) the discretization process should be as fast as possible. Class-attribute Interdependence Maximization (CAIM) by Kurgan and Cios (2004) is another top-down discretization algorithm with good performance in comparison with seven state-of-the-art top-down discretization algorithms. On average, experiments show that CAIM obtains high CAIR value, and using it as a preprocessor for classification algorithm, it produces the least number of rules and reaches the highest classification accuracy (Kurgan & Cios, 2004). Later, MACA was developed to flexibly deal with the effect of the class attribute by investigating the relationship of the class attribute and other attributes. Since their finding reveals that some attributes are not highly dependent on the class attribute, the objective to optimize the CAIR value in the discretization might not be the best objective. This rationale leads to the discretization being done separately by each attribute cluster where the objective of discretization in each attribute cluster is to optimize the CAIR by treating the most representation attribute as the class attribute of each attribute cluster. Their experiments using simulated data, repository data and real data showed that the classification accuracy can be enhanced in comparison to the other discretization methods.

Top-down (splitting) and bottom-up (merging) discretization algorithms consist of unsupervised and supervised ones. Two typical unsupervised top-down algorithms are Equal Width and Equal Frequency (Chiu, Wong, & Cheung, 1991). Other the state-of-the-art supervised top-down algorithms are Paterson-Niblett (Paterson & Niblett, 1987), maximum entropy (Wong & Chiu, 1987), information entropy maximization (Fayyad & Irani, 1993), class-attribute dependent discretizer (CADD) (Ching, Wong, & Chan, 1995), class-attribute interdependence maximization (CAIM) (Kurgan & Cios, 2004), fast class-attribute interdependence maximization (FCAIM) (Kurgan & Cios, 2003) and class-attribute contingency coefficient (CACC) (Tsai, Lee, & Yang, 2008). FCAIM has been proposed as a faster version of CAIM extension. The discretization criterion, the stopping criterion and the time complexity between them are the same while the only difference is the initialization of the boundary point. FCAIM was faster than CAIM with similar C5.0 classification accuracy where CAIM obtained a slightly better CAIR value (Kurgan & Cios, 2003). Experiments showed that CAIM and CACC are superior to other top-down discretization algorithms as their discretization schemes can generally maintain higher interdependence between target class (also called class label or class attribute) and discretized attributes, generate lesser number of rules to attain higher classification accuracy (Kurgan & Cios, 2004; Tsai, Lee, & Yang, 2008). That the abovementioned supervised discretization algorithms aim at seeking a local optimal solution, optimal class dependent discretization (OCDD) searches for global optimum discretization scheme which is proven to be an effective approach experimentally (Liu, Wong, & Wang, 2004). It is based on the concept of dynamic programming which searches for the best partition from all possible settings for each iteration. Our current work in this thesis adopts this kind of class dependent approaches in some of the experiments to optimize the dependency of the attributes toward the class attribute in the partitioning, showing good results towards the improvement of the classification accuracy.

Four famous bottom-up algorithms are ChiMerge (Kerber, 1992), Chi2 (Liu & Setiono, 1997), Modified Chi2 (Tay & Shen, 2002) and Extended Chi2 (Su & Hsu, 2005). Since bottom-up (merging) algorithms start with all continuous values and recursively remove points by merging intervals, the computational complexity is generally higher than top-down (splitting) algorithms. To merge adjacent intervals, the significance test is performed to test whether or not two adjacent intervals should be merged. Another basic requirement is that some parameters need to be specified by users such as the significance level, maximal and minimal intervals and etc. Using these bottom-up approaches as preprocessors for C5.0 classification, experiments by Su and Hsu (2005) showed that Extended Chi2 outperformed the other bottom-up discretization algorithms as its discretization scheme can reach the highest accuracy on average.

In this thesis, we adopt both supervised and unsupervised discretization techniques when developing the mining algorithms so as to cope with different situations in a flexible manner. To the best of our knowledge, supervised discretization algorithms are generally with better performance, which can improve the classification accuracy and simplify the classification rules, than unsupervised discretization algorithms due to the reason that the supervised one is benefited from a priori knowledge.

2.2.3 Dimensionality reduction and attribute clustering.

Since the introduction of machine learning, researchers have targeted a relatively small set of attributes. Our previous research study (Wong, Wu, Wu, & Chan, 2010) has found that "as the size of databases and the diversity of attributes increased, the performance of data clustering was challenged although the classification problems were not seriously affected yet their effectiveness was diminishing". In supervised learning, the problems were partly solved through dimensionality reduction and feature engineering that use domain knowledge of the data to create features making machine learning work. Dimensionality reduction can be categorized mainly into feature extraction and feature selection (Tang, Aleyani & Liu, 2014). Subsequently in the late 1980s, when data mining and pattern discovery became an independent discipline and prominent in the database community, the solutions to tackle dimensionality problems have been being developed. Nowadays, the performance of many state-of-the-art clustering algorithms are heavily dependent on the quality of data pre-processing and preparation due to the nature of large-scale mixed-mode database with a large number of attributes. In many machine learning algorithms and pipelines, data preparation is always the first step to begin with. The gathering of all relevant data and the aggregation of different data sources to extract raw attributes that might have predictive power will easily grow the dimensions of the data. In unsupervised learning, attribute clustering was employed to tackle the dimensionality problems. In supervised learning, class-dependent discretization can also be used to convert the continuous data into interval data (Au, Chan, Wong, & Wang, 2005). To cluster or select attributes, the t-value method is widely used (Agrawal et al., 1992). Au et al. (2005) argue that the t-value can only be used when the samples are pre-classified. If no class information is provided, it cannot be used for attribute selection. So, the attribute clustering algorithm (ACA) was proposed to cluster attributes (Au et al., 2005). In ACA, continuous data must be converted into interval data before the application of attribute clustering. MACA (Wong et al., 2010) and FMACA (Wu, Wong & Chan, 2011) extended ACA so that it is able to deal with mixed-mode data by introducing attribute interdependence redundancy measures between attributes of various attribute types and a multiple interdependence measure (Alon et al., 1999; Wong, Chiu, & Huang, 2002) for selecting attributes with the highest correlation with the rest of attributes within an attribute cluster. A method using a Chi square-based discretization method for feature selection that eliminates some irrelevant and/or redundant attributes can be adopted as well (Liu & Setiono, 1997). This method discretizes numeric attributes repeatedly until some inconsistencies are found. Feature extraction approaches transform features into new feature space with reduced dimensionality. Some approaches combine the transformed features with original features. Common feature extraction approaches are principal component analysis (PCA), linear discriminant analysis (LDA) and canonical correlation analysis (CCA). Feature selection approaches choose a subset of features from the original set of features that minimize redundancy and maximize relevance to the class attribute. Feature selection can be supervised, unsupervised or semi-supervised. A comprehensive review of unsupervised methods and supervised methods can be found in the paper of Alelyani, Tang and Liu (2013) and Tang, Aleyani and Liu (2014) respectively. Basically, supervised methods include filter models that separate feature selection from classifier learning relying on measures of training data such as fisher score (Duda, Hart & Stork, 2001) and information gain (Peng, Long & Ding, 2005), wrapper models that use the predictive accuracy of a selected learning algorithm to assess the quality of selected features and embedded models that perform feature selection during the training of a classifier. However, these methods rely on the quality of the labeled data. Unsupervised methods, similarly, include filter models that evaluate the score of each feature according to certain criteria such as information entropy (Dash, Choi, Scheuermann & Liu, 2002) and dependency measure (Talavera, 1999), wrapper models that utilizes a clustering algorithm to evaluate the quality of selected features and hybrid models that combine filter and wrapper models. While unsupervised feature selection methods do not require class information, an evaluation of the relevance of features is difficult. To overcome the drawback of supervised and unsupervised feature selection methods, we, in this thesis, propose a unified framework to benefit from the efficient feature extraction, and better classification and clustering quality from the attribute clustering and feature selection models.

After dimensionality reduction and transforming all data attributes, pattern discovery (Wong & Wang, 2003), pattern clustering (Wong & Li, 2008; Wong & Li, 2010) and pattern summarization (Wong & Li, 2008; Wong & Li, 2010) can then be applied to the data set.

2.2.4 Pattern discovery.

Pattern discovery for intelligent decision support, knowledge-based reasoning, and data analysis applies more and more to large-scale complicated systems and problem domains (Chiu, Wong, & Cheung, 1991). In most of the existing systems, data preprocessing, such as data cleansing, filtering and attribute reduction, is the first step to remove noises, to bring out more relevant information from the data and to reduce the search space (Wong et al., 2010). They point out that,

> However, they often depend on prior knowledge, such as parameters and preconceived classificatory framework. Thus, they sometimes could be very biased to the application area and usually involve long iterative search and examination process. To respond to these needs, a data-driven pattern discovery approach has been advanced (Wang & Wong, 1979). It is able to discover, in an unbiased manner, statistically significant events automatically from a relational table, and to generate decision rules for categorization and prediction. (p. 860)

In general, pattern discovery can be defined as, being a subfield of data mining, extracting previously unknown patterns, which can be a set of items, subsequences, or substructures that occur frequently together or are correlated strongly, and regularities in the data by exploring a space of possible patterns to determine which are present in a set of reference data (Agrawal, Imieliński & Swami, 1993; Wong & Wang, 2003). It is a useful tool for categorical data analysis (Agresti, 2003). One of our early works (Wu, Chen, Zhu & Chan, 2013) was to successfully apply the pattern discovery approach to extract patterns from the relational database of a mobile application advertising service to learn a predictive model to optimize the click-through rates. This optimization task is a simplified problem of this thesis that aims at dealing with raw spatio-temporal data. Although the data of this problem is also spatio-temporal, the complexity of it is rather simple. It demonstrated the practicality of pattern discovery approach. Pattern discovery theory forms the foundation for many data mining tasks such as association, sequential pattern mining, cluster analysis, classification, pattern analysis in spatiotemporal, multimedia, time series and stream data and so on (Han, Pei & Kamber, 2011).

In many real-world problems, one of the drawbacks of applying a pattern discovery approach is that it typically produces an overwhelming number of patterns, resulting in a very difficult and time-consuming effort for problem comprehension and interpretation. To combine fragments of information from individual patterns to produce more generalized forms of information and to use them to further explore or analyze the data, pattern clustering (Wong & Li, 2008; Li, 2010) is developed to simultaneously cluster the discovered patterns and their associated data. Pattern pruning and summarization can be applied as pattern post-processing method to select from the discovered patterns a most representative subset which could be considered as the summary of the pattern cluster, rendering a small number of patterns that retain the most crucial information (Wong & Li, 2008; Li, 2010; Zhou, Li & Wong, 2016). As a consequence, the important local patterns are retained, and the pattern groups are rendered in the original data space globally.

2.3 Overview of Spatial Trajectory Data Mining Problems and Algorithms: A Survey of Related Work

The main difficulty of applying big data techniques to spatial trajectory data set lies in the lack of efficient and effective algorithms for mining the data sets. Analyzing and extracting meaningful patterns from spatial trajectory data sets are considered as challenging, mainly due to the often large-in-size and sometimes noisy nature of spatial trajectory data. A number of existing data representation and mining methods are modified to deal with spatial trajectory data sets, while researchers are still devoting effort to find more generic and computationally and time efficient algorithms on representing and mining such data sets.

2.3.1 Spatial trajectory clustering.

Yuan et al. (2017) have conducted a comprehensive survey on trajectory clustering algorithms. They summarized the research areas into three aspects. The first attempts to extract features from full trajectories such as speed, direction, acceleration, and others and discover movement patterns. The proposed approach on spatial trajectory data mining in this thesis falls into this aspect. The second attempts to find suitable distance measurements between trajectories. The third attempts to develop scalable algorithms in runtime and storage. They further divided the algorithms into 5 categories, namely spatial-based, time-depended, partition- and group-based, uncertainty-based, and semantic-based. We do not adopt this categorization as some of the aspects and algorithms are beyond the scope of this work. The following will present an overview of the related work.

Mamoulis, Cao, Kollios, Hadjieleftheriou, Tao and Cheung (2004) developed a novel method for mining historical spatial trajectory data. Considering the specific data type used in this thesis as a sample data set, which is spatial trajectory data, most of the related methods are based on *k*-means algorithm. A research project by Ashbrook and Starner (2002) applied a variant of *k*-means algorithm for clustering the GPS data. However, existing approaches that use *k*-means require a reasonable estimation of the number of clusters *k*, which might be difficult to obtain (Cao, 2009).

Another approach to cluster spatial trajectory data uses a hybrid-clustering algorithm that combines hierarchical method with grid-based method. Hierarchical clustering is particularly useful when it is difficult to determine the best clustering from optimizing certain scoring functions. The general idea is to create an initial clustering by putting all data point into a disjoint set of clusters. The proximity is calculated based on the distance between cluster centroids. Then at each step, clusters that are nearest are successively merged together, reducing the number of clusters. The iteration stops when there is no merging possible. This approach, however, is not so practical when applied to large-scale data sets, and improvements using pre-grouping and indexing the data have been introduced to reduce the complexity of the algorithm (Cao, 2009).

2.3.2 Spatial trajectory classification.

The aim of trajectory classification is to differentiate between trajectories of different status, such as moving behaviors, styles, and purposes. A class label can be assigned to a raw trajectory and it can lead to many practical applications, including, but not limited to, ride sharing, map navigation, and context-aware pervasive systems. Here, we summarize the relevant background of trajectory classification based on Zheng (2015) and the motivation for the current work as below.

A typical trajectory classification follows three steps from trajectory segmentation, to feature extraction from each segment and then model building to classify each segment. Some existing sequential pattern mining methods (e.g. Hidden Markov Model, Dynamic Bayesian Network, Conditional Random Field) can be applied by treating a trajectory as a sequence. An early work from Krumm and Horvitz (2004) uses an HMM to classify trajectories into binary statuses. Later, some other work in (Sohn et al., 2006; Zhu et al., 2011) attempt to classify the user mobility and taxi status into 3 statuses according to GPS trajectories based on point-based detection method and geographic data such as road networks and points of interest. Some more recent approaches (Dong, Li, Yao, Li, Yuan & Wang, 2016; Endo, Toda, Nishida & Kawanobe, 2016) consider using deep learning for feature extraction to avoid handcrafted features. In the experiment section, deep learning based methods will be assessed for performance comparison. The current work is capable to deal with a class with multiple possible values and does not require to rely on domain knowledge of geography.

To classify a trajectory with multiple possible values of a class during a single trip, a trajectory is partitioned into segments and is extracted speed-related features to feed into a Decision Tree classifier with graph-based post-processing step to enhance the inference (Zheng, Liu, Wang & Xie, 2008). This method is tailored to tackle a situation where the moving object can change modes in a single trip. The current work considers diverse types of features including route, speed, turning and stop point in feature generation step and considers both local and global information in pattern mining and classifier training step.

Some promising algorithms for location-based activity recognition and popular place discovery are proposed by Liao, Fox, and Kautz (2007) and Patterson, Liao, Fox, and Kautz (2003). They divided a GPS trajectory into 10-m segments based on corresponding street patches by using the CRF-based map-matching algorithm. Then, the model classifies a GPS sequence into an activity sequence and identify popular places of a person based on the extracted street features.

A recent survey by Eldawy and Mokbel (2015) categorized the existing work in the area of big spatial data in three dimensions – implementation approach, underlying architecture, and spatial components. For the implementation approach, more than half of the existing approaches apply an existing system for non-spatial trajectory data as a black box and rely on user-defined functions to query these systems. The drawback is these systems are not optimized to deal with spatial trajectory data. Some of them introduced spatial trajectory data elements to the existing system rather than building from scratch to improve the performance but still, the core algorithms are not optimized to deal with the integration. For underlying architecture and spatial components, since most of the approaches are based on existing systems, their architecture and spatial components follow the existing systems. MapReduce-based system is commonly used (Dean & Ghemawat, 2008). Some may use a resilient distributed data set (RDD) (Zaharia, et al., 2012), key-value stores (Chang et al., 2008). To the best of our knowledge, we cannot find any notable work that utilizes column-oriented databases such as Vertica (Stonebraker et al., 2005), Dremel (Melnik et al., 2010) and Impala (Bittorf et al., 2015). The reason for not utilizing them is due to the core of the existing systems not designed for spatial trajectory data, the scalability and powerful feature of processing points and polygons of spatial data by them hence cannot be integrated and activated. To query these systems such as basic search operations, join queries, computational queries, and data mining queries, most of them rely on iterative processing algorithms such as *k*NN and *k*-means, that is the solution is refined in each iteration until a solution which meets certain accepting criteria is found. The performance of these algorithms is largely dependent on the computing framework. Hadoop (Bu, Howe, Balazinska & Ernst, 2010) although dominates the area of big data processing is not suitable to operate iterative algorithms because of the large overhead for each iteration. Therefore, we cannot find notable work in spatial trajectory data mining that uses Hadoop. *k*-means algorithm is often implemented straightly in the MapReduce framework so that each iteration is processed by a separate MapReduce job (Zhao, Ma & He, 2009). The performance is bad as the overhead of Hadoop in each iteration consumes a lot of resources. Spark is well suited to machine learning algorithms as it allows to load data into a cluster's memory and query it repeatedly (Zaharia, Chowdhury, Franklin, Shenker & Stoica 2010). However, to the best of our knowledge, we cannot find any notable work that uses Spark for spatial trajectory data mining tasks. It is worth to investigate the feasibility and practicality of using Spark as the implementation of the mining algorithms.

This section surveyed the state-of-the-art work in the area of big spatial trajectory data. The existing approaches, architectures, and components are studied. It is true that a number of significant studies have recently been done on how data scientists and engineers tackle spatial trajectory data from adopting existing data analysis approaches and systems, but the question of how those systems and approaches integrate spatio-temporal components and patterns has rarely been considered. Consequently, we identify some research gaps from which to develop an integrated algorithm to the spatial trajectory data mining.

2.4 Data Mining Techniques for Multivariate Spatial Time Series: A Survey of Related Work

Recently, there have been some data mining and statistical techniques available for use in MSTS data analysis. In the following sub-sections, we briefly review some existing tools, techniques, and databases developed specifically for tackling MSTS analysis problems. In below sections, the related works that lead to the development of this thesis will be provided.

2.4.1 Overview of multivariate time series data mining problems and approaches.

Time series value is numeric. In the real world, it is unusual to find time series behave independently of others. Analyzing multiple time series simultaneously is an important task and is referred to as MTS analysis. MTS, which may carry a class label (Xing, Pei & Keogh, 2010), can be considered as a collection of vectors. In the spatio-temporal context, MTS can exist in various locations and be referred to as MSTS. For example, climate data is MSTS recorded from several sensors to describe the weather by meteorological variables such as temperature, precipitation, humidity, wind speed at a given location, and may come from multiple locations of either arid or semiarid climate types, labeled as “dry arid” and “dry semiarid”.

Although quite a number of studies have been conducted on time series data analysis and mining methods, only a few of them focus on those with multivariate nature. Fu (2011) gave a comprehensive review on time series data mining and mentioned some recent research issues related to MTS. We incorporate some of these relevant issues in this subsection and summarize the current development. A primitive approach proposed by Sankoff (1983) aims to find the longest common subsequence shared by MTS, but the algorithm grows exponentially in time with respect to the number of sequences concerned, making this approach undesirable against most of the possible applications.

Later, Oates (1999) came up with a method to discover "distinctive subsequence", in other words, "abnormal" pattern occurrences in an MTS. In their study, they randomly sampled the time series with length L sub-sequences and tried to discover patterns by clustering these subsequences based on Dynamic Time Warping. However, this study was restricted to discover distinctive subsequences, but also did not consider the possible relations between different variables in the data set.

A number of studies on MTS tried to utilize principle component analysis (PCA) to reduce the number of dimensions of the feature space in the data set. PCA extracts principle components, which represent the most distinctive features of an MTS data set. In 2001, Rosén and Yuan (2001) utilized dynamic PCA to obtain principle components from different MTS and then used fuzzy *c*-means to cluster the results obtained by PCA. This approach has been used by a lot of researchers for mining MTS and is effective when they treat a set of MTSs as a single item and aim to cluster a number of these items. However, it lacks the ability to find patterns between variables in an MTS. Based on this, Yang and Shahabi (2004) proposed a variant that uses PCA and singular value decomposition (SVD) to raise the precision of similarity measure between different MTS. Similar optimizations over PCA based clustering have been proposed by others, such as Singhal and Seborg (2005), using similarity factors based on PCA and Mahalanobis distance between data sets for the similarity measure. Some work adopted Euclidean distance for similarity measure. PCA is also used in feature selection, as a preprocessing step in MTS analysis by Yoon et al. (2005).

Hidden-Markov models (HMM) were used by Owsley et al. (1997) to cluster MTS. This method focuses on clustering different MTS data sets and requires well-established a priori information of initial classes. Zhou and Chan (2014) proposed a model-based clustering algorithm to cluster MTS based on the discovered temporal patterns in each MTS and compare them with those discovered in the others so that MTS that exhibit similar patterns can be grouped together in the same cluster. This method discovers temporal patterns using confidence value a.k.a lift ratio to represent the relationship between different variables. It is application independent and can perform without any domain knowledge about relevant features or any assumption about underlying data models.

2.4.2 Multivariate spatial time series pattern mining and clustering.

Technological advancement in remote sensing technology such as telematics, climatology is considered to be one of the most promising factors for allowing the large-scale collection of multivariate spatial time series data. An MSTS is a complex data type that consists of MTS in multiple locations. A better understanding of MSTS may hence result in better understanding of how regional variables are inter-related and changed temporally and how this local information is correlated globally to other locations spatially. Unfortunately, since mining MSTS involves the consideration of high dimensions of time domain in the time series or sequence with each interacting with one or more other time series in the same or across space domain, the task of mining spatio-temporal patterns from MSTS is very difficult. In an attempt to handle this problem, some researchers and practitioners have developed some algorithms, including model-based approach (Owsley et al., 1997; Zhou & Chan, 2014), and similarity-based approach (Singhal & Seborg, 2005; Yang & Shahabi, 2004; Yoon et al., 2005). Some of them are summarized briefly in the previous section. Other than the above approach, some more attempts have been done to analyze MSTS using data mining techniques. Given a data set, the data mining goal is to discover hidden regularities and structure inherent in it. It is different from hypothesis-based approaches that look for known and pre-specified patterns in the data set. Data mining/pattern discovery approach is data-driven. The obvious difference is that pattern mining does not require patterns to be known ahead of the time, but the search process automatically detects and extracts patterns hidden in the data set.

For MSTS analysis, several pattern-mining approaches have been proposed in the literature. If the data sets of interest contain spatial information, one could obtain more interesting results by mining such information. Coppi, D'Urso, and Giordani (2010) altered the fuzzy c-means to incorporate spatial influence while clustering MTS. This method introduced a term called a spatial penalty, influenced by the contiguity between different spatial units. Data sets that represent neighbors in space are more likely to be classified into the same cluster by this term. An application of clustering Italian provinces proved the effectiveness of it.

Shumway (2014) used Kullback-Leibler discrimination and Chernoff information measure to calculate the disparity between 2 sets of MTS. They calculated the distance matrix based on the discrimination results of these 2 methods and used it for *k*-means clustering. A method of measuring linear and non-linear dependence between groups of MTSs was proposed by Pascual-

Marqui (2007). Our study in this thesis will also reference other methods to measure the relation between univariate time series.

Längkvist, Karlsson, and Loutfi (2014) give a review of the recent developments in deep learning and unsupervised feature learning for time-series problems. They argue that since these deep learning techniques have shown promise for modeling image data and static, applying them to time series data is getting more and more attention. Here we summarize the important models and techniques that are related to the current work. Having the major advantage of learning features from the data without the need to engineer handcrafted features, they present models and techniques that are used for modeling temporal relations. These models and techniques include Restricted Boltzmann Machines (RBM) (Hinton, Osindero & Teh, 2006; Hinton & Salakhutdinov, 2006; Lee, Ekanadham & Ng, 2008), Conditional RMB (cRBM), Gated RBM (GRBM) (Memisevic & Hinton, 2007), auto-encoder (Poultney, Chopra & Cun, 2006; Bengio, Lamblin, Popovici & Larochelle, 2007; Bengio, 2007), Recurrent Neural Network (RNN) (Hüsken and Stagge, 2003), deep learning, convolution and pooling, temporal coherence, Hidden Markov Model (HMM) (Rabiner & Juang, 1986). RBM is a generative probability model between input units and latent units connected with a weight matrix and bias vectors. cRBM is an extension of RBM that models multivariate time series data. It consists of auto-regressive weights that model short term temporal structure and connections between past visible units to the current hidden units. GRBM is another extension of the RBM that models the transition between two input vectors. It models a weight tensor between the input, the output, and the latent variables. Auto-encoder was originally introduced as a dimensionality reduction algorithm whose basic linear version learns the same representation as a Principal Component Analysis (PCA). The layers of visible units, hidden units, and the reconstruction of the visible units are connected via weighted matrices and the hidden layer and reconstruction layer have bias vectors. RNN is used for modeling sequential data. It is obtained from the feedforward network by connecting the neuron's output to their inputs. The short-term time-dependency is modeled by the hidden-to-hidden connections without using any time delay-taps. They are usually trained iteratively via a procedure known as back-propagation-through-time (BPTT). RNNs can be seen as very deep networks with shared parameters at each layer when unfolded in time. The goal of a deep network is to build features at the lower layers that will disentangle the factors of variations in the input data and then combine these representations at the higher layers. Convolution is a technique that is particularly interesting for high-dimensional data, such as images and time-series data. In a convolutional setting, the hidden units are not fully connected to the input but instead divided into locally connected segments. Convolution has been applied to both RBMs and auto-encoders. Pooling is an operator used together with convolution which combines nearby values in input or feature space through a max, average or histogram operator. The purpose of it is to achieve invariance to small local distortions and reduce the dimensionality of the feature space. Temporal coherence here refers to techniques that capture temporal coherence in data such as smoothness penalty on the hidden variables in the regularization. HMM is a popular model for modeling sequential data driven by two probability distributions, namely transition distribution, which defines the probability of going from one hidden state to the next hidden state, and observation distribution, which defines the relation between observed values and hidden states. A recent attempt by Tian, Zhou, and Guan (2017) has proposed a general framework to integrate traditional clustering methods into deep learning (DL) models. They claimed that most existing DL based clustering techniques have separate feature learning (via DL) and clustering (with traditional clustering methods), their proposed framework simultaneously learns feature representation and does cluster assignment under the same framework. It is a general and flexible framework that can employ different networks and clustering methods. In their demonstration, they integrated *k*-means and Gaussian Mixture Model (GMM) into deep networks. Based on the insight from this framework, in our experiment, we will train a deep network by inputting the clustering assignment for performance comparison.

While most of the studies reviewed above focus on clustering different MTS data sets, some studies emphasize discovering patterns within a single data set. Bünau, Meinecke, Király, and Müller (2009) came up with a method, which breaks down an MTS data set into stationary and non-stationary parts. Being named "stationary subspace analysis", this method extracts stationary sources from non-stationary time series and was successfully applied to EEG data for extracting stationary patterns of brain activity. As for discovering common trends in MTS, Zuur, Fryer, Jolliffe, Dekker, and Beukema (2003) proposed a method using EM algorithm to perform dynamic factor analysis which models data as trends, explanatory variables and noise. This method could handle data set with missing values or few data points, so it claimed to achieve superior applicability over its precedents. Another area in MTS analysis is the change point problem, which finds out the timestamps when the covariance structure of the series changes abruptly. Lavielle and Teyssiere (2006) proposed a method to solve this problem, which outperforms previous methods.

There are studies conducted over other MTS topics. Tsay, Peña, and Pankratz (2000) tried to characterize and identify outliers. Frenzel and Pompe (Frenzel & Pompe, 2007) applied partial mutual information to analyze coupling between time series data sets. Amiri-Simkooei (2009) analyzed noise in multivariate GPS time-series.

In summary, previous studies on MTS focus on either finding patterns between different MTS data sets or within a single data set. However, for a set of MSTS, it may be interested in both finding temporal patterns within an MTS and finding those across space. Moreover, conducting further correlation analysis over such spatio-temporal patterns might be able to unveil more useful knowledge.

This page is intentionally left blank.

3 THE PROPOSED APPROACH

In this chapter, the problems of mining of spatio-temporal patterns in i) spatial trajectory database and ii) multivariate spatial time series database are defined. A new theoretical framework handling the mining of these data is proposed. The proposed approach consists of a collection of techniques for 1) discovering statistically significant patterns from spatial trajectory database and multivariate spatial time series database automatically; 2) using the discovered patterns from i) spatial trajectory database and ii) multivariate spatial time series database to construct a transformed relational database to represent the original database for further analysis; 3) applying a pattern discovery approach to a transformed database for clustering and classification. Based on this theoretical and systematic framework design, this chapter will describe how these techniques are integrated into the proposed approach.

3.1 A Formal Problem Description and Technical Preliminaries

Before we introduce the theoretical framework for pattern discovery for spatial trajectory data and multivariate spatial time series data, let us begin with some of the conventions, terminologies, and definitions that will be used within the entire thesis.

3.1.1 Notation and mining problems of spatial trajectory.

Let's assume a set of trips in a moving object database D that captures the movement of multiple objects over a lengthy period of time. Each object is tracked as a set of trips. A trip or a trajectory T is a trace created by a moving object in geographical space over a period of time $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_i \rightarrow \cdots \rightarrow p_n$. We consider both terms, trip and trajectory, as interchangeable hereafter. Each trip can be identified by a trip ID and can be labelled with a class label. A formal definition and the relationship of these notations will be presented in section 4.2 in details.

3.1.1.1 Pattern discovery from features of spatial trajectory.

Before classifying spatial trajectories, a trajectory can be transformed into several continuous and discrete features in order to effectively extract spatial and temporal information to characterize the trajectory. In order to pre-process the data for the efficient and effective use of pattern discovery techniques, the continuous values should be properly discretized to categorical values. In pattern discovery, the goal is to identify interesting patterns with n different orders (number of the variables of attributes they span).

3.1.1.2 Classification of spatial trajectory. Now each trip is associated with a class label so in supervised learning, this can be treated as a classification problem of identifying to which of a set of classes $|C|$ a new trip T_i belongs, on the basis of a training set of moving object database D containing trips whose class is known. Therefore, the formulation of the classification problem is for each moving object, we train a classifier using its own trips and then predict whether or not a trip, which can be from its own or from other moving objects, belongs to this moving object.

3.1.1.3 Clustering of spatial trajectory. We also argue that in some scenarios, the class label C is unknown in the entire moving object database D so in unsupervised learning, this can be treated as a clustering problem of grouping a set of trips in such a way that trips in the same group are more similar to each other than to those in other group. This is to partition these trips, $T_1, \dots, T_i, \dots, T_{|D|}$, into k clusters:

$$cluster = \{cluster_1, \dots, cluster_k\}$$

according to the similarities of these trips. The proposed system is able to deal with both scenarios whereas class information is available or unavailable.

3.1.2 Notation and mining problems in multivariate spatial time series. We are also concerned with mining a set of *Multivariate Spatial Time Series* (MSTS) data to reveal patterns in both time and space domain. Suppose there are multiple spatial locations $L = \{l_1, \dots, l_{|L|}\}$, each of which is represented by a region label and its set of geographic coordinates g so that $L = \{(r_1, g_1), \dots, (r_{|L|}, g_{|L|})\}$ where each set of geographic coordinates g contains longitude x and latitude y coordinates. $|L|$ is the number of locations in the study area. There are totally N distinct regions $R = \{r_1, \dots, r_i, \dots, r_N\}$ which partition the locations of the study area. $G(x, y)$ is a function to retrieve the region label r_i for a given longitude x and latitude y. To represent neighborhood between regions, let W be an adjacency matrix that assigns equal weights to all neighbors of regions, that is, $\{W\}_{i,j} = 1$ if region r_i and r_j share a common border or 0 otherwise. For each location, there exists at least 1 MSTS. A MSTS is associated with a longitude x and latitude y. With $G(x, y)$, a MSTS can be converted to a MTS associated with a region label $r_i, i \in \{1 \dots N\}$. A MTS consists of m individual time series $TS = \{1, \dots, m\}$. A time series TS is a finite sequence of real values $(v_1, v_2, \dots, v_n)$ containing n observations with unique time points $TP = \{1, \dots, n\}$. A symbol sequence S is a sequence of characters $s_1, s_2, \dots, s_n$ over an alphabet set ε, where each $s_i \in \varepsilon$. ε is a set of distinct characters with size $|\varepsilon|$. n is the length of S. $S[i, j]$ is its substring from index i to j. Each character represents an event so S can be called an event sequence. After discretization, a TS can be transformed into a symbol sequence S. SAX (Lin, Keogh, Wei & Lonardi, 2007), a well-known discretization method for time series data mining practitioners, is adopted here for discretization. Therefore, a MTS can be transformed into a set of multiple symbol sequences $S_1, S_2, \dots, S_m$. A pattern P is a short sequence of consecutive characters $p_1, p_2, \dots, p_{|P|}$ over ε where $|P|$ is the length of the pattern. A pattern P is always associated with a symbol sequence S. P occurs in an interval $[i, j]$ in S if and only if $P = S[i, j]$. o_P denotes the occurrence of P. All occurrences of P are recorded in its occurrence list L_P so $|L_P|$ is the number of occurrences of P in S. A frequent pattern is a pattern with its number of occurrences $|L_P| > min_o$ where min_o specifies the minium number of occurrences required.

3.1.2.1 Temporal association of frequent patterns. A temporal association pattern TP is an association of frequent patterns occurring sequentially in time. Each pattern P^i is a block of a TP. It implies P^{i+1} occurs within a certain specified time delay t_d after P^i occurs for $i = 1, \ldots, LT - 1$. There are totally LT blocks for a TP and we call LT level of TP. max_{LT} specifies the maximum level of TP. When all frequent patterns of a TP are from the same sequence, TP is called an auto association pattern or intra pattern. Otherwise, it is called cross association pattern or inter pattern. $|TP|$ is total number of temporal association patterns.

3.1.2.2 Spatial association of temporal patterns. A spatial association of temporal pattern, a.k.a spatio-temporal pattern, SP is an association of multiple temporal association patterns co-occurring in multiple regions. Each TP^i is a building block of SP. It implies that TP^{i+1} occurs in a region r_j other than the region r_i of TP^i where $r_i \neq r_j$. There should be at least 2 TPs, i.e. 2 blocks, in a SP. Otherwise, every TP is a SP. There are totally LS blocks for SP and we call LS the level of SP. max_{LS} specifies the maximum level of SP. $|SP|$ is the total number of spatial association patterns.

3.2 The Solution

Given a data set of spatial trajectory and/or multivariate spatial time series, we propose to use a new data mining approach for the discovery of patterns. To solve the problems of i) classification and clustering of spatial trajectory / multivariate spatial time series data, and ii) association discovery of spatio-temporal patterns for them, the proposed approach comprises of a collection of techniques for multiple phases: 1) feature generation and discretization on spatial trajectories as pre-processing, 2) frequent pattern mining and temporal association pattern discovery on MSTS in order to obtain temporal pattern sets as pre-processing, 3) interesting association pattern discovery from transformed data, 4) clustering and re-clustering to summarize information, and 5) classification to describe important classes or to predict class labels. Figure 1 shows the general framework for the proposed data mining approach and how these techniques are integrated to form the data mining system.

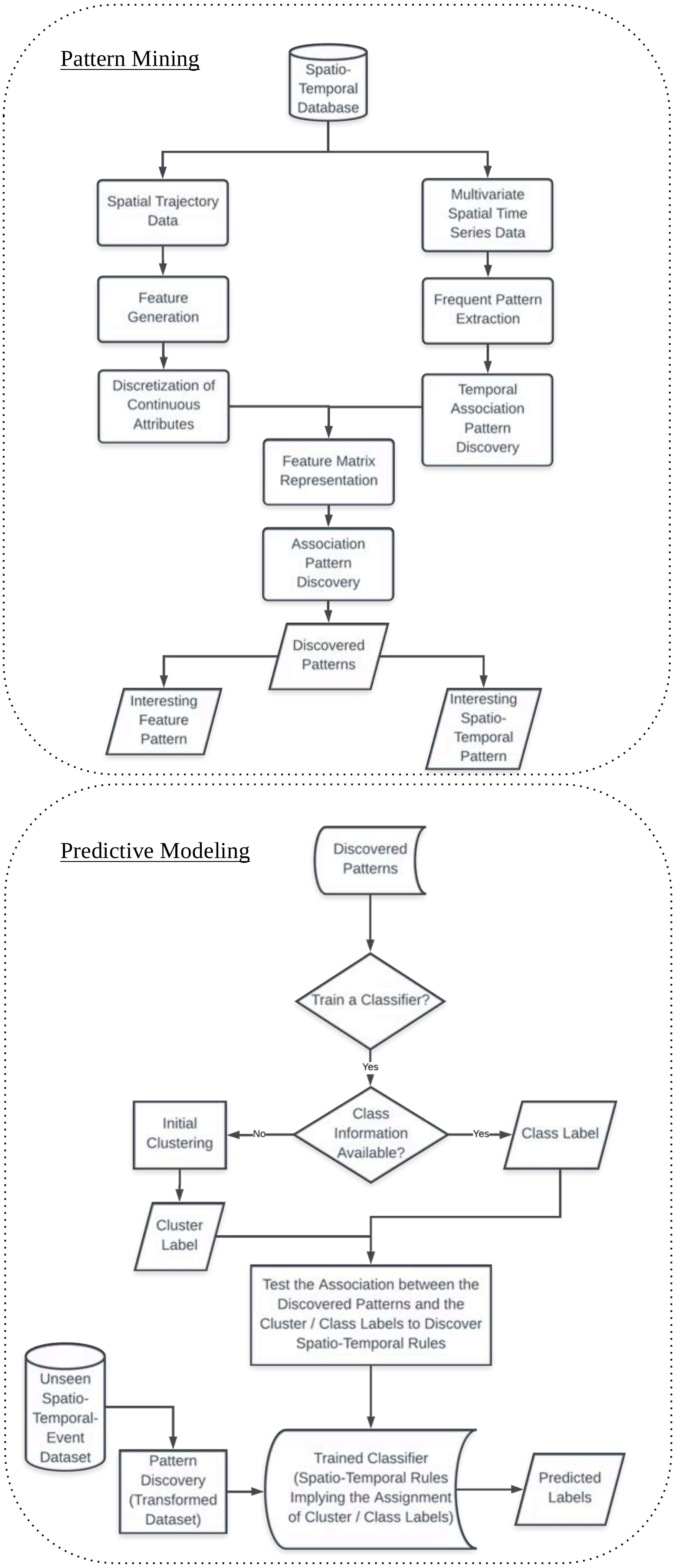

Pattern Mining
Spatio-Temporal Database
Spatial Trajectory Data
Multivariate Spatial Time Series Data
Feature Generation
Frequent Pattern Extraction
Discretization of Continuous Attributes
Temporal Association Pattern Discovery
Feature Matrix Representation
Association Pattern Discovery
Discovered Patterns
Interesting Feature Pattern
Interesting Spatio-Temporal Pattern
Predictive Modeling
Discovered Patterns
Train a Classifier?
Yes
Class Information Available?
No
Yes
Initial Clustering
Class Label
Cluster Label
Test the Association between the Discovered Patterns and the Cluster / Class Labels to Discover Spatio-Temporal Rules
Unseen Spatio-Temporal-Event Dataset
Pattern Discovery (Transformed Dataset)
Trained Classifier (Spatio-Temporal Rules Implying the Assignment of Cluster / Class Labels)
Predicted Labels

Figure 1. The General Framework of The Proposed Data Mining Approach.

The proposed data mining approach is comprised of two systems, namely pattern mining and predictive modeling. It is able to deal with two spatio-temporal data types – spatial trajectory and multivariate spatial time series. In the pattern mining system, the algorithm firstly performs feature engineering to pre-process the data. For spatial trajectory data, it generates low-level features from each trip and discretizes the feature values of continuous type into categorical values. These features characterize the trajectories and retain the important spatial and temporal information. After the discretization, the transformed feature matrix represents the original spatial trajectory data. For MSTS, after multiple time series of a region are discretized into time sequences, frequent sequential patterns based on a threshold value are extracted for the discovery of temporal association patterns. The discovered temporal patterns associated with a statistical significance value are treated as the features to characterize the region. This transformed feature matrix represents the original MSTS data. The next phase is an association discovery process on the transformed feature matrix based on a statistical significance test to search for interesting spatio-temporal patterns with different orders inherent in the data.

In the predictive modeling system, the algorithm takes the discovered patterns as rules to build a classifier based on an information theoretic measure to detect the association between these discovered patterns and the class label. If the original data set is unlabeled, an initial clustering phase using the-state-of-the-art algorithm will be performed to generate the cluster labels that will be treated as the class labels for building the classifier in the re-clustering phase. Once the classification model is built, to deploy and use it, when unseen or new data of spatial trajectory type or multivariate spatial time series type are put into the system, the classifier is able to automatically predict the class of them.

This page is intentionally left blank.

4 Pattern Discovery for Spatial Trajectory

4.1 Background

In this chapter, a series of algorithms that constitute a pattern-mining system for identifying interesting patterns, classification and clustering in spatial trajectory is proposed. A good pattern-mining system for spatial trajectory should not store all the exact location information of users but should extract useful patterns for predictive modeling and this has not been widely discussed in the literature. Our study proposes a pattern discovery approach to extract interesting driving patterns to characterize the driving trip data set for further classification analysis in a supervised learning setting. We perform experiments on a number of real data sets of spatial trajectories to compare the classification accuracy to show the effectiveness of the representation of the driving data. We also investigate the clustering of spatial trajectory data using a transformed feature set in an unsupervised learning setting. We develop algorithms and evaluate the performance by comparing the proposed techniques with different traditional clustering techniques. The major contribution of this approach includes 1) we define the characteristics of labeled and unlabeled trajectory data with its associated attributes (features) and the classification and clustering problem of it. 2) We propose a general model for classifying and clustering feature-based trajectories. 3) We incorporate the pattern discovery approach to discover statistically significant patterns for feature-based trajectories. 4) We apply the proposed algorithms and pattern discovery approach to evaluate and compare the results of different traditional clustering methods on a real-time ridesharing data set. To demonstrate the effectiveness of the representation of the trajectory data, we perform experiments on a number of real data sets of spatial trajectories, which include a synthetic data set, real physical exercise data set, GeoLife data set and a case study on a driver telematics data set, to compare the classification accuracy using famous classifiers such as C4.5 decision tree, random forest, logistic regression, support vector machine and convolutional neural network.

In this section, we introduced the background of pattern discovery for spatial trajectory. The rest of this chapter will examine methodically in detail the process for the proposed system. Section 4.2 describes the technical preliminaries that will restate the notations and definition of spatial trajectory that are used in the proposed algorithms. Section 4.3 to section 4.6 describes and explains each step of the proposed pattern mining system for spatial trajectory with relevant mathematical notations to formally define the solution methods. Section 4.7 discusses the computational complexity of the proposed algorithm. Section 4.8 reports the experimental results obtained from both synthetic data set and real-world data set. This chapter ends with a summary in section 4.9.

4.2 Technical Preliminaries

Given a moving object database D that captures the movement of multiple objects over a lengthy period of time. Each object is tracked as a set of trips. A trip or a trajectory T is a trace created by a moving object in geographical space over a period of time. The proposed algorithm models the spatial trajectory as trip T from the moving object database D, which has been briefly defined in Chapter 3. Let's formally define them as follows.

Definition 4.1. A trip T is a trajectory of a moving object represented by a set of time ordered points, e.g. $T: p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_i \rightarrow \cdots \rightarrow p_n$ where each point p consists of a geospatial coordinate set and a timestamp that is

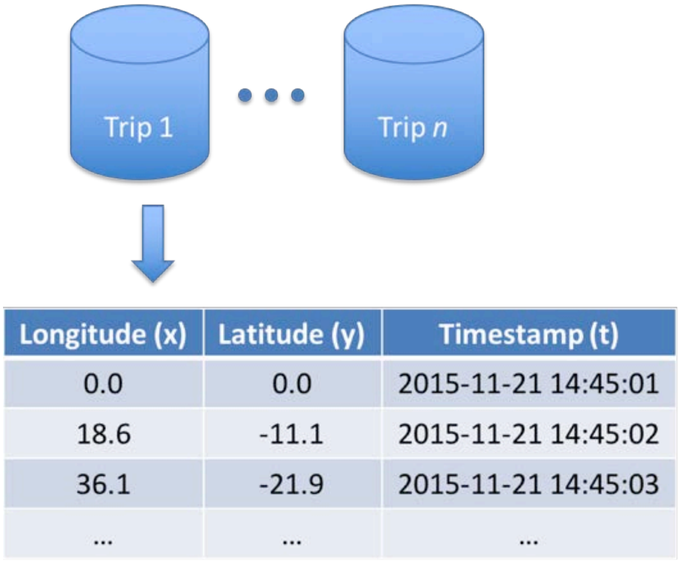
$p = (longitude, latitude, timestamp)$. Thus, T can be denoted by:

$T = \{(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_i, y_i, t_i), \dots, (x_n, y_n, t_n)\}$, such that $t_i < t_{i+1}$ for all $i \epsilon \{1, \dots, n\}$ and each (x_i, y_i, t_i) is the object's location in longitude and latitude at time t_i. A special property of T is (x_1, y_1) = $(0, 0)$ as during data collection process, all trips are all centered to start at (0, 0) with the direction randomly rotated from the start of the trip for trajectory anonymization purpose. To emphasize, this special property is to preserve the privacy of the trips and moving objects. We also assume that the trip data are equally spaced, so the points are sampled in a regular way of the same time length.

Definition 4.2. Let D be a set of trips in a moving object database. The cardinality of D, $|D|$, is the number of trips. Thus, D can be denoted by:

$D = \{(T_1, c_1), (T_2, c_2), \dots, (T_i, c_i), \dots, (T_{|D|}, c_{|C|})\}$. Each trip T in D has a unique trip ID and is associated with a class label $C \in \{c_1, c_2, \dots, c_{|C|}\}$ (a class label is also known as moving object ID who is used to identify the unique object), and we assume that there are totally $|C|$ distinct class labels in D. To clarify the definition of class label a.k.a moving object ID, the number of classes of the database will be equal to the number of moving objects. A moving object as identified by its moving object ID may have many trips. There exists a one-to-many relationship between moving objects and trips.

Figure 2 demonstrates an example of a trip in a moving object database. Let's assume in Figure 2 the trips are associated to the same moving object, filtered by a class label, and are all centered to start at (0, 0) with the direction randomly rotated from the start of the trip for the reason of anonymousness. The timestamp value is for demonstration purpose only. This moving object database that stores various temporally ordered trips, which are measured in 2-dimensional coordinates at each timestamp, of multiple moving objects can be seen as the database about moving 'points', which shall well represent one kind of spatio-temporal databases.



Longitude (x)	Latitude (y)	Timestamp (t)
0.0	0.0	2015-11-21 14:45:01
18.6	-11.1	2015-11-21 14:45:02
36.1	-21.9	2015-11-21 14:45:03
...	...	...

Figure 2. Schematics of Trips in Moving Object Database.

The proposed system goes through the following steps.

- It firstly generates low-level features from each trip and discretizes the features of continuous type.
- Then, it discovers the statistically significant patterns from the discretized attributes data set and detects higher-order patterns from the lower ones. Based on the discovered patterns, it can construct the graph representation for efficient retrieval and effective visualization of patterns.
- If class information is available, a classifier can be trained using the discovered patterns.
- If class information is unavailable, a cluster model by a 2-step clustering can be trained.

4.3 Feature Generation and Discretization

In the proposed system, we assume that the locations of objects are recorded over a long history, so each moving object may contain many trips. Each trip is tracked as a set of GPS coordinates per time interval (i.e. second). Some might argue that the position of a moving object is sampled as point-based data and not as interval-based due to many reasons such as energy saving and communication loading (Zheng, 2015) but, however, it leads to an object's movement between sampling points uncertain. To avoid this, due to technological advances in sensor technology and communication bandwidth, the sampling rate is increased to a level that can neglect the uncertainty of the positions between two sampling points with high energy efficiency. Based on the information of space (positions) and time, we can describe each trip by 4 categories of spatio-temporal attributes as the features:

a) Route-related attributes

b) Speed-related attributes

c) Turning-related attributes

d) Stop points-related attributes

A full list of attributes is shown in Table 4.1 to describe the 4 categories. Notice that different spatial and temporal attributes can be generated based on the nature of the data source and the actual application. To demonstrate the practicality of the proposed feature generation method, we use a particular set of attributes to represent the characteristics of recorded GPS data, which shall be seen as a specific application of a more general approach.

Table 1

Full List of Extracted Attributes

Type	**Attribute**	**Description**	**Number of generated attributes**
a	Total distance traveled	The total distance that an object traveled in this trip.	1
a	Traveling duration	The total time duration that an object spent on this trip.	1
b	Speed distribution	The 0th, 10th…90th, 100th, percentile of the speed along the trip.	11
b	Acceleration	The 0th, 10th…90th, 100th, percentile of	11

Type	Attribute	Description	Number of generated attributes
	distribution	the acceleration along the trip.	
b	Derivative of acceleration distribution	The 0th, 10th…90th, 100th, percentile of the derivative of acceleration.	11
b	Total energy	The energy is assumed to be proportional to the absolute difference of squared velocity for adjacent time points. The total energy is the sum of all the energy spent at each time point.	1
c	Turning angle distribution	The 0th, 10th…90th, 100th, percentile of the turning angle along the trip.	11
c	Distribution of (turning angle * speed)	The 0th, 10th…90th, 100th, percentile of the (turning angle*speed) along the trip.	11
d	Number of stop points	The number of stop points along the trip.	1
d	Post-stop acceleration distribution	The 0th, 10th…90th, 100th, percentile of the acceleration while an object starts moving after each stop.	11
d	Pre-stop	The 0th, 10th…90th, 100th, percentile of	11

Type	Attribute	Description	Number of generated attributes
	deceleration distribution	the deceleration while an object tries to stop moving.	
Total number of generated attributes:			81

Let $A = \{a_1, a_2, \dots, a_m\}$ be the attribute set. For each trip, we calculate an 81-dimension vector (i.e. $m = 81$) as its attribute set to represent the feature of a trip. After the feature generation, the data set is represented by a feature matrix (**FM**) sized $N \times M$ of which each vector X_i, where $i = 1, \dots, N$, is then characterized by M continous attributes (Figure 3).

$$
\begin{array}{c|c|c|c|c|c|}
 & a_1 & \cdots & a_j & \cdots & a_m \\
\hline
X_1 & x_{11} & \cdots & x_{1j} & \cdots & x_{1m} \\
\hline
\vdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\hline
X_i & x_{i1} & \cdots & x_{ij} & \cdots & x_{im} \\
\hline
\vdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\hline
X_N & x_{N1} & \cdots & x_{Nj} & \cdots & x_{Nm} \\
\hline
\end{array}
$$

Figure 3. Feature Matrix Representation for Spatial Trajectory after Feature Generation.

With FM, a discretization method is applied to these vectors to discretize the continuous attribute values to categorical values, in order to pre-process the data for the efficient and effective use of pattern discovery techniques. There are several reasons to support the discretization. First, it greatly reduces the computational complexity for pattern discovery and later the classification task for which categorical data classification is more efficient. Second, the visualization of the patterns using categorical labels is more human readable and user-friendly, especially using graph representation to visualize the patterns. Next section will introduce the pattern representation and visualization. The experiment section will compare the classification accuracy with algorithms using continuous attributes without discretization. The result indicates that the accuracy will not be worsened significantly so the benefits of discretization here outweigh the drawback of data loss. Discretization rules can be predefined subject to the user's understanding of the data set. Each discretization rule takes one or more continuous type of attributes and outputs a categorical value. Therefore it can summarize the continuous attributes into categorical attributes. Table 2 demonstrates the sample discretization rules.

Table 2

Sample Rules for Discretization of Continuous Attribute Values

Rule	**Logic**
TripLength	If the trip length > median(all trips' length) and trip duration > median(all trips' duration) then return 1 else return 0.
MedianSpeed	If the median speed of the trip > median(the median speed of all trips) then return 1 else return 0.
SpeedyTurning	If the median of (speed * angle) > median(the median (speed * angle) of all trips) then return 1 else return 0.
PostStopAcc	If the maximum of post-stop acceleration > median(maximum of post-stop acceleration of all trips) then return 1 else return 0.
PreStopAcc	If the maximum of pre-stop deceleration > median(maximum of pre-stop deceleration of all trips) then return 1 else return 0.
StopPoints	If the number of stop points > median(number of stop points of all trips) then return 1 else return 0.

This set of discretization rules in Table 2 as an example expands the generated attributes in Table 1Table *1* to further characterize the trips. An alternative unsupervised discretization approach, namely Mixed-mode Attribute Clustering Algorithm (**MACA**), proposed by our previous studies (Wong, Wu, Wu, & Chan, 2010; Wu, Chan, & Wong, 2011) can also be applied to the generated continuous features if these features are highly correlated. Since class information is available, an additional discretization step can be applied using supervised discretization described in MACA and CAIM by Kurgan and Cios (2004). Discretization is beyond the scope of this study, so we will not further focus on it. After the discretization, the data set is represented by a transformed feature matrix (**TFM**) sized $N\times M$ of which each vector T_i, where $i = 1, \dots, N$, is then characterized by M categorical attributes and the pattern discovery technique can readily be applied.

4.4 Discovery of Interesting Patterns

In pattern mining, the main task is to generate interesting patterns with n different orders (number of the variables of attributes they span). After feature generation and discretization, we discover interesting patterns from the transformed feature matrix (**TFM**) produced in section 4.3. Pattern discovery starts by searching the second order patterns ($n = 2$) from the first order patterns and statistically significant patterns (interesting patterns) will be retained to search for third order patterns and so on. Detail mathematical proof of the methods can be found in (Chan & Wong, 1990; Wong & Wang, 1997). The discovery process detects patterns by a statistical significance test defined in definition 4.3 based on adjusted residual. Let’s consider a simple real-world example to illustrate the discovered interesting patterns based on statistical significance. The XOR, pronounced as Exclusive OR, problem is a digital logic gate that gives a true output when the number of true inputs is odd. It involves 3 binary variables, X, Y, and, $Z = X \oplus Y$, i.e. Z is true when either X or Y, but not both, is true. For a real world situation, when there is a narrow bridge in a road with only a single lane, and two vehicles X and Y in the opposite sides. The result is a vehicle passing through the bridge Z. If X does not try to cross $\{X = F\}$ and Y does not try to cross $\{X = F\}$, there is not car crossing at all $\{Z = F\}$. If both try to cross at the same time $\{X = T, Y = T\}$, then there will not be possible for any car crossing at all $\{Z = F\}$. Only if one car crosses and the other remains still $\{X = T, Y = F\}$ or $\{X = F, Y = T\}$, then it can be a car crossing $\{Z = T\}$. In this example, we assume that without domain knowledge, nobody knows it is the XOR problem. We would like to discover interesting patterns to see whether or not the occurrence of the association pattern, or simply pattern, $\{X = T, Y = T, Z = F\}$ is just a random happening. If the observed frequency of this pattern deviates significantly from the random assumption, we know this happening is not random given that we can estimate its frequency of occurrences under the random assumption. This happening is referred to as an interesting pattern in the statistical sense. To illustrate, considering an XOR database contains 10,000 samples in which each value of a variable, i.e. $\{X = T\}$, occurs 5,000 times. The expected frequency of occurrences of the pattern $\{X = T, Y = T, Z = F\}$ under the independence assumption is 50% x 50% x 50% x 10,000 = 1,250. Let’s say its observed frequency of occurrences is 2,500, we are going to detect whether or not the difference between the observed frequency of occurrences and expected frequency of occurrences (i.e. 2,500 – 1,250) is significant enough to conclude that the pattern is not a random happening. To test this, we introduce to apply a statistical significance test based on adjusted residual analysis. Given a significant level, if the statistical significance test is passed, then the observed frequency of the pattern is signficantly greater than the expected frequency, thus not a random happening.

Definition 4.3 Let *ST* be a statistical significance test. If the frequency of occurrences of a pattern P_n is significantly deviated from its expectation based on a default probabilistic model, we say that P_n is a statistically significant pattern, or an interesting pattern of order n.

Let us denote the observed occurrences of pattern as o_{P_n} and its expected occurrences as e_{P_n}. e_{P_n} is computed by:

$$e_{P_n} = |D| \prod_{i \in n, P_i \in P_n} P(P_i) \tag{4.1}$$

where $P(P_i)$ is estimated by the proportion of the occurrence of P_i to the sample size $|D|$, which is the number of trips.

To test whether or not P_n is a statistically significant pattern, standardized residual z_{P_n} defined in (Haberman, 1974) is used to measure the deviation between o_{P_n} and e_{P_n}:

$$z_{P_n} = \frac{o_{P_n} - e_{P_n}}{\sqrt{e_{P_n}}} \tag{4.2}$$

where z_{P_n} is considered to be of normal distribution only when the asymptotic variance of z_{P_n} is close to one. Otherwise, it has to be adjusted by its variance (Wong & Wang, 1997).

To normalize z_{P_n} for a more precise analysis, the adjusted residual d_{P_n} is defined as:

$$d_{P_n} = \frac{z_{P_n}}{\sqrt{v_{P_n}}} \tag{4.3}$$

where v_{P_n} is the maximum likelihood estimate of the variance of z_{P_n}.

For a pattern P_n of a pattern candidate set PC_n, we construct the contingency table to count the occurrences of it and compute the adjusted residual. Based on 5% significant level, if the adjusted residual is greater than 1.96, which is the predefined minimum threshold, then observed frequency of the pattern is significantly greater than the expected frequency. The significant level is a user input parameter. Some common settings include 1%, 5% and 10%, with significance threshold values 2.575, 1.96 and 1.645 respectively. The higher the adjusted residual value indicates the pattern is more deviated from expectation. The default significant level we adopt in our proposed pattern mining algorithm is 5% whose significance threshold value is 1.96 by convention. In this case, P_n is referred to as a statistically significant pattern.

The first order patterns are all composed of discretized attributes and are stored in pattern candidate set PC_1. To discover the 2nd order patterns for each object, we construct a pattern candidate set PC_2, which contains all the combinations of the first order patterns. For each combination, we perform the statistical significance test described above to search for a set of statistically significant 2nd order patterns, P_2.

To search for higher-order patterns P_{n+1} and avoid the exhaustive search, we construct the next candidate set PC_{n+1} based on only statistically significant patterns in one order lower and apply the statistical significance test to filter out insignificant patterns in PC_{n+1}. A user input parameter $order_{max}$ is required for limiting the max order for the search operation. The operation terminates after $(n + 1) = order_{max}$ iteration. Thus, the method is efficient since the search space for the next order of patterns will be greatly reduced by previous iterations. Combining both steps, we give the pseudo-code as Figure 4.

```
Input:
  $D = \{(T_1, c_1), \dots, (T_{|D|}, c_{|C|})\}$ (original database)
  $order_{max}$ (max order for detecting patterns)
  $sig$ (significance threshold value)
Output:
  $P$ (set of discovered patterns)
Variables:
  $A = \{a_1, a_2, \dots, a_m\}$ (attribute set)
  $d_{P_n}$ (adjusted residual of pattern $P_n$)
  $PC_n$ (pattern candidate set of patterns of order $n$)
Algorithm:
  $P = \emptyset$
  For each trip $T_i \in D$
    transform it into a set of attributes $A$
  For iterator $n$ = 2: $order_{max}$
    If $n = 2$
      initialize $PC_n$ based on all possible combination of $A$
    Else
      initialize $PC_n$ based on all possible combination of $PC_{n-1}$
    For each pattern $P_n$ in $PC_n$
      calculate $d_{P_n}$
      If $d_{P_n} > sig$
        insert $P_n$ into $P$ ($P_n$ is statistically significant)
      Else
        remove $P_n$ from $PC_n$ ($P_n$ is not statistically significant)
    End
  End
  Return $P$
```

Figure 4. The Pseudo-Code of the Proposed Pattern Discovery Algorithm.

In order to represent and manipulate the interesting patterns efficiently, we need a simple but powerful pattern representation structure to encode the discovered patterns. Attributed hypergraph (AHG) representation (AHR) proposed by (Wang & Wong, 1996) is a lucid structure and general enough to encode different order patterns. It provides a simple and direct transformation from pattern retrieval to graph manipulation where a number of mature graph algorithms could be adopted. In this paper, we use an attributed hypergraph to represent interesting patterns on a spatio-temporal data set for the sake of visualization, efficiency, and implementation. The below definition 4.4 to 4.8 defines the notation of this representation. To clearly explain and illustrate definition 4 to 8, Figure 5 which is also the top 3 discovered patterns from the case study visualizes the attributed hypergraph representation (AHR).

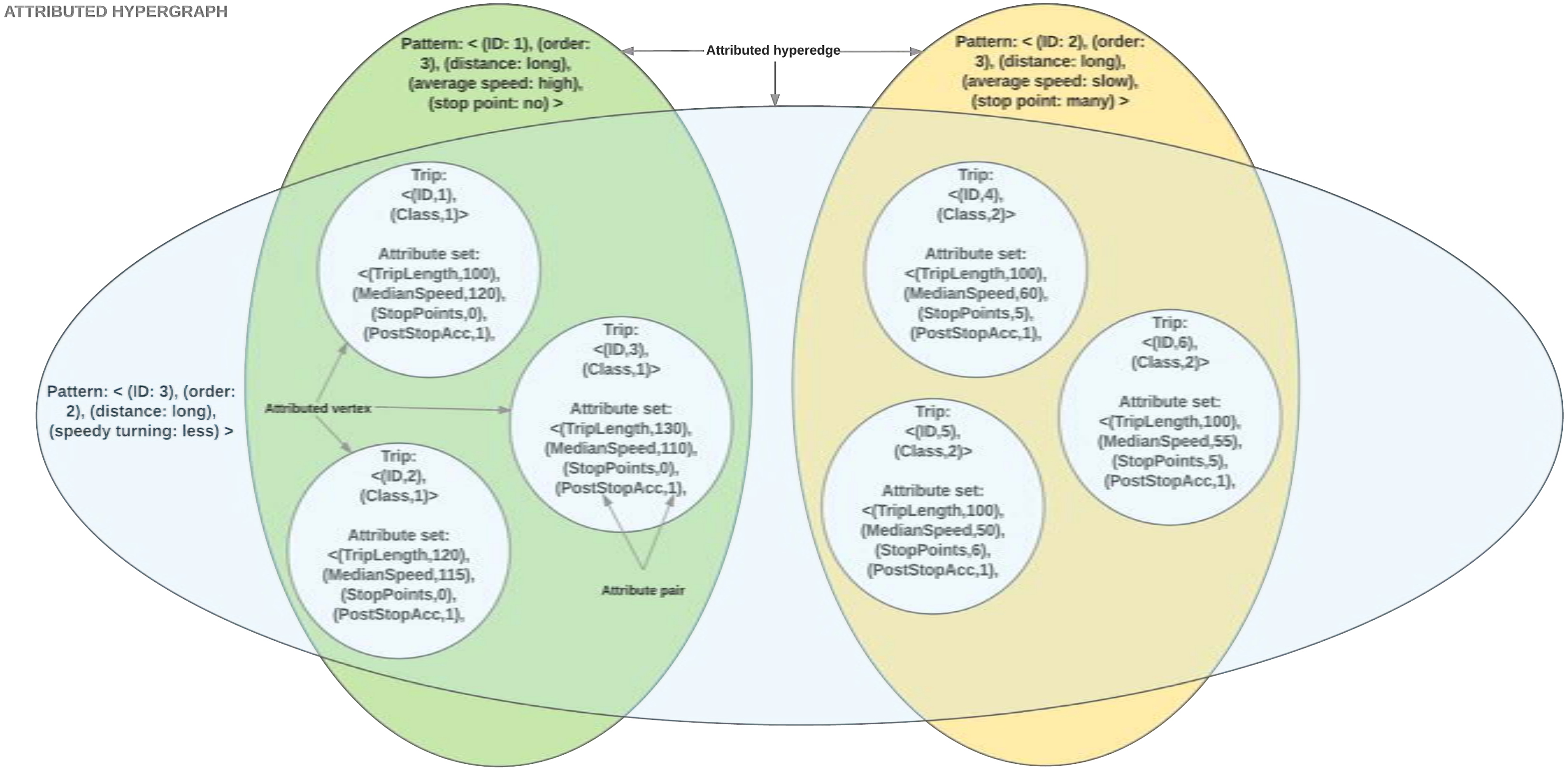

Pattern: < (ID: 1), (order: 3), (distance: long), (average speed: high), (stop point: no) >
Attributed hyperedge
Pattern: < (ID: 2), (order: 3), (distance: long), (average speed: slow), (stop point: many) >
Trip: <(ID,1), (Class,1)>
Attribute set: <(TripLength,100), (MedianSpeed,120), (StopPoints,0), (PostStopAcc,1),
Trip: <(ID,3), (Class,1)>
Attribute set: <(TripLength,130), (MedianSpeed,110), (StopPoints,0), (PostStopAcc,1),
Trip: <(ID,2), (Class,1)>
Attribute set: <(TripLength,120), (MedianSpeed,115), (StopPoints,0), (PostStopAcc,1),
Pattern: < (ID: 3), (order: 2), (distance: long), (speedy turning: less) >
Attributed vertex
Attribute pair
Trip: <(ID,4), (Class,2)>
Attribute set: <(TripLength,100), (MedianSpeed,60), (StopPoints,5), (PostStopAcc,1),
Trip: <(ID,6), (Class,2)>
Attribute set: <(TripLength,100), (MedianSpeed,55), (StopPoints,5), (PostStopAcc,1),
Trip: <(ID,5), (Class,2)>
Attribute set: <(TripLength,100), (MedianSpeed,50), (StopPoints,6), (PostStopAcc,1),

Figure 5. Illustration of Attributed Hypergraph Representation.

Definition 4.4 An attribute pair (A_k, A_v) is an ordered pair where A_k is the attribute name and A_v is the attribute value.

For example, an attribute pair can describe the total distance traveled on a trip. The attribute name A_k can be assigned to "total distance travelled" and the A_v is the measure of the total distance, i.e. $(\text{total distance}, 100)$.

Definition 4.5 An attribute set is an m tuple $< a_1,\ a_2,\ a_3, \dots, a_i, \dots, a_m >$ where each element is an attribute pair.

In the spatio-temporal context, the attribute set is used to describe the properties of a trip or relation between two trips. For example, an attribute set for describing a trip can be:

$$< (\text{total distance}, 100), (\text{time duration}, 600), (\text{number of stops}, 10) >$$

Definition 4.6 A pattern P_n is a realization of attribute values on an attribute set. The order of the pattern is the number of tuples n in the attribute set.

Definition 4.7 An attributed vertex is a vertex with an associated attribute set. An attributed hyperedge e is a set of attributed vertices, associated with a pattern.

In the proposed system, each trip is represented by an attributed vertex with discretized attributes as its attribute set. The relation among a group of trips is represented by an attributed hyperedge. For example, trip 1, trip 2 and trip 3 all contain the same 2nd order pattern

$$< (\text{distance: long}), (\text{average speed: high}) >$$

Then the hyperedge connecting them should be

$$e = \{\mathrm{trip1}, \mathrm{trip2}, \mathrm{trip3}\}$$

with attribute set

$$< (\mathrm{distance\colon long}), (\mathrm{average\ speed\colon high}) >.$$

Definition 4.8 An attributed hypergraph is an ordered pair $H = (V, E)$ where $V = \{v_1, v_2, \dots, v_n\}$ is a set of attributed vertices and $E = \{e_1, e_2, \dots, e_m\}$ is a set of attributed hyperedges such that $e_i \neq \emptyset\ (i = 1, 2, 3 \dots m)$, and $\bigcup_{i=1}^{m} e_i = V$.

Using the results of feature generation and discretization stage, we can construct the attributed vertices each of which represents a trip described by an attribute set. Based on the definition of attributed hyperedge, each element in $\bigcup_{i=2}^{n} P_i$ defines an attributed hyperedge. We assign each attributed vertex to the hyperedge if the attributes of the attributed vertex satisfy the corresponding patterns of the attributed hyperedge.

After we construct all the attributed hyperedges E and attributed vertices V, then the attributed hypergraph is generated, denoted as $H(V, E)$**.** In the spatio-temporal data context, if the attributed graph is generated using only the trips of the same moving object, this is treated as the moving object signature of this object.

4.5 Classification for Spatial Trajectory

Once the interesting patterns are generated in section 4.4, these patterns can be used for further analysis. If class information is available, we can train a classification model based on the interesting patterns discovered from the transformed feature matrix (**TFM**) produced in section 4.3. Let p^i be an interesting pattern discovered from class i, $\mathrm{c_i}$. In a supervised manner, if the interesting pattern p^i is conditioned by the class label $\mathrm{c_i}$, it can be treated as a classification rule (Wang & Wong, 2003), i.e. if {antecedent or left-hand-side or LHS} then {consequent or right-hand-side or RHS}. The weight of evidence measure W in information theory (Wang & Wong, 2003) is used to quantify the evidence of the joined significant rules to support or against a certain class membership. An example rule for the moving object data set is if $\{$ distance $=$ long and average speed $=$ high $\}$ then $\{$ class $= 1\}$ with a weight of evidence of a certain value.

Definition 4.9 The weight of evidence measure provided by a pattern p^i for or against the classification of a trajectory T into class c_i is defined as:

$$\begin{aligned}
W^i\left(T\middle|p^i\right) &= W(T \in c_i \,/\, T \notin c_i \;|T \text{ is characterized by } p^i) \\
&= I\left(T \in c_i : T \text{ is characterized by } p^i\right) \\
&\quad - I\left(T \notin c_i : T \text{ is characterized by } p^i\right) \\
&= \log\frac{P(T \in c_i \mid T \text{ is characterized by } p^i)}{P(T \in c_i)} \\
&\quad - \log\frac{P(T \notin c_i \mid T \text{ is characterized by } p^i)}{P(T \notin c_i)} \\
&= \log\frac{P\left(T \text{ is characterized by } p^i\middle|\, T \in c_i\right)}{P(T \text{ is characterized by } p^i |\, T \notin c_i)}
\end{aligned} \tag{4.4}$$

where $I(\,)$ is the mutual information. It is positive if p^i provides positive evidence supporting T is classified to $\mathrm{c_i}$, otherwise, it is negative, or zero.

$P(\,T \text{ is characterized by } p^i \big|\, T \in c_i)$ is the probability that a trajectory T contains a pattern p^i given that T belongs to c_i. It is computed by counting the occurrence of trajectories in the database containing pattern $p^i$and belonging to class c_i divided by the number of trajectories belonging to class c_i.

$P(\,T \text{ is characterized by } p^i \big|\, T \notin c_i)$ is the probability that a trajectory T contains a pattern p^i given that T does not belong to c_i. It is computed by counting the occurrence of trajectories in the database containing pattern p^i and not belonging to class c_i divided by the number of trajectories not belonging to class c_i.

$W^i(T|p^i)$ can be interpreted as a measure of the difference in the gain in information when a trajectory T containing p^i is classified into $\mathrm{c_i}$ as opposed to other classes. $W^i(T|p^i)$ is positive if p^i provides positive evidence supporting the classification of T into $\mathrm{c_i}$, otherwise, it is negative.

Given the interesting patterns $P = \{p_1^1, \dots, p_j^i, \dots,\ p_{m_i}^{|C|}\}$, discovered for each corresponding $|C|$ classes, $\mathrm{c_1}, \dots,\ \mathrm{c_i}, \dots,\ \mathrm{c_{|C|}}$, an unseen trajectory T_u can be classified by matching it against the patterns in each of classes. An unseen trajectory T_u is first transformed to a list of attributes using the proposed feature generation and discretization approach to construct a set of patterns P^u for matching. Then for every pattern p_j^i that T_u matches, there is some evidence $W^i(T_u|p_j^i)$ provided by it for or against the classification of T_u into $\mathrm{c_i}$. Assuming that T_u matches with $n_i \leq m_i$ patterns in P of $\mathrm{c_i}$, we calculate a total weight of evidence measure for T_u to classified into $\mathrm{c_i}$.

Definition 4.10 The total weight of evidence provided by each of individual patterns is a measure for T_u to be classified into c_i and is defined as:

$$W^i(T_u)$$

$$= W(\, T_u \in c_i / \, T_u \notin c_i \;\mid T_u \text{ is characterized by } p_1^i, \dots, p_j^i, \dots, p_{m_i}^i) \tag{4.5}$$

$$= \sum_{j=1}^{m_i} W(\; T_u \in c_i / \, T_u \notin c_i \;\mid T_u \text{ is characterized by } p_j^i)$$

The task of classification is to maximize $W^i(T_u)$. The major steps are given in Figure 6. The total weight of evidence for T_u to be classified into each of $c_1, c_2, \dots, c_{|C|}$ is computed and T_u is assigned to the class that can give the highest total weight of evidence. This measure is able to differentiate the case that when some identical trajectories refer to different classes in the training set as the class assignment of T_u is by the highest total weight of evidence.

```
Input:
  $T_u$ (an unseen trajectory)
Output:
  $c_u$ (assigned class of $T_u$)
Variables:
  $D = \{(T_1, c_1), \dots, (T_{|D|}, c_{|C|})\}$ (original database)
  $P = \{p_1^1, \dots, p_j^i, \dots, p_{m_i}^{|C|}\}$ (set of discovered patterns)
  $C = \{c_1, c_2, \dots, c_{|C|}\}$ (set of classes)
  $W^i(T|p^i), i = 1, \dots, m_i$ (set of weight of evidences)
  $P^u$ (set of patterns from $T_u$ for matching)
  $W^i(T_u)$ (set of total weight of evidences)
Algorithm:
  $P^u$ = transform $T_u$ into a set of patterns for matching
  For each discovered pattern $p_j^i \in P$
    For each pattern for matching $p_x \in P^u$
      If $p_j^i$ matches $p_x$
        For each class $c_i \in |C|$
          $W^i(T_u)$ = $W^i(T_u)$ + $W^i(T|p_j^i)$
        End
      End
    End
  End
  $c_u$ = class $c_i$ with max($W^i(T_u)$)
  Return $c_u$
```

Figure 6. The Pseudo-Code of The Proposed Classification Algorithm.

4.6 Clustering and Re-Clustering for Spatial Trajectory

If class information is not available, we can perform cluster analysis after transformed the raw spatial trajectories to the Feature Matrix (**FM**) and transformed feature matrix (**TFM**) in section 4.3. We treat FM as the input with each vector characterized by M continuous attributes as an object. The clustering approach consists of two steps, namely initial clustering and re-clustering. The initial clustering adopts the state-of-the-art clustering algorithm to assign cluster labels to the objects. It locally optimizes the clustering by extracting local information using a pair-wise distance measure between objects. A good clustering result should generate good cluster label, so we treat the cluster label as the class label to perform classification in order to globally partition the objects. The re-clustering step is essentially to apply a classification algorithm described in section 4.5 on TFM with the cluster labels treated as the class labels. It extracts global information through the discovery of interesting associations between objects and cluster labels.

In the initial clustering phase, given X objects, $x_1, \dots, x_i, \dots, x_N$, from FM we adopt a popular agglomerative hierarchical clustering algorithm (Sibson, 1973) to repeatedly group the most similar pair of clusters into a new cluster until forming a single cluster with all objects. This process prefers to leave uncertain objects ungrouped instead of forcing them into one of the cluster groups that make the discovered clusters less reliable. The output of this process is a dendrogram that displays the grouping results after each iteration of merging. The objects going to a specific branch in a dendrogram form a cluster. It is optional to apply a suitable cutoff level to obtain a specific number of clusters, i.e. $\{cluster_1, \dots, cluster_k\}$, based on prior knowledge. If there is no domain knowledge to specify the number of clusters or inspecting the dendrogram to see if it suggests a particular number of clusters is considered subjective, one can apply some heuristics to determine the optimal number of clusters. There are two major groups of methods namely direct methods and statistical testing methods. Direct methods such as average silhouette method optimize criteria, which measures the quality of a clustering, by running a clustering algorithm for different values of the number of clusters *k* and select the *k* with the best quality, i.e. highest average silhouette value. Average silhouette method by Kaufman and Rousseeuw (1990) computes the average silhouette of observations for different values of *k*. The optimal number of clusters *k* is the one that maximize the average silhouette over a range of possible values for *k*. Statistical testing methods are made up of comparing evidence against the null hypothesis. One of the methods in this group is gap statistic by Tibshirani, Walther, and Hastie (2001). The gap statistic compares the total within intra-cluster variation for different values of *k* with their expected values under the null reference distribution of the data. The estimate of the optimal clusters will be a value that maximizes the gap statistic (i.e. that yields the largest gap statistic). In addition to the above methods, there are more than 30 other methods that have been studied to determine the optimal number of clusters. Charrad, Ghazzali, Boiteau, and Niknafs (2012) have published and provided software package, NbClust, in R that implements these methods for this problem and cluster validity. However, determining the optimal number of clusters is beyond the scope of this thesis so we will not discuss it in further. As mentioned above, leaving uncertain objects ungrouped instead of forcing them into one of the cluster groups that make the discovered clusters less reliable produces a set of reliable initial clusters. Pearson correlation coefficient is used for the distance measure rather than Euclidean distance as it is known to be better in dealing with noise (Ma, Chan & Chiu, 2005). For a pair of objects X_i and X_j with values of M continuous attributes in FM, the similarity measure is defined as:

$$Sim(X_i, X_j) = \frac{M\sum_{k=1}^{M} d_{ik}d_{jk} - \sum_{k=1}^{M} d_{ik}\sum_{k=1}^{M} d_{jk}}{\sqrt{M\sum_{k=1}^{M} {d_{ik}}^2 - (\bar{d_\imath})^2}\sqrt{M\sum_{k=1}^{M} {d_{jk}}^2 - (\bar{d_\jmath})^2}} \qquad (4.6)$$

In re-clustering phase, objects that are not assigned to any cluster in the initial clustering phase will be assigned and those that have assigned will be re-evaluated to decide whether or not they should be re-assigned to a different cluster. Treating the assigned cluster label in the initial clustering as the class label, we can apply the classification algorithm described in section 4.5 using the objects that have assigned to clusters from TFM to train a classifier. Let X' be a set of objects that are not assigned to any cluster in the initial clustering phase, and X'' be a set of objects that have assigned to clusters in the initial clustering phase. The union of X' and X'' is X. To assign a class label to X' and to re-evaluate X'', the trained classifier will be used to predict the cluster labels for them. Figure 7 provides the details for this re-clustering step.

```
Input:
  $X_u$ ($X_u \in \{X', X''\}$ an object from $X'$ and $X''$)
Output:
  $cluster_u$ (an assigned cluster labels to $X_u$)
Variables:
  $P = \{p_1^1, \dots, p_j^i, \dots, p_{m_i}^{|C|}\}$ (set of discovered patterns)
  $Cluster = \{cluseter_1, cluseter_2, \dots, cluseter_{|k|}\}$ (set of clusters)
  $W^i(X|p^i), i = 1, \dots, m_i$ (set of weights of evidence)
  $P^u$ (set of patterns from $X_u$ for matching)
  $W^i(X_u)$ (set of total weights of evidence)
Algorithm:
  $P^u$ = transform $X_u$ into a set of patterns for matching
  For each discovered pattern $p_j^i \in P$
    For each pattern for matching $p_x \in P^u$
      If $p_j^i$ matches $p_x$
        For each $cluster_i \in Cluster$
          $W^i(X_u)$ = $W^i(X_u)$ + $W^i(X|p_j^i)$
        End
      End
    End
  End
  $cluster_u$ = $cluster_i$ with max($W^i(X_u)$)
  Return $cluster_u$
```

Figure 7. The Pseudo-Code of Re-Clustering Spatial Trajectory.

4.7 Complexity Analysis

Let N be the number of trips in the database, M be the number of discretized attributes, Q be the number of discovered interesting patterns, L be the number of distinct class labels, and k be the number of distinct cluster labels.

Feature generation and discretization step loops over all the N trips to apply the rules, for which the computational complexity is $O(MN)$.

Discovery of interesting pattern step computes the contingency table for all $\binom{M}{2}$ pairs of discretized attributes, so the computational complexity is $O(M^2N)$. For $i > 2$, to generate i^{th} order patterns, the computational complexity is $O(M^iN)$. For higher order patterns, the candidate set is greatly reduced by previous iterations as the algorithm only considers growing statistically significant patterns, so all pattern candidate set should be much less than all possible combinations of i$^{\text{th}}$ order patterns, i.e. $|PC_i| < M^i$. If we predefine the number of discretized attributes, the overall computational complexity is linear in terms of N. Suppose there exist $|\bigcup_{i=2}^{n} P_i| = Q$ interesting patterns after the pattern discovery process, the computational complexity of constructing the attributed hypergraph covering N trips is $O(NQ)$.

Classification step calculates the weight of evidence of all individual interesting patterns supporting or refusing the classification of a trip into a class. Given Q interesting patterns and L discint class labels, it calculates the weights of evidence QL times. Each time of the calculation scans through the entire database of N trips. Consequently, it takes NQL operations to obtain all weights of evidence between each interesting pattern and each class. Hence, its computational complexity is $O(NQL\ )$. For the classification of an unseen trip, it requires the generation of M discretized attributes and match them against Q interesting patterns to look for the highest total weight of evidence to support the classification into a certain class. As a result, it takes MQL operations in total for which the computational complexity is $O(MQL\ )$.

Clustering and re-clustering step scans through the entire $N\times M$ feature matrix once in the initial clustering, so the computational complexity of calculating the pair-wise similarity is $O(MN^2)$. Given Q interesting patterns and k discint cluster labels, it calculates the weight of evidence measure kQ times. Each time scans through the entire database of N trips. Consequently, it takes kNQ operations to obtain all weights of evidence between each interesting pattern and each cluster. Hence, its computational complexity is $O(kNQ)$. The re-evaluation and assignment of ungroup objects requires the generation of M discretized attributes and match them against Q interesting patterns to look for the highest total weight of evidence among k discint cluster labels to support the assignment into a certain cluster. As a result, it takes kMQ operations in total for which the computational complexity is $O(kMQ)$.

4.8 Experimental Results

To evaluate the performance of the proposed algorithms on spatial trajectory data mining tasks, we carried out a number of experiments using synthetic data, real data and conducted a case study. In the first experiment, we embedded some association relationships in a synthetic data set and then tested whether or not the proposed algorithms were able to discover the patterns hidden in the underlying association relationships. In the second experiment, we classified a small-scale real data set of GPS tracks of our own physical exercise data to verify the effectiveness of the proposed algorithms. In the third experiment, a large-scale publicly available spatial trajectory data set was tested against the proposed algorithms and other the-state-of-the-art algorithms for the performance comparison. In the fourth experiment, we conducted a case study using a real driver telematics data set provided by an insurance company to reveal driving patterns and test whether or not the proposed algorithms were able to discriminate drivers based on the discovered patterns.

The proposed system for spatial trajectory data is implemented by realizing the corresponding methods in the above sections of this chapter. Fig. 3.1 shows the basic structure of the system for such data. The spatial trajectory pattern discovery system reads input data for each moving object from a given database, and then detects interesting association patterns for each trip of a moving object. After all interesting patterns are obtained, the system builds a classifier by calculating all weights of evidence provided by each of interesting patterns for or against the classification of a trip into a moving object. For simplicity, we, by default, applied equal frequency algorithm with the number of bins = 10 to guide the discretization of features/attributes and construct the patterns up to the 3rd order with predefined minimum adjusted residual threshold = 1.96 (i. e. based on 5% significance level), unless otherwise stated. The system is implemented in the programming language Python 3.6 with NumPy, a package for scientific computing, and Pandas, a software library for data manipulation. It will first generate a feature set for each trip and discretize the continuous values into discretized attributes according to the predefined rules as shown in Table 2 of section 4.3. The experiments are carried out on a personal computer (MacBook Pro) with 2.9 GHz Intel Core i5 processor and 16 Gb RAM running macOS version 10.13. Partial results described in this chapter have been published in Wu and Chan (2017, 2018a).

4.8.1 Synthetic data set.

In this experiment, we test the proposed algorithms for effectiveness when it is used to distinguish the physical exercise types using the discovered patterns from the simulated GPS tracks data. We generated 1,000 trips using a GPS generator that is publicly available (http://www.gpsies.com/). Some association relationships were embedded in the generated trips for our algorithms to test whether or not we can discover the patterns hidden in the underlying association relationships. The first type of physical exercises is cycling, and the second type is running with the following embedded association patterns:

IF {“max acceleration” = “high” and “average speed” = “high”} THEN {“type” = “cycling”}

IF {“distance” = “long” and “average cosine of turning angles” = “high”} THEN {“type” = “cycling”}

IF {“max acceleration” = “low” and “average speed” = “low”} THEN {“type” = “running”}

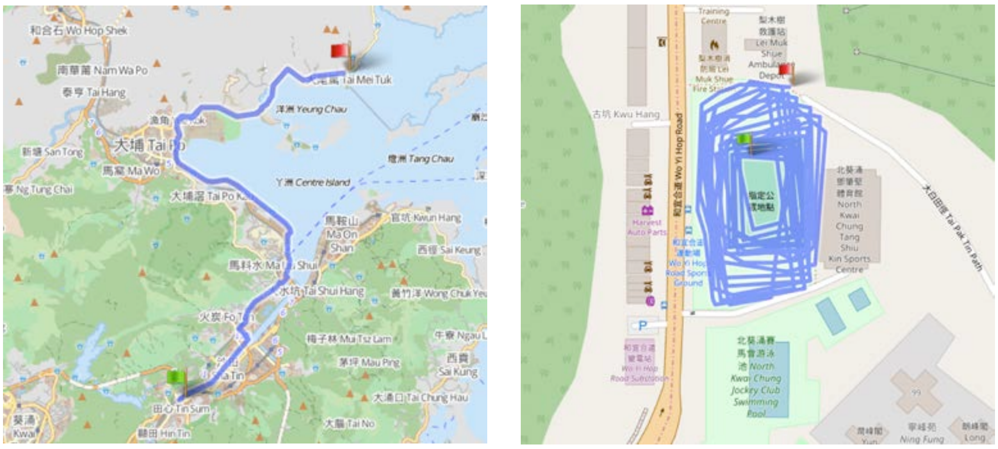
IF {“distance” = “short” and “average cosine of turning angles” = “low”} THEN {“type” = “running”}

The characteristics and routes of both types of physical exercises are shown in Table *3* and Figure 8 respectively. The cycling route is one of the famous cycling routes in Hong Kong, from Tai Wai to Tai Mei Tuk with cycling track along the Shing Mun River. The running route is in a standard oval running track with 400 meters length. Their spatial difference allows us to make sense of embedding the association patterns for the selected physical activities.

Table 3

The Characteristics of the Synthetic Data of GPS Tracks

Characteristics	**Cycling**	**Running**
Max acceleration	0.4m/s^2	0.25m/s^2
Average speed	25km/h	10km/h
Distance	23km	5km
Average cosine of turning angles	0.9949	0.8879



a) Cycling route

b) Running route

Figure 8. Routes of The Synthetic Data of GPS Tracks.

Each type of physical exercises contains 500 trips. To further examine the performance of the proposed algorithm in the presence of uncertainty, 10% of noise, which is 50 trips of each type of exercises, was added randomly to the data sets by removing 30% records of the latitude and longitude of the randomly chosen consecutive time intervals, i.e. increasing the average speed by 30%. We applied the proposed algorithms to generate the feature matrix (**FM**) from the synthetic GPS trajectory data and discretize the values of the FM for pattern discovery. As a result, the proposed algorithms discovered 454 interesting association patterns from the FM. The top 10 discovered patterns and rules together with adjusted residuals and weight of evidence are given in Table 4.

Table 4

Top 10 Discovered Patterns and Rules in Synthetic Data of GPS Tracks

IF {pattern} THEN {outcome}	**Adjusted residual**	**Weight of evidence**
IF {“30th percentile of acceleration” = “low” and “30th percentile of cosine of turning angle” = “low”} THEN {“type” = “running”}	9.036962	*infinity*
IF {“50th percentile of speed” = “low” and “70th percentile of acceleration” = “low”} THEN {“type” = “running”}	9.017009	*infinity*
IF {“50th percentile of acceleration” = “low” and “30th percentile of cosine of turning angle” = “low”} THEN {“type” = “running”}	8.502651	*infinity*
IF {“30th percentile of acceleration” = “low” and “average speed” = “low”} THEN {“type” = “running”}	7.951466	*infinity*

IF {“20th percentile of acceleration” = “high” and “average speed” = “100th percentile of acceleration derivative” = “high”} THEN {“type” = “cycling”}	7.534125	*infinity*
IF {“40th percentile of acceleration” = “low” and 50th percentile of acceleration” = “low”} THEN {“type” = “running”}	7.42295	*infinity*
IF {“90th percentile of cosine of turning angle” = “high” and “average speed” = “100th percentile of cosine of turning angle” = “high”} THEN {“type” = “cycling”}	7.289175	*infinity*
IF {“30th percentile of acceleration” = “low” and 40th percentile of acceleration” = “low”} THEN {“type” = “running”}	6.97137	*infinity*
IF {“70th percentile of speed” = “high” and “average speed” = “80th percentile of cosine of turning angle” = “high”} THEN {“type” = “cycling”}	6.970406	*infinity*
IF {“30th percentile of cosine of turning angle” = “low” and “100th percentile of cosine of turning angle” = “low”} THEN {“type” = “running”}	6.885303	*infinity*

For further experimentation, we used the transformed feature matrix to train classifiers, except deep learning which uses the raw trajectory data, to predict the type of physical exercise and compared their classification accuracy in Table 5.

10-fold cross validation over records is adopted so the synthetic data set is partitioned into 10 equal sized sub data set. Of the sub data sets, a single sub data set is treated as the testing set to validate the classification model and the remaining ones are treated as the training set to build the classifier. The cross-validation is repeated 10 times, with each of the 9 sub data sets used exactly once as the testing set. The 10 classification results are averaged to produce the final result. For comparison, we trained classifiers using the-state-of-the-art algorithms including C4.5 Decision Tree (Quinlan, 1993), Random Forest (Breiman, 2001), Logistic Regression (Le Cessie & Van Houwelingen, 1992), Support Vector Machine with Polynomial kernel (Platt, 1998) with the default parameters and deep learning based on Convolutional Neural Network (CNN) (Schmidhuber, 2015). Note that for CNN, we train the classifier using the raw trajectory data without feature engineering (i.e. no transformed feature matrix) as CNN can learn an internal representation of the trajectory data. The recent literature reported deep learning model based on CNN could have achieved comparable performance to models fit on a version of the data set with engineered features. Therefore, we implemented a one-dimensional convolutional neural network (1D CNN) model for comparison. Since CNN cannot handle time series of various length directly, padding zeros are inserted at the end of each time series. The network architecture is defined as having two 1D CNN layers, followed by a dropout layer for regularization, and then a pooling layer. The rectified linear unit (ReLU) function is chosen as the activation function that is a de-facto standard in recent deep learning models. Each type of activation function has its pros and cons (Gu, Wang, Kuen, Ma, Shahroudy, Shuai & Cai, 2015). It is common to define CNN layers in groups of two in order to give the model a good chance of learning features from the input data. Practically, CNNs learn very quickly, so the dropout layer is intended to help slow down the learning process and hopefully result in a better final model. The pooling layer reduces the learned features to 1/4 their size, consolidating them to only the most essential elements. After the CNN and pooling, the learned features are flattened to one long vector and pass through a fully connected layer before the output layer used to make a prediction. The fully connected layer ideally provides a buffer between the learned features and the output with the intent of interpreting the learned features before making a prediction. For this model, we will use a standard configuration of 64 parallel feature maps and a kernel size of 3. The feature maps are the number of times the input is processed or interpreted, whereas the kernel size is the number of input time steps considered as the input sequence is read or processed onto the feature maps. The efficient Adam version of stochastic gradient descent will be used to optimize the network, and the categorical cross entropy loss function will be used given that we are learning a multi-class classification problem. The model is fit for a fixed number of epochs, in this case, 10, and a batch size of 32 samples will be used, where 32 windows of data will be exposed to the model before the weights of the model are updated.

As displayed in Table 5, the proposed classifier based on the weight of evidence measure using the discovered patterns and rules outperforms the other classifiers. In general, the other classifiers trained using the transformed feature matrix are able to obtain good accuracy (> 85%) so it is concluded that the transformed feature matrix representation can provide high-quality data summary to characterize the original spatial trajectory data for building a classifier.

Table 5

Experimental Results on Prediction of Physical Exercise Type on Synthetic Data

Classifier	**Classification accuracy**
C4.5 Decision Tree	95.2381%
CNN (raw trajectory)	85%
Logistic Regression	90.4762%
Random Forest	95.2381%
Support Vector Machine	94.2381%
The Proposed Classifier	**97.5000%**

To further examine the proposed clustering approach, we removed the ground truth that is to exclude the physical exercise type (the class attribute) from the synthetic data set to learn a clustering model by setting the number of clusters = 2. Then, we add back the class attribute to calculate the error percentage of the group assignment based on the majority value of the class attribute within each cluster. To compare the effectiveness of the learned representation, we trained clustering models using a number of traditional clustering algorithm including expectation maximization (Dempster, 1977), *k*-means (MacQueen, 1967), hierarchical clustering (Sibson, 1973) and cobweb (Fisher, 1987). Table *6* shows the clustering results. Obviously, the clustering result does correspond to the binary types of the physical exercise in the data set. The proposed clustering model is superior to the other clustering models. It is important to note that the proposed approach is a meta-algorithm that consists of an initial clustering done by hierarchical clustering and then a re-clustering step that treats the assigned cluster label as a class label to train a classifier and predict the labels as the final clustering results. Therefore, a good initial clustering result that locally optimizes the distance between groups can be further enhanced by the classifier based on the weight of evidence measure that globally optimizes the assignment of clustering label by taking into consideration the statistical significance of the patterns inherent in the training set.

Table 6

Clustering Results on Synthetic Spatial Trajectory Data

Clustering model	**Error percentage**
Expectation maximization	4.76%
k-means	4.76%
Hierarchical	4.76%
Cobweb	9.52%
The proposed 2-step	**4.28%**

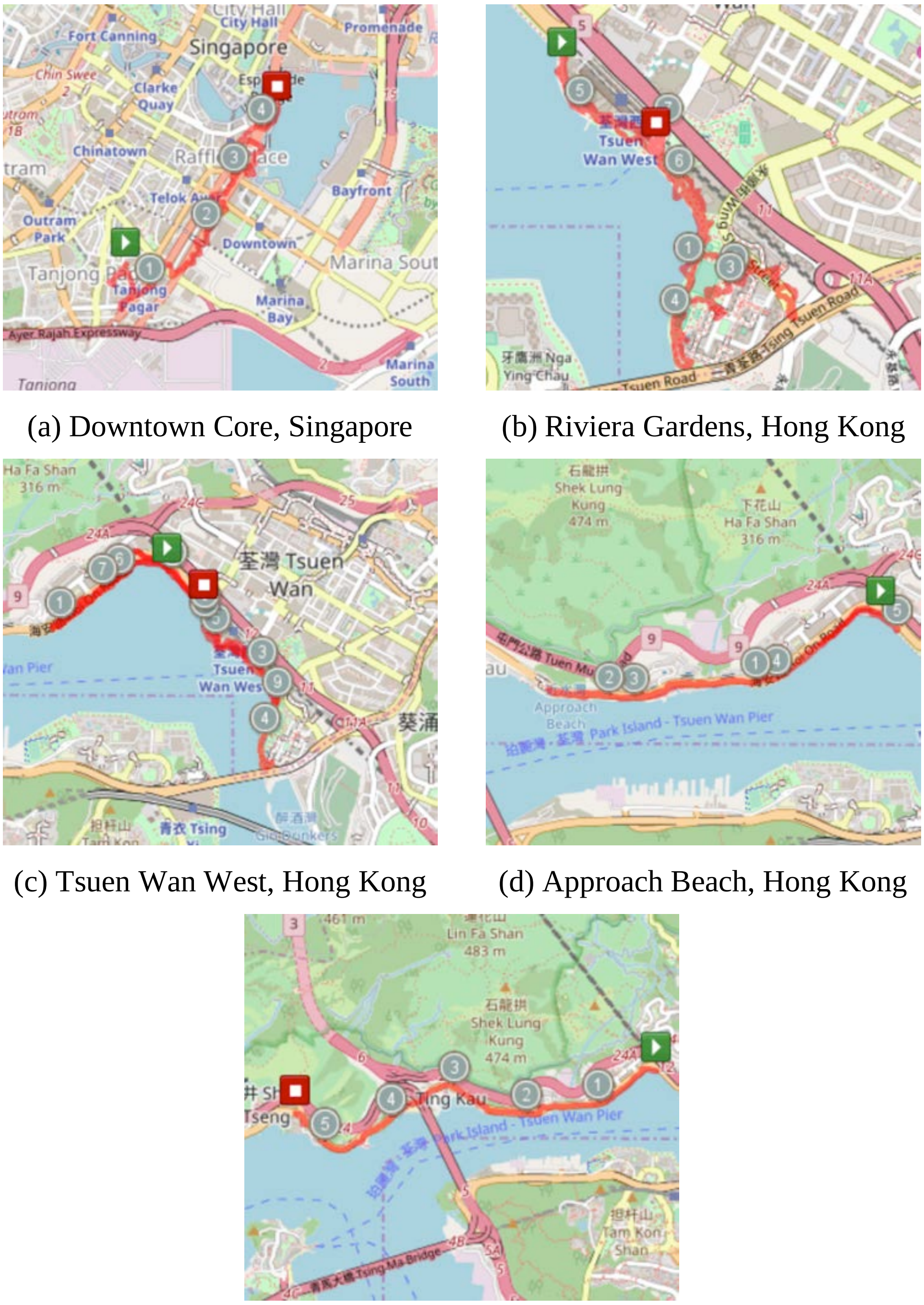
4.8.2 Real physical exercise data set.

This experiment aims at testing the effectiveness of the proposed algorithms when dealing with real-data with the presence of variation in every single spatial trajectory. The previous experiment using synthetic data cannot fully validate the classification power of the extracted features, as the spatial difference between trajectories of the same type of activities is very minimal.

In this experiment, we use our own physical exercise data to train a classifier to predict the type of exercises based on the discovered patterns. This GPS track data set consists of two types of physical exercises - hiking and running. Some routes of the same exercise type are in different areas. Even some routes in the same area are not completely identical as it is impossible to move along with the exact same route every time and there exists noise in the data during data collection due to signal interference, changing sampling rate and GPS accuracy of the smartphones.

The data is collected from September 2016 to June 2018 with 60 trajectories, i.e. 30 running routes and 30 hiking routes. We used a mobile application “MapMyRun“ to track the GPS during workout sessions and the route data can be downloaded after signing in www.MapMyRun.com. Route data are available in 3 formats: a GPS Exchange (GPX) of route information, a Keyhole Markup Language (KML) of the geographic path taken during the route, and a Training Center XML (TCX) with additional data with each track point such as heart rate. We opted to convert points from TCX files to extract latitude, longitude, and timestamp of each point along a route using Python (Python Software Foundation. Python Language Reference, version 3.6 available at www.python.org). For running, the routes contain different running routes in

New Territories, Hong Kong and Downtown Core, Singapore. For hiking, the recorded trajectories include hiking trails in New Territories and Kowloon, Hong Kong. Figure 9 and Figure 10 present five representative routes of both physical activity types. It is noteworthy to mention that for each type of physical activities, the route structure of each workout session can be very different even in the same district. Therefore, using the area and shape of the map to learn a classifier to predict the types is not an effective way but also poses some security and privacy concerns with the application of classification algorithms in the networking services storing personally identifiable information.



(a) Downtown Core, Singapore

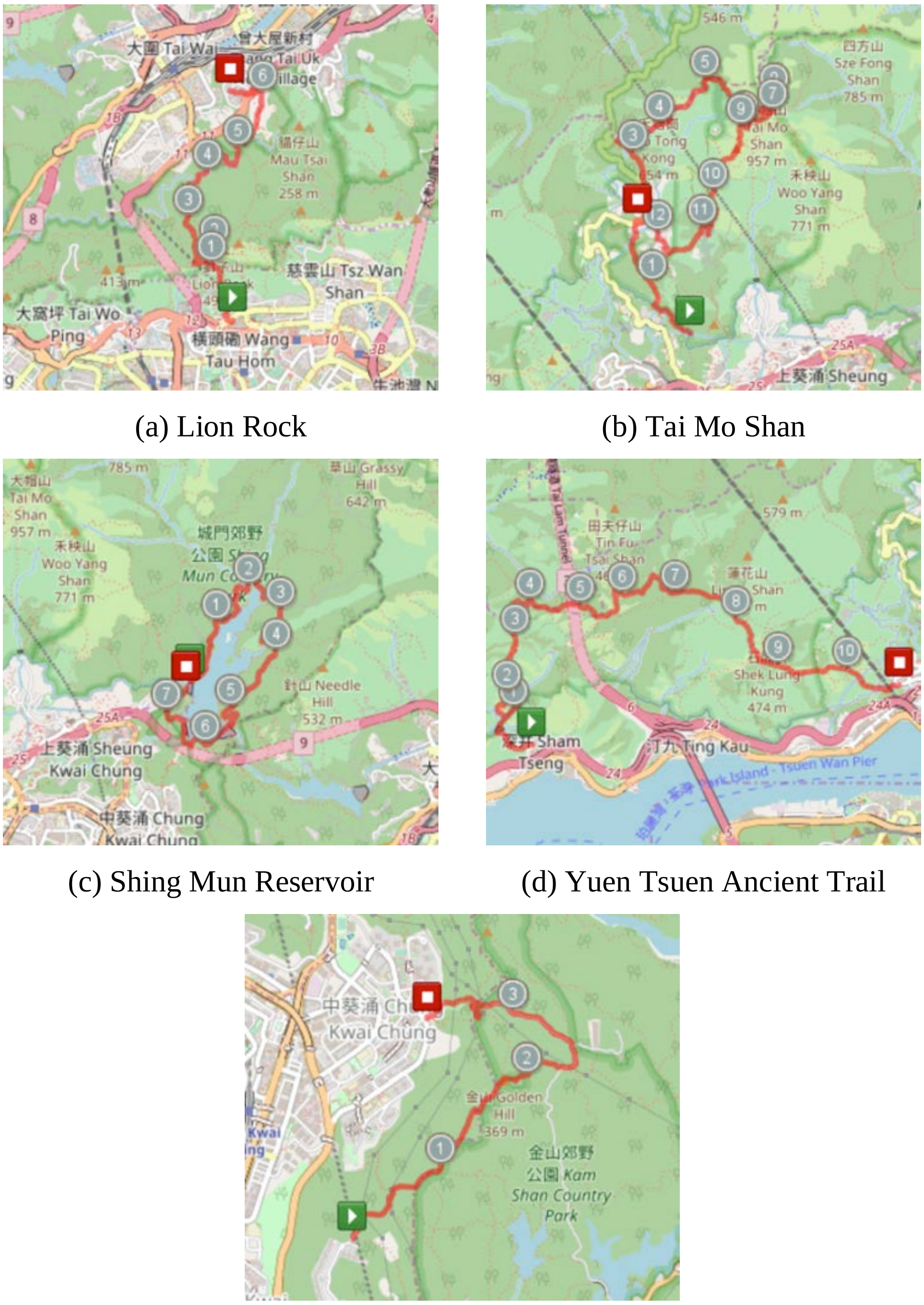
(b) Riviera Gardens, Hong Kong

(c) Tsuen Wan West, Hong Kong

(d) Approach Beach, Hong Kong

(e) Sham Tseng, Hong Kong

Figure 9. Representative Routes of Running Exercise.



(a) Lion Rock

(b) Tai Mo Shan

(c) Shing Mun Reservoir

(d) Yuen Tsuen Ancient Trail

(e) Kam Shan

Figure 10. Representative Routes of Hiking Exercise.

To assess the proposed algorithms' ability to deal with uncertainty, an appropriate amount of noise (10%) was added to the data set. We picked 3 trajectories from each type of exercises and re-label them with the wrong types. We used the proposed algorithms to pre-process the trajectories and generated a feature matrix with discretized attributes. The proposed algorithm mined 2,397 statistically significant patterns and 1,995 rules. Top 10 patterns and rules with their adjusted residuals and weights of evidence are shown in Table 7. To investigate if the discovered patterns and rules are able to differentiate the types of exercises, we build classifiers using the proposed algorithms, Support Vector Machine (SVM) (Platt, 1998) and Random Forest (RF) (Breiman, 2001) with default parameters. 10-fold cross validation is adopted to validate the result. The classification accuracy is listed in Table 8.

Table 7

Top 10 Discovered Patterns and Rules in Real Physical Exercise Data Set

IF {pattern} THEN {outcome}	**Adjusted residual**	**Weight of evidence**
IF {"30th percentile of speed" = "high" and "70th percentile of speed" = "high"} THEN {"type" = "running"}	11.051264	*infinity*
IF {"average acceleration derivative without stop" = "high" and "average acceleration derivative with stop" = "high"} THEN {"type" = "hiking"}	11.051264	*infinity*
IF {"100th percentile of cosine of turning angle" =	11.036207	*infinity*

“high” and “average speed without stop” = “high”} THEN {“type” = “running”}		
IF {“average acceleration derivative without stop” = “low” and “average acceleration derivative with stop” = “low”} THEN {“type” = “running”}	11.022704	*infinity*
IF {“40th percentile of acceleration derivative” = “low” and “70th percentile of acceleration derivative” = “low”} THEN {“type” = “running”}	11.010663	*infinity*
IF {“40th percentile of acceleration derivative” = “high” and “number of turning points” = “high”} THEN {“type” = “hiking”}	10.633763	*infinity*
IF {“50th percentile of speed” = “high” and “100th percentile of cosine of turning angle” = “high”} THEN {“type” = “running”}	10.591856	*infinity*
IF {“average speed without stop” = “high” and “average acceleration derivative with stop” = “low”} THEN {“type” = “running”}	10.547291	*infinity*
IF {“average acceleration derivative with stop” = “low” and “stop points per km” = “low”} THEN {“type” = “running”}	10.498896	*infinity*
IF {“10th percentile of speed” = “high” and “average acceleration derivative with stop” = “low”} THEN {“type” = “running”}	10.122051	*infinity*

Table 8

Experimental Results on Prediction of Physical Exercise Type on Real Physical Exercise Data

Classifier	**Classification accuracy**
Random Forest	86.7998%
SVM	83.7410%
The Proposed Classifier	**90.9100%**

It is interesting to note that all these classifiers can achieve high classification accuracy (> 80%) although noise is present in the real data set. The proposed classifier slightly outperforms the other two classifiers. It can be attributed to the positive influence of using the statistically significant patterns to be the rules to train the classifier.

4.8.3 GeoLife data set. In the previous 2 sets of experiments, the effectiveness to extract interesting patterns and to classify the trajectories with the presence of uncertainty and variation of records from the same class has been assessed and validated using a synthetic data set and a small-scale real data set. In addition to testing the practicality, in this experiment, we applied the proposed algorithms on GeoLife data set published by Microsoft Research. Researchers can follow the instruction from the project website to obtain a copy of the data set for research purposes (Zheng, Liu, Wang, & Xie, 2008). The project team in

Microsoft Research maintained the website and data set and published several versions of the data set. This is a large-scale data set with multiple years of user-contributed records. According to the latest version (version 1.3 released on 1 August 2012 covering April 2007 to August 2012) of the published data, the GPS trajectories in the data set were basically positioned every 1 to 3 seconds, and 69 users annotated labels of transportation modes. To prepare the data, user data with less than 10 annotations were excluded and eventually selected the data of 54 users for our experiments. The preparation method is in consistency with Endo, Toda, Nishida, and Kawanobe (2016) to make comparison possible. Each annotation contains a transportation mode, as well as beginning and ending times of the transportation. We labeled and extracted the section of GPS trajectories between the beginning and ending times with an annotation of transportation modes and used them as the selected data set for the experiments. Although there are eleven types of annotations, we used only seven types (walking, bus, car, bike, taxi, subway, and train), because the other four contains too few trajectories, i.e. less than 100 trajectories among all users of these four types. We also removed some trips that are too short, i.e. less than 10 GPS tracks. After the data selection, 8,764 trajectories are obtained for experimentation.

To evaluate the effectiveness of the proposed methods, we use the following methods to i) pre-process the data set and ii) train classifiers using the state-of-the-art algorithms to compare with the proposed algorithms.

i) Data pre-processing:

- Basic Feature (BF) extraction (Zheng, Liu, Wang & Xie, 2008): 10-dimensional features are extracted such as velocity, distance and time.

- BF + Advanced Feature (AF) extraction (Zheng, Li, Chen, Xie & Ma, 2008; Zheng, Chen, Li, Xie & Ma, 2010): 13-dimensional features are extracted including BF and advanced features such as stop rate and velocity change rate.
- Bag of Visual Words (BoVW) extraction: Image features are extracted from trajectory images using the method proposed by Vedaldi and Fulkerson (2010).
- SDNN (Endo, Toda, Nishida & Kawanobe, 2016): Deep features are extracted simply from vectors of a series of latitude, longitude and time stamp by Deep Neural Network (DNN).
- IDNN (Endo, Toda, Nishida & Kawanobe, 2016): Deep features are extracted by DNN from trajectory images.
- BF + AF + IDNN (Endo, Toda, Nishida & Kawanobe, 2016): Features are aggregated by BF, AF, and IDNN.
- FM + UMACA: The proposed feature matrix (FM) is discretized by the Unsupervised version of Mixed-mode Attribute Clustering Algorithm (UMACA) by our previous study (Wong, et al., 2010; Wu, Chan, & Wong, 2011).
- FM + SMACA: The proposed feature matrix (FM) is discretized by the Supervised version of Mixed-mode Attribute Clustering Algorithm (SMACA) by our previous study (Wong, et al., 2010; Wu, Chan, & Wong, 2011).
- CNN: A deep neural network architecture based on Convolutional Neural Network described and implemented in section 4.8.1 to learn the classifier from raw GPS trajectory (x_i, y_i, t_i) without feature

transformation. This is treated as a baseline method to assess the classification accuracy without data transformation.

ii) Classification: The above pre-processed data in i), except CNN, are fed into 3 classification models, including logistic regression (LR) (Le Cessie & Van Houwelingen, 1992), support vector machine (SVM) with Polynomial kernel (Platt, 1998), and C4.5 decision tree (DT) (Quinlan, 1993), with default parameters.

To demonstrate the flexibility of the proposed method, we applied the proposed algorithm to generate the feature matrix (**FM**) from the GeoLife data as a pre-processing step and then fed it into the above 3 described classifiers. Also, the FM, whose attribute values are continuous, can be discretized using supervised and unsupervised discretization methods, namely SMACA and UMACA, by our previous study (Wong, et al., 2010; Wu, Chan, & Wong, 2011). We conducted experiments on both 2 sets of discretized FM and 1 set of FM with continuous values for performance comparison. 5-fold cross validation over records is adopted to validate the classification models. The experimental results are given in Table 9. As shown in Table 9, the best accuracy can be obtained by logistic regression model trained by the data of the proposed FM with supervised discretization. It can be attributed to the improvement of the classification performance by the supervised discretization. If no supervised information is used, the proposed FM can still achieve high accuracy better than most of the other pre-processing methods, except IDNN that takes trajectory image features extracted by DNN. The use of trajectory images is sensitive to some areas of application as trajectory images can contain the precise series of GPS coordinates, leading some security concerns about unknowingly disclosing private information, such as the location of their home and office. It is interesting to show that simply treating raw GPS data as three-dimensional signal inputs (x_i, y_i, t_i) for deep learning algorithms performs poorly (41.935% accuracy only). A practical and promising way of transforming GPS trajectories into an easier consumable feature matrix for characterizing driving style and classification needs to be developed. This finding is consistent to the empirical studies in (Dong, Li, Yao, Li, Yuan & Wang, 2016) which is the first attempt of extending deep learning to driving behavior analysis based on GPS data.

To further evaluate the performance of our pattern mining and classification approach, we applied our pattern-mining algorithm on the 2 sets of discretized FM to discover a set of interesting patterns. As a result, the proposed algorithm discovered 4,815 interesting association patterns. Finally, we made use of the discovered patterns to train a classifier to predict the mode of transportation by calculating the weights of evidence and compared the classification accuracy to the other selected approaches with the highest accuracy according to Table 9. The results are reported in Table *10*. As displayed in Table *10*, the proposed classifier based on the weight of evidence measure using the discovered patterns on FM discretized in a supervised manner outperforms the other classifiers. In general, the proposed classifier trained by unsupervised discretization and other classifiers trained by the FM can obtain comparable accuracy. Therefore, it is interesting to conclude that the transformed feature matrix representation can provide high-quality data summary to characterize the original spatial trajectory data for building a classifier.

Table 9

GeoLife Data Set Classification Results using Different Pre-Processing Methods and Classification Models

Classifier / **Features**	**LR**	**SVM**	**DT**	**Classifier with the highest accuracy**
BoVW	57.9000%	**60.2000%**	54.8000%	SVM
SDNN	**38.6000%**	38.6000%	36.2000%	LR
BF	45.8000%	47.9000%	**63.2000%**	DT
BF + AD	48.3000%	52.4000%	**64.8000%**	DT
IDNN	**66.3000%**	64.9000%	62.6000%	LR
BF + AF + IDNN	**67.9000%**	66.0000%	65.9000%	LR
FM + SMACA	**68.6715%**	68.2908%	65.1409%	LR
FM without Discretization	60.3238%	46.1636%	**64.6271%**	DT
FM + UMACA	65.1602%	**66.7788%**	64.2685%	SVM
CNN	41.9350%			CNN

Table 10

GeoLife Data Set Classification Results of the Proposed Classifier and Other Selected Classifiers

(Pre-processing) + classification model	**Classification accuracy**
(BoVW) + SVM	60.2000%
(SDNN) + LR	38.6000%
(BF) + DT	63.2000%
(BF + AD) + DT	64.8000%
(IDNN) + LR	66.3000%
(BF + AF + IDNN) + LR	67.9000%
(FM + SMACA) + LR	68.6715%
(FM without Discretization) + DT	64.6271%
(FM + UMACA) + SVM	66.7788%
CNN	41.935%
(FM + SMACA) + Proposed classifier	**68.7200%**
(FM + UMACA) + Proposed classifier	66.7900%

4.8.4 Case study on driver telematics data set.

In this case study, we will investigate 1) the execution time of the whole pipeline against data samples of different input size, 2) analyze the output 2nd order patterns, and 3) compare the classification accuracy of the proposed system with other classification methods which take the data pre-processed by the proposed system.

An insurance company provides a large driver telematics data set. In the data set, 3,000 drivers' telematics data are collected. Each driver is recorded with 200 driving trips. Totally, there are 600,000 driving trips. This is considered the largest real-world data set among the experiments. The trips are recordings of the vehicle's GPS position in meters every second. Among these 200 trips for each driver, there may be a random number of trips not belonging to a driver due to transmission noise and error. Nevertheless, it is guaranteed that most of the trips in each driver are from the same driver. Without being told which false trips and the amount of them are, one goal is to identify them based on their telematics features to avoid using them to form interesting patterns. The ground truth of the data set is provided for the verification of the classification algorithm. Each driver and each driving trip is treated as a moving object and a trip of a moving object in the proposed algorithm respectively.

The result of execution time against different input size is shown in Figure 11 and Table *11*. The input size is sampled following a geometric series by its first term 1 and its common ratio 2 and also adding an interval per every hundred or thousand in between until 20,000, which is good enough to show if the algorithm is said to run in linear time. Figure 11 is a scattered plot of the data in Table 11 with a linear trend line. In effect, the execution time of the proposed system increases linearly per the input data size. It verified the analytical result in the complexity analysis. The proposed algorithms for discretization and pattern discovery are based on the pair-wise computation of attribute values and pattern occurrence counting so these operations can be parallelized to run in a distributed computing environment. We believe that the proposed system can deal with large input data size without increasing the execution time exponentially.

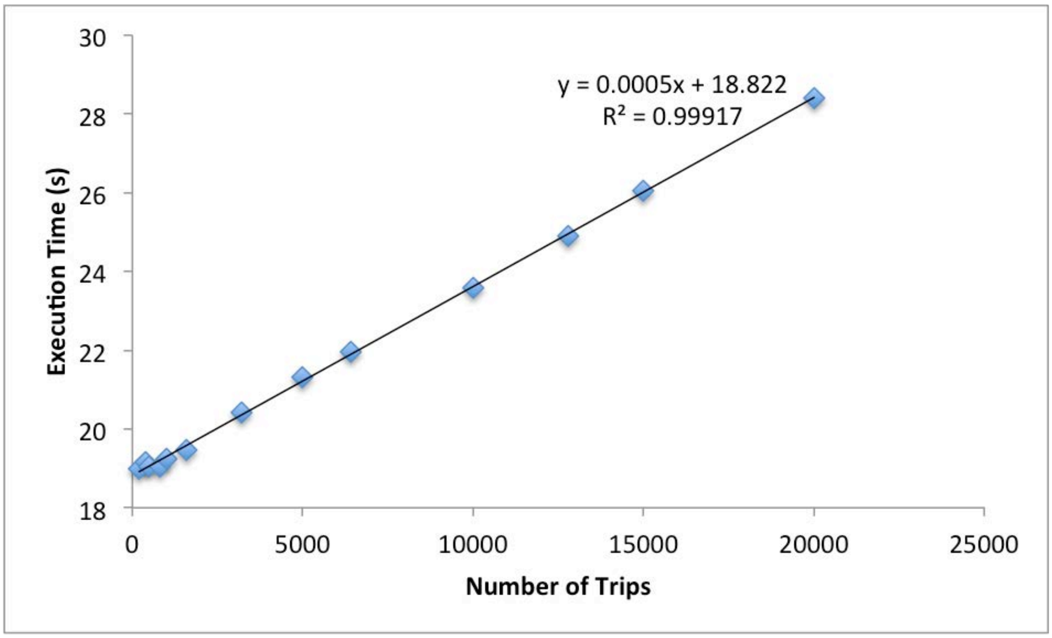


Figure 11. The Plot of Execution Time against Data Input Size.

Table 11

Execution Time against Data Input Size

Input size (# trip)	**200**	**400**	**500**	**800**	**1000**
Time (s)	19.003491	19.146416	19.037805	19.03937	19.246096
Input size (# trip)	**1600**	**3200**	**5000**	**6400**	**10000**
Time (s)	19.456244	20.42157	21.315115	21.961753	23.598002
Input size (# trip)	**12800**	**15000**	**20000**		
Time (s)	24.896966	26.064321	28.404084		

From the output of all 2nd order patterns, we extract and discuss the top 3 interesting patterns ranked by their adjusted residual values. One prominent pattern detected for a driver is:

$$< (\text{'MedianSpeed'}, 1), (\text{'StopPoints'}, 0) >$$

which means for a driver whose trips contain this pattern tends to drive very fast with few stop points. This may imply that this type of drivers usually takes the highway route, which is usually in high speed with no traffic lights to pause the drivers. Whereas for another type of drivers, the prominent pattern is:

$$< (\text{'PostStopAcc'}, 1), (\text{'PreStopAcc'}, 1) >$$

which means his or her driving trips usually have high acceleration after a stopping point and high deceleration for stop purpose. This may imply he or she may be a reckless driver with a hard brake before the stop and rushing out after stopping points. Most drivers have a common pattern:

$$< (\text{'MedianSpeed'}, 1), (\text{'MedianSpeedyTurning'}, 1) >$$

which means these drivers driving at a high speed are reluctant to slow down before turning.

To assess the classification power of the feature extraction and the proposed sample discretization rules from the raw trajectory data described in Table 2 of section 4.3 and the proposed classifier, we generated two data sets and fed them to different classification algorithms to compare the accuracy. Dataset A contains only the 81 extracted attributes and these attributes are discretized by MACA. Dataset B is a combined data set of Dataset A and the discretized attributes by using the proposed sample discretization rules. These data sets are then fed into 4 classifiers including the proposed classifier using the weight of evidence measure, Random Forest (RF) (Breiman, 2001), support vector machines (SVM) with Polynomial kernel (Platt, 1998) with default parameters and an ensemble model based on RF and SVM with each coefficient = 0.5.

The classification task is for each driver, we train a classifier using his or her own driving trips and then predict whether or not a trip, which can be from him/her or from others, belongs to this driver. For each data set of a driver, we add the same number of trips randomly selected from the other drivers as noise. 10-fold cross validation is adopted to assess the quality of the model. We completed this classification task for every driver in our data set, of which these classification accuracy results are averaged to produce an overall accuracy for comparison purpose. The results are reported in Table 12. Random Forest achieved the best accuracy with Dataset A where SVM is the worst. Generally, the accuracy can be further enhanced using the proposed discretized rules to generate more attributes as shown in results on Dataset B. One interesting note is that RF performed better even without the additional discretized attributes. It might be due to its' ability to have an internal unbiased estimate of the generalization error as the forest building progresses. It is interesting to see that the performance of different classifiers is not varied significantly with only less than 8.3% difference in accuracy. Overall, the classificatory power of the generated features that characterize the trajectories can be revealed based on the positive experimental results.

Table 12

The Accuracy of Different Classification Algorithms and Data Sets with Different Pre-Processing

Data set	**Classification algorithm**			
	Proposed algorithm	**RF**	**SVM**	**Ensemble**
A) Only extracted attributes	80.08%	**81.46%**	73.18%	78.98%
B) Combined A) and attributes discretized by proposed discretization rules	**81.86%**	81.37%	77.32%	80.16%

For further experimentations, we cluster the driver telematics data after removing the driver label. The clustering goal is to cluster the trips so that trips in the same cluster are more likely from the same drivers. To evaluate re-clustering, since the cluster label is known, we can calculate i) F1-measure, defined as $F(C_x, C_y) = \frac{2R(C_x,C_y)P(C_x,C_y)}{R(C_x,C_y)+P(C_x,C_y)}$, where $R(C_x, C_y) = \frac{cnt_{C_x,C_y}}{cnt_{C_x}}$, $P(C_x, C_y) = \frac{cnt_{C_x,C_y}}{cnt_{C_y}}$, $cnt_{C_x,C_y}$is the count of records with cluster label C_x in the assigned cluster C_y, cnt_{C_x} is the count of records with cluster label C_x, and cnt_{C_y} is the count of records in the assigned cluster C_y, and ii) clustering accuracy, defined as $CA = \frac{\sum_{i=1}^{k} cnt_{c^i}}{Total\ number\ of\ records}$, where cnt_{c^i} is the count of records in the i^{th} cluster and k is the number of clusters. To assess the clustering power, we perform an initial clustering by hierarchical clustering algorithm on (Feature Matric) FM to generate cluster labels and insert them into Transformed Feature Matrix (TFM). We argue that a good representation of driving behaviors from the transformed trajectory inputs facilitates the learning of a good cluster model. This TFM is fed into 4 classification algorithms (the proposed classifier, Random Forest (RF), Support Vector Machines (SVM) and an ensemble model based on RF and SVM. We selected 2 subsets of the data to conduct 2 experiments (small-scale and large-scale). The small-scale data set randomly picks 50 drivers so it contains 10,000 trips while the large-scale one randomly picks 1,000 drivers so it contains 200,000 trips. The clustering task is for each data set, we cluster it into *k* clusters where *k* = number of drivers. In re-clustering step, we adopt *10*-fold cross validation to assess the quality. The results are shown in Table 13 and Table 14. Generally, the accuracy can be further enhanced using the proposed approach. It is due to the weight of evidence measure to precisely measure the information uncertainty on the interesting patterns voting the class assignment. SVM and RF do not consider the interestingness of patterns so noise in the data set that cannot be filtered affects the performance. RF performed slightly better even with the presence of noise. It might be because of its ability to have an internal unbiased estimate of the generalization error as the forest building progresses. The performance of different classifiers in the re-clustering step is competitive with less than 10% difference of accuracy. It shows that a good representation of the transformed data can yield high quality clustering results by the-state-of-the-art methods.

Table 13

The Accuracy of Different Classification Algorithms for Re-Clustering on 10,000 Driving Trips

Method	**Accuracy (%)**	**F1-measure**
The proposed	**77.5**	**0.72**
SVM	70.3	0.64
RF	76.7	0.69
Ensemble	73.5	0.67

Table 14

The Accuracy of Different Classification Algorithms for Re-Clustering on 200,000 Driving Trips

Method	**Accuracy (%)**	**F1-measure**
The proposed	**62.8**	**0.59**
SVM	57.5	0.52
RF	60.3	0.56
Ensemble	58.9	0.54

In terms of the system design, the proposed system can well summarize low level telematics features and derive meaningful high-level features without losing too much information for further knowledge discovery processes such as clustering and prediction using the state-of-the-art classification algorithms while preserving the privacy of individual drivers' location information as the algorithms does not require using the trajectory images. These experiments have also demonstrated that the proposed system can efficiently, effectively and flexibly encode telematics data and transform them into attributed hypergraph representation (Figure 5 which depicts the top 3 discovered pattern) for pattern visualization which is human readable.

4.9 Summary of the Chapter

An extensive set of experiments conducted illustrates the overall algorithmic design and demonstrates the feasibility and practicality by the experimental results. Some system components and parameters can be configured to adjust the performance according to the application domains, such as the selection of a threshold for discretization rules and the significant level of the statistical significance test. The suggested direction for further research in this area will be discussed in the future work.

The proposed pattern discovery system for spatial trajectory data follows the basic structure of the contemporary data mining and pattern discovery system but possesses some very important and positive differences. The contemporary methods normally make use of a single attribute type of spatiotemporal data for pattern generation. To ensure a good representation of the object as well as the movements, we support to use multiple attribute types through the application of discretization in both supervised and unsupervised manners. For instance, route-related attributes are more correlated to speed-related attributes. This attribute combination is believed to provide better mining capabilities as multiple attribute types can describe an object more effectively and comprehensively. Since the mixed mode nature of attributes requires a reliable discretization method that can capture the correlated information inherent in the attribute set, we experimented using different discretized data sets to train classifiers and realized the positive effect contributed by that.

We have proposed a pattern discovery and representation method for spatial trajectory data, adopting an AHR that contains various possible objects, each of which provides both generalization within the object (i.e. moving attributes in vertices) and good inter-object relationship (i.e. interesting patterns in hyperedges). Further, the method shall require only a small training set for pattern and rule generation to characterize the data and train a classifier. It can also deal with large growing unsupervised data that may be inserted into the proposed system by updating the weights of evidence of the discovered rules at any time. Thus, the proposed system is developed to be able to learn new moving behaviors to optimize the existing classifier on the fly. The statistical back and the algorithmic design of the underlying techniques of the proposed system has been the foundation of the current research and development. The discussion of the experimental results demonstrated the feasibility and practicality of the proposed approach. The current system can be expanded and modified to tackle more pattern mining problems. In the next chapter, we will describe the pattern discovery approach to mine multivariate spatial time series data.

This page is intentionally left blank.

5 Pattern Discovery for Multivariate Spatial Time Series

5.1 Background

In this chapter, we propose to develop a pattern discovery approach consisting of some algorithmic techniques for discovering patterns from multivariate spatial time series (**MSTS**) and learning inference model. The approach targets to mine spatio-temporal patterns from MSTS and is able to tackle classification and clustering tasks based on the discovered patterns. We implement the proposed algorithms for performance evaluation and validation. To compare the performance, we benchmark the proposed algorithms with different traditional pre-processing techniques, a deep learning model (CNN) and clustering algorithms, including TARM, PCA, ARMA, k-means, HMM, on a simulated data set and a real data set. This proposed approach has also been applied to 3 case studies to demonstrate the applicability and practicability. Our major contribution is three-fold. 1) Classical spatial analysis studies entities using their topological, geometric or geographic properties. We define the MSTS data structure, its associated temporal attributes, and its clustering problem. 2) Traditionally, analytical techniques favor the spatial definition of objects as points. We propose a more general model to characterize MSTS. 3) Conventional approaches capture spatial dependency to provide information on spatial relationships in variable level. Our approach goes deeply to the attribute-value level.

Let's begin with a simple illustrative example of a multivariate spatial time series (MSTS) with 3 regions (*i*, *j*, *k*) containing patterns {ABC, GC, HC}, {AEB, BK} and {BBC, CDE} as shown in Figure 12. The symbols in the patterns are from the alphabet set ε due to the transformation of the time series (TS) in sequences (S). Here from Figure 12, this MSTS can be referred to as 3 multivariate time series (MTS) across 3 spatial locations. In region *i*, every pattern repeats 2 times and thus has occurrences of 2. A pattern is made up of a collection of blocks in which each block contains a local part of the pattern occurring in a univariate time series. Pattern {ABC, GC, HC} has 3 blocks, pattern {AEB, BK} has 2 blocks and pattern {BBC, CDE} has 2 blocks. The first and second patterns known as intra-patterns are along individual time series while the third pattern known as inter-pattern is across time series. There are some varying time delays between the blocks in the patterns. These patterns occurring sequential in time are referred to as temporal association patterns (TP). Thus, the number of blocks in a TP is also called the levels of TP (LT). Consider some temporal association patterns are also detected in region *j* and *k*. Among region *i*, *j* and *k*, pattern {ABC, GC, HC} span across them. This pattern is referred to as spatial association pattern (SP).

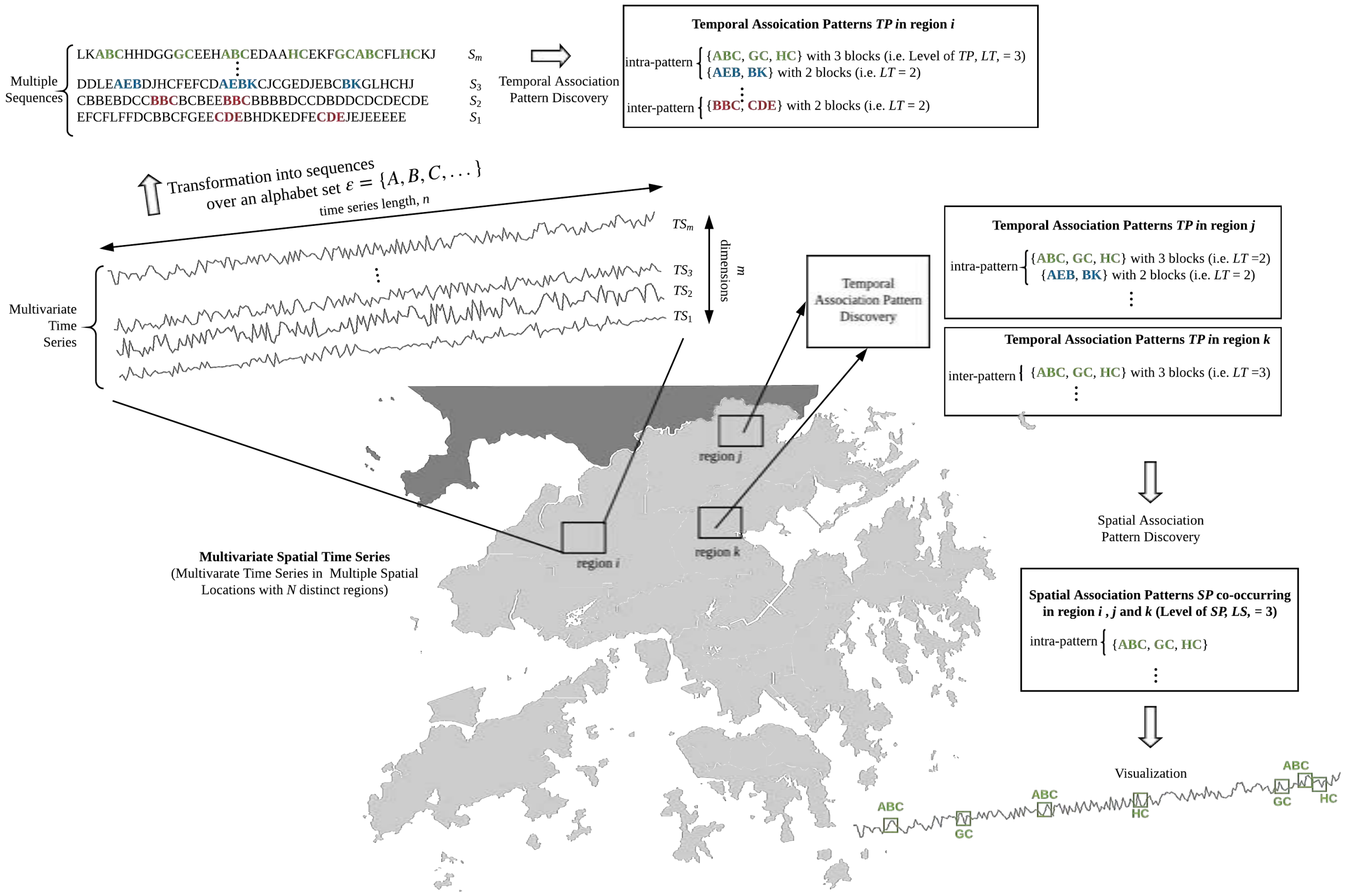

Multiple Sequences
LKABCHHDGGGCEEHABCEDAAHCEKFGCABCFLHCKJ S_m
DDLEAEBDJHCFEFCDAEBKCJCGEDJEBCBKGLHCHJ S_3
CBBEBDCCBBCBCBEEBBCBBBBDCCDBDDCDCDECDE S_2
EFCFLFFDCBBCFGEECDEBHDKEDFECDEJEJEEEEE S_1
Temporal Association Pattern Discovery
Temporal Assoication Patterns *TP in* region *i*
intra-pattern {ABC, GC, HC} with 3 blocks (i.e. Level of *TP*, *LT*, = 3)
{AEB, BK} with 2 blocks (i.e. *LT* = 2)
inter-pattern {BBC, CDE} with 2 blocks (i.e. *LT* = 2)
Transformation into sequences over an alphabet set $\varepsilon = \{A, B, C, ...\}$
time series length, *n*
TS_m
TS_3
TS_2
TS_1
m dimensions
Multivariate Time Series
Temporal Association Pattern Discovery
Temporal Association Patterns *TP in* region *j*
intra-pattern {ABC, GC, HC} with 3 blocks (i.e. *LT* =2)
{AEB, BK} with 2 blocks (i.e. *LT* = 2)
Temporal Association Patterns *TP in* region *k*
inter-pattern { {ABC, GC, HC} with 3 blocks (i.e. *LT* =3)
region *j*
region *i*
region *k*
Multivariate Spatial Time Series
(Multivarate Time Series in Multiple Spatial Locations with *N* distinct regions)
Spatial Association Pattern Discovery
Spatial Association Patterns *SP* co-occurring in region *i* , *j* and *k* (Level of *SP*, *LS*, = 3)
intra-pattern { {ABC, GC, HC}
Visualization
ABC
GC
ABC
HC
ABC
GC
HC

Figure 12. Schematic of Multivariate Spatial Time Series.

In this section, we introduced the background of pattern discovery for multivariate spatial time series. The rest of this chapter will systematically explain and illustrate the process for the proposed system. Section 5.2 describes the technical preliminaries that will restate the notations and definition of MSTS that are used in the proposed algorithms. Section 5.3 to section 5.6 formalizes the proposed pattern mining steps for MSTS with relevant mathematical notations to formally define the solution methods. Section 5.7 discusses the computational complexity of the proposed algorithm. Section 5.8 reports the experimental results obtained from both the synthetic data set and real-world data sets. This chapter ends with a summary in section 5.9.

5.2 Technical Preliminaries

In this section, we present technical preliminaries for mining MSTS. Given a set of MSTS, the proposed approach incorporates an effective initial multiple time series pattern-mining algorithm (Zhuang, Li, & Wong, 2014) to detect temporal patterns in a set of MTS for each location. Then, we propose a new algorithm to detect co-occurrence of the discovered temporal patterns across locations by mining a transformed spatio-temporal pattern matrix (**STPM**) that characterizes the space to form spatio-temporal patterns. Furthermore, we effectively integrate this spatio-temporal pattern-mining algorithm for classification and clustering. If the set of MSTS is labeled, the discovered patterns can be weighted to support or against a certain class membership for the construction of a classifier. If the set of MSTS is unlabeled, the discovered patterns in one location are compared against those discovered in the others so that MSTS that have similar discovered patterns are grouped together into the same cluster. In this work, we focus on spatio-temporal pattern discovery and the clustering to validate the effectiveness of the STPM representation, assuming MSTS is unlabeled. If the MSTS is labeled, the proposed approach can plug it into the clustering algorithm, as the clustering algorithm is made up of a classification algorithm in the second phase.

Definition 5.1 Multiple spatial locations. Suppose there are multiple spatial locations $L = \{l_1, \dots, l_{|L|}\}$, each of which is represented by a region label and its set of geographic coordinates g so that $L = \{(r_1, g_1), \dots, (r_{|L|}, g_{|L|})\}$ where each set of geographic coordinates g contains longitude x and latitude y coordinates. $|L|$ is the number of locations in the study area. There are totally N distinct regions $R = \{r_1, \dots, r_i, \dots, r_N\}$ which partition the locations of the study area. $G(x, y)$ is a function to retrieve the region r_i for a given longitude x and latitude y. To represent neighborhood between regions, let W be an adjacency matrix that assigns equal weights to all neighbors of regions, that is, $\{W\}_{i,j}$ = 1 if region r_i and r_j share a common border or 0 otherwise. For each location, there exists at least 1 *MSTS*.

Definition 5.2 Multivariate spatial time series (MSTS). A $MSTS = \{(MTS_1, x_1, y_2), \dots, (MTS_N, x_N, y_N)\}$ consists of N number of multivariate time series (MTS) each of which is associated with a longitude x and latitude y. With $G(x, y)$, a MSTS can be converted to a MTS with a region label $r_i, i \in \{1 \dots N\}$ so that $MSTS = \{(MTS_1, r_1), \dots, (MTS_N, r_N)\}$.

Definition 5.3 Multivariate time series (MTS). An MTS consists of m individual time series $TS = \{1, \dots, m\}$. A time series TS is a finite sequence of real values $(v_1, v_2, \dots, v_n)$ containing n observations with unique time points $TP = \{1, \dots, n\}$.

Definition 5.4 Sequence. A symbol sequence S is a sequence of characters $s_1, s_2, \dots, s_n$ over an alphabet set ε, where each $s_i \in \varepsilon$. ε is a set of distinct characters with size $|\varepsilon|$. n is the length of S. $S[i,j]$ is its substring from index i to j. Each character represents an event so S can be called an event sequence. After discretization, a TS can be transformed into a symbol sequence S. SAX (Lin, Keogh, Wei & Lonardi, 2007), a well-known discretization method for time series data mining practitioners, is adopted here for discretization. Therefore, a MTS can be transformed into a set of multiple symbol sequences $S_1, S_2, \dots, S_m$.

Definition 5.5 Pattern. A pattern P is a short sequence of consecutive characters $p_1, p_2, \dots, p_{|P|}$ over ε where $|P|$ is the length of the pattern. A pattern's length should be at least 2. Otherwise, each symbol in the alphabet set ε is a pattern.

Definition 5.6 Pattern occurrence. A pattern P is always associated with a symbol sequence S. P occurs in an interval $[i,j]$ in S if and only if $P = S[i,j]$. o_P denotes the occurrence of P. All occurrences of P are recorded in its occurrence list L_P so $|L_P|$ is the number of occurrences of P in S.

Definition 5.7 Frequent pattern. A frequent pattern is a pattern with its number of occurrences $|L_P| > min_o$ where min_o specifies the minimum number of occurrences required.

Definition 5.8 . Temporal association of patterns. A temporal association pattern *TP* is an association of patterns occurring sequentially in time. Each pattern P^i is a block of a TP. It implies P^{i+1} occurs within a certain specified time delay t_d after P^i occurs for $i = 1, \dots, LT - 1$. There are totally LT blocks for a TP and we call LT level of TP. max_{LT} specifies the maximum level of TP.

When all frequent patterns of a TP are from the same sequence, TP is called an auto association pattern or intra pattern. Otherwise, it is called cross association pattern or inter pattern. $|TP|$ is total number of temporal association patterns.

Definition 5.9 Temporal pattern occurrence. The set of occurrences of TP in the *MTS* is denoted by L_{TP}. $|L_{TP}|$ is the total number of occurrences.

Definition 5.10 Spatial association of patterns. A spatial association pattern SP is an association of multiple temporal association patterns co-occurring in multiple regions. Each TP^i is a building block of SP. It implies that TP^{i+1} occurs in a region r_j other than the region r_i of TP^i where $r_i \neq r_j$. There should be at least $2\, TPs$, i.e. 2 blocks, in a SP. Otherwise, every TP is a SP. There are totally LS blocks for SP and we call LS the level of SP. max_{LS} specifies the maximum level of SP. $|SP|$ is the total number of spatial association patterns.

Definition 5.11 Spatial association occurrence. The set of occurrences of SP in all *MTS* in different regions is denoted by L_{SP}. $|L_{SP}|$ is the total number of occurrences.

Definition 5.12 Statistical significance. The statistical significance based on adjusted residual (Wong & Wang, 1997) d_P measures how significantly the observed frequency of occurrences of an association pattern $|L_P|$, which can be a TP and/or a SP, deviates from its expected frequency E_P adjusted by its variance V_P. It is given below.

$$d_P = \frac{(|L_P| - E_P)/\sqrt{E_P}}{\sqrt{V_P}}, P \in \{TP, SP\} \qquad (5.1)$$

Definition 5.13 Weight of evidence. The weight of evidence measure provided by a pattern p^i for or against the classification of an object X into class c_i is defined as:

$$W^i\left(X\middle|p^i\right) = W(X \in c_i / X \notin c_i \mid X \text{ is characterized by } p^i)$$

$$= I\left(X \in c_i : X \text{ is characterized by } p^i\right)$$

$$- I\left(X \notin c_i : X \text{ is characterized by } p^i\right)$$

$$= \log \frac{P(X \in c_i \mid X \text{ is characterized by } p^i)}{P(T \in c_i)} \qquad (5.2)$$

$$- \log \frac{P(X \notin c_i \mid X \text{ is characterized by } p^i)}{P(X \notin c_i)}$$

$$= \log \frac{P\left(X \text{ is characterized by } p^i \middle| X \in c_i\right)}{P(X \text{ is characterized by } p^i \mid X \notin c_i)}$$

where $I(\,)$ is the mutual information. It is positive if p^i provides positive evidence supporting X is classified to $\mathrm{c_i}$, otherwise, it is negative, or zero.

$P\left(X \text{ is characterized by } p^i \middle| X \in c_i\right)$ is the probability that an object X contains a pattern p^i given that X belongs to c_i. It is computed by counting the occurrence of objects in the database containing pattern $p^i$and belonging to class c_i divided by the number of objects belonging to class c_i.

$P\left(X \text{ is characterized by } p^i \middle| X \notin c_i\right)$ is the probability that an object X contains a pattern p^i given that X does not belong to c_i. It is computed by counting the occurrence of objects in the database containing pattern p^i and not belonging to class c_i divided by the number of objects not belonging to class c_i.

$W^i(X|p^i)$ can be interpreted as a measure of the difference in the gain in information when an object X containing p^i is classified into $\mathrm{c_i}$ as opposed to other classes. $W^i(X|p^i)$ is positive if p^i provides positive evidence supporting the classification of X into $\mathrm{c_i}$, otherwise, it is negative.

Given the interesting patterns $P = \{p_1^1, \dots, p_j^i, \dots, p_{m_i}^{|C|}\}$, discovered for each corresponding $|C|$ classes, $\mathrm{c_1}, \dots, \mathrm{c_i}, \dots, \mathrm{c_{|C|}}$, an unseen object X_u can be classified by matching it against the patterns in each of classes. An unseen MSTS object X_u is first transformed to a list of attributes using the proposed STPM to construct a set of patterns P^u for matching. Then for every pattern p_j^i that X_u matches, there is some evidence $W^i(X_u|p_j^i)$ provided by it for or against the classification of X_u into $\mathrm{c_i}$. Assuming that X_u matches with $n_i \leq m_i$ patterns in P of $\mathrm{c_i}$, we calculate a total weight of evidence measure for X_u to be classified into $\mathrm{c_i}$.

Definition 5.14 Total weight of evidence. The total weight of evidence provided by each of individual patterns is a measure for X_u to be classified into $\mathrm{c_i}$ and is defined as:

$$\begin{aligned} &W^i(X_u) \\ &= W(\, X_u \in c_i / \, X_u \notin c_i \mid X_u \text{ is characterized by } p_1^i, \dots, p_j^i, \dots, p_{m_i}^i) \\ &= \sum_{j=1}^{m_i} W(\; X_u \in c_i / \, X_u \notin c_i \;\mid\; X_u \text{ is characterized by } p_j^i) \end{aligned} \tag{5.3}$$

5.3 Temporal Association Pattern Discovery

After SAX transformation of MTS into multiple symbol sequences, we discover frequent patterns (definition 5.7) from each sequence. To reduce the number of frequent patterns, we prune them by adopting the algorithms proposed by Zhuang, Li, and Wong (2014) to effectively extract closed and non-redundant patterns.

Once all non-redundant patterns are extracted, a hierarchical clustering algorithm, which repeatedly groups the most similar pair of clusters into a new cluster until forming a single cluster with all objects, with an appropriate priority queue to improve the runtime is used to group similar patterns to further reduce the number of patterns. After this, by updating the occurrence list L_{TP}, a new occurrence list L_C is created for each cluster of patterns by taking the union of all occurrences of patterns in the cluster.

After all pattern clusters, treated as the building blocks for each sequence, are discovered, we can detect temporal associations (definition 5.8) among them based on statistical significance (definition 5.12). It forms pattern clusters for each sequence first to yield intra-patterns and grows temporal associations level by level to yield inter patterns. To compute the statistical significance d_{TP} for temporal association patterns and given the above definitions to count the observed frequency of occurrences of association patterns, we want to estimate the expected frequency of occurrences of association patterns and its variance. Let TP' as the temporal association composed of $LT - 1$ blocks of TP and P^{LT} as the last block of TP. Assuming that the last block occurs randomly at a position in its respective sequence with probability estimated as $p = \frac{|L_{P^{LT}}|}{|S|}$, for temporal association of patterns, we take the association of TP' with P^{LT} as the estimate of the expected frequency of occurrences of TP. Then, the expected frequency of occurrences E_{TP} for TP is calculated as $Pr\ (\gamma' \vee \rho^{LT})\ \cdot |L_{TP'}|$ and variance $V_{TP} = E_{TP} \cdot\ (1 - Pr\ (\gamma' \vee \rho^{LT}))$, where γ' is an occurrence of TP', ρ^{LT} is an occurrence of P^{LT} and $Pr\ (\gamma' \vee \rho^{LT})$ is the probability that γ' is associated with ρ^{LT}. $Pr\ (\gamma' \vee \rho^{LT})$ is calculated as $1 - (1 - p)^{t_d + 1}$. After all temporal association patterns in a region r_i are discovered, the above procedure repeats until all regions R are processed.

5.4 Spatio-Temporal Pattern Matrix Representation (STPM)

For all regions $R = \{r_1, \dots, r_{\mathrm{N}}\}$, putting all temporal association patterns together, we can form the final group of M temporal association patterns, where $M = R \times M'$ and M' is the number of discovered TP in each region r_i. Here, d_{TP} refers to as statistical significance d_P of *TP*. We can generate a *M*-dimensional feature space and transform each region r_i into a vector X_i in the *M*-dimensional feature space. Each vector X_i, where $i = 1, \dots, N$, is then characterized by M attributes, denoted as $A = \{A_1, \dots, A_j, \dots, A_M\}$, whose values $d_{i1}, \dots, d_{ij}, \dots, d_{iM}$, where d_{ij} (definition 3.1.12) represents the amount and the statistical significance of association patterns found in the j^{th} attribute in region i. We call this representation a spatio-temporal pattern matrix (STPM) that characterizes the region by the discovered *TP*. To better visualize STPM, a table formatted with *N* rows and *M* columns (Table 15) is used. The discovered *TP* of each region (record) are the columns (attributes) in the Table 15. Sorting the column by d_{TP} values can rank them based on their statistical significance.

Table 15

Spatio-Temporal Pattern Matrix (STPM)

$\boldsymbol{d_{TP}}$	$\boldsymbol{A_1}$	…	$\boldsymbol{A_j}$	…	$\boldsymbol{A_M}$
$\boldsymbol{X_1}$	d_{11}	…	d_{1j}	…	d_{1M}
…	…	…	…	…	…
$\boldsymbol{X_i}$	d_{i1}	…	d_{ij}	…	d_{iM}
…	…	…	…	…	…
$\boldsymbol{X_N}$	d_{N1}	…	d_{Nj}	…	d_{NM}

5.5 Spatial Association Pattern Discovery

After transforming the MSTS into STPM, we detect interesting associations between temporal associations across regions. To do so, we first discretize the values of d_{TP} based on the unsupervised algorithm described in Wong et al. (2010) and Wu, Chan and Wong (2011), which uses an information measure that reflects interdependence to group attributes and identify the representative attribute in each attribute group to drive the discretization. For now, each attribute A_j contains only interval event values denoted as $A_j = \{A_j^b | b = 1, \dots, B\}$ where B is the number of bins. Optimizing the discretization is not the issue addressed here, so we will not further discuss it.

To detect an association, we construct a contingency table to count occurrences of values between 2 attributes i.e. A_j^p is p^{th} value of j^{th} attribute and $A_{j'}^k$ is k^{th} value of j'^{th} attribute, $j \neq j'$. Let o_{pk} be the total number of occurrences when $A_j = A_j^p$ and $A_{j'} = A_{j'}^k$; $e_{pk} = \frac{1}{T}\sum_{i=1}^{B} o_{pi} \sum_{i=1}^{B'} o_{ik}$ where $\sum_{i=1}^{B} o_{pi}$ is the total number of counts when $A_j = A_j^p$, $\sum_{i=1}^{B'} o_{ik}$ is the total number of counts when $A_{j'} = A_{j'}^k$, and T is the number of records. With o_{pk} and e_{pk}, we can detect whether or not o_{pk} is significantly different from e_{pk} by adjusted residual (definition 5.12 and equation 5.1) substituting $|L_P|$, E_P and V_{pk} by o_{pk}, e_{pk} and V_{pk} respectively to obtain the following equation:

$$d_{pk} = \frac{(o_{pk} - e_{pk})/\sqrt{e_{pk}}}{\sqrt{V_{pk}}} . \tag{5.4}$$

To reveal statistical significance, at 95% confidence level, if $d_{pk} > 1.96$, we can conclude it is a positive association; if $d_{pk} < -1.96$, it is a negative association; if $-1.96 < d_{pk} < 1.96$, it is random. Table 16 visualizes the contingency table. According to definition 5.10, for $d_{pk} > 1.96$, A_j^p and $A_{j'}^k$ should come from different regions for their association to be considered as a SP. Combining the step from section 5.3 to section 5.5, Figure 13 illustrates the algorithm by the pseudo-code. This SP and its occurrences o_{pk} will be added to L_{SP} (definition 5.11).

Table 16

Contingency Table of A_j and $A_{j'}$

A_j $A_{j'}$	$\boldsymbol{A_j^1}$	**…**	$\boldsymbol{A_j^p}$	**…**	$\boldsymbol{A_j^B}$	**Total**
$\boldsymbol{A_{j'}^1}$	o_{11} (e_{11})	$\cdots$	o_{p1} (e_{p1})	$\cdots$	o_{B1} (e_{B1})	o_{+1}
…	$\cdots$	…	…	…	…	…
$\boldsymbol{A_{j'}^k}$	o_{1k} (e_{1k})	$\cdots$	o_{pk} (e_{pk})	$\cdots$	o_{Bk} (e_{Bk})	o_{+k}
…	$\cdots$	…	…	…	…	…
$\boldsymbol{A_{j'}^{B'}}$	$o_{1B'}$ $(e_{1B'})$	$\cdots$	$o_{pB'}$ $(e_{pB'})$	$\cdots$	$o_{BB'}$ $(e_{BB'})$	$o_{+B'}$
Total	o_{1+}	$\cdots$	o_{p+}	$\cdots$	o_{B+}	$Total$

```
Input:
  $MSTS = \{(MTS_1, x_1, y_2), \ldots, (MTS_N, x_N, y_N)\}$ (original database)
  $|\varepsilon|$ (size of alphabet set $\varepsilon$ for SAX)
  $min_o$ (min number of occurrences for frequent patterns)
  $t_d$ (max time delay for temporal association $TP$)
  $max_{LT}$ (max level of temporal association $TP$)
  $max_{LS}$ (max level of spatial association $SP$)
  $sig$ (significance threshold value)
Output:
  $P$ (set of discovered spatio-temporal patterns)
Algorithm:
  $P = \emptyset$; $T = \emptyset$ (let $T$ be an empty set to store all $TP$)
  For each $MTS_i \in MSTS$
    Convert all $TS \in MTS_i$ into a set of sequences $S$ by SAX with $|\varepsilon|$
    Construct suffix tree $ST$ for $S$
    Traverse $ST$ to discover a set of frequent pattern $FP$ with $min_o$
    Initialize $T$ by assigning $FP$ as 1st level of $TP$
    While max level of $T$ <= $max_{LT}$
      For each pair $TP^j$ and $TP^k \in T$ with time delay between them < $t_d$
        Let $TP'$ be the candidate pattern of associating $TP^j$ and $TP^k$
        Calculate $d_{TP'}$ after obtaining $L_{TP'}$ (for detecting association)
        Evaluate statistical significance by checking $d_{TP'}$ > $sig$
        If $d_{TP'}$ > $sig$
          Add $TP'$ to $T$
        End
      End
    End
  End
    Add $T$ to attribute set $A$
  Construct STPM using $MSTS_x, x = \{1 \ldots N\}$ and $A_y, y = \{1 \ldots M\}$
  Initialize $PC_n$ candidate set of spatio-temporal patterns of order $n$
  For iterator $n$ = 2: $max_{LS}$
    If $n = 2$
      Initialize $PC_n$ based on all possible combination of $A$
    Else
      Initialize $PC_n$ based on all possible combination of $PC_{n-1}$
    End
    For each pattern $P_n \in PC_n$
      Calculate $d_{P_n}$
      If $d_{P_n} > sig$
        Insert $P_n$ into $P$ ($P_n$ is statistically significant)
      Else
        Remove $P_n$ from $PC_n$ ($P_n$ is not statistically significant)
      End
    End
  End
  Return $P$
```

Figure 13. The Pseudo-Code of Spatio-Temporal Pattern Discovery Algorithm.

5.6 Clustering, Re-clustering, and Classifying Spatio-Temporal Pattern Matrix

With STPM, it is readily available to perform cluster analysis on the set of d_{TP} by treating each vector X_i, characterized by M attributes, as an object by the state of the art clustering algorithm. The proposed clustering approach consists of 2 stages, initial clustering and re-clustering. The initial clustering stage, which locally optimizes the clustering, assigns cluster labels to the objects. We believe a good clustering result should generate good cluster label so we treat the cluster label as the class label to perform classification so as to globally partition the objects. This re-clustering stage basically is to fit a clustered STPM into a classifier we previously proposed in chapter 4's section 4.5. Therefore, if the MSTS is labeled, we can perform classification analysis directly by skipping the initial clustering stage.

For initial clustering, we adopt a popular agglomerative hierarchical clustering to repeatedly group the most similar pair of clusters into a new cluster until forming a single cluster with all objects. It is optional to apply a suitable cutoff level to obtain a specific number of clusters based on prior knowledge. If there is no domain knowledge to specify the number of clusters or inspecting the dendrogram to see if it suggests a particular number of clusters is considered subjective, one can apply some heuristics to determine the optimal number of clusters. Section 4.6 discussed such heuristics. Pearson correlation coefficient is used for distance measure rather than Euclidean distance so as to be better in dealing with noise (Ma, Chan & Chiu, 2005). For a pair of objects X_i and X_j with values of M attributes, the similarity measure is defined as:

$$Sim(X_i, X_j) = \frac{M \sum_{k=1}^{M} d_{ik} d_{jk} - \sum_{k=1}^{M} d_{ik} \sum_{k=1}^{M} d_{jk}}{\sqrt{M \sum_{k=1}^{M} {d_{ik}}^2 - (\overline{d_\imath})^2} \sqrt{M \sum_{k=1}^{M} {d_{jk}}^2 - (\overline{d_\jmath})^2}}\,. \tag{5.5}$$

To reflect the contiguity between different spatial units, we introduce $W_{i,j}$ (definition 5.1) and add it as a term to $Sim(X_i, X_j)$ as the spatial penalty. This forms a new similarity measure $Sim'(X_i, X_j)$, defined as:

$$Sim'(X_i, X_j) = Sim(X_i, X_j) + W_{i,j}, \tag{5.6}$$

for clustering. Treating the assigned cluster label in initial clustering as the class label, let p^i be an interesting pattern discovered from class i, c_i. In a supervised manner, if the interesting pattern p^i is conditioned by the class label c_i, it can be treated as a classification rule (Wang and Wong, 2003), i.e. if {antecedent or left-hand-side or LHS} then {consequent or right-hand-side or RHS}. The weight of evidence measure W (definition 5.13) in information theory (Wang and Wong, 2003) is used to quantify the evidence of the joined significant rules to support or against a certain class membership. An example rule for classifying a MTS is if $\{\text{temporal pattern 1} = \text{high and temporal pattern 2} = \text{low}\}$ then $\{\text{class} = 1\}$ with a weight of evidence of a certain value.

The task of classification is to maximize the total weight of evidence $W^i(X_u)$ (definition 5.14). The total weight of evidence for X_u to be classified into each of $c_1, c_2, \ldots, c_{|C|}$ is computed and X_u is assigned to the class that can give the highest total weight of evidence. This measure is able to differentiate the case that when some identical objects refer to different classes in the training set as the class, assignment of X_u is by the highest total weight of evidence. Combining the above steps, the pseudo-code is given in Figure 14.

```
Input:
  $X_u$
Output:
  $label_u$ (an assigned cluster or class label to $X_u$)
Variables:
  $P = \{p_1^1, \dots, p_j^i, \dots, p_{m_i}^{|C|}\}$ (set of discovered patterns from STPM)
  $Label = \{label_1, label_2, \dots, label_{|k|}\}$ (set of cluster labels or class labels)
  $W^i(X|p^i), i = 1, \dots, m_i$ (set of weights of evidence)
  $P^u$ (set of patterns from $X_u$ for matching)
  $W^i(X_u)$ (set of total weights of evidence)
Algorithm:
  $P^u$ = transform $X_u$ into a set of patterns for matching
  For each discovered pattern $p_j^i \in P$
    For each pattern for matching $p_x \in P^u$
      If $p_j^i$ matches $p_x$
        For each $label_i \in Label$
          $W^i(X_u)$ = $W^i(X_u)$ + $W^i(X|p_j^i)$
        End
      End
    End
  End
  $label_u$ = $label_i$ with max($W^i(X_u)$)
  Return $label_u$
```

Figure 14. The Pseudo-Code of Re-Clustering and Classification Algorithm for STPM.

5.7 Complexity Analysis

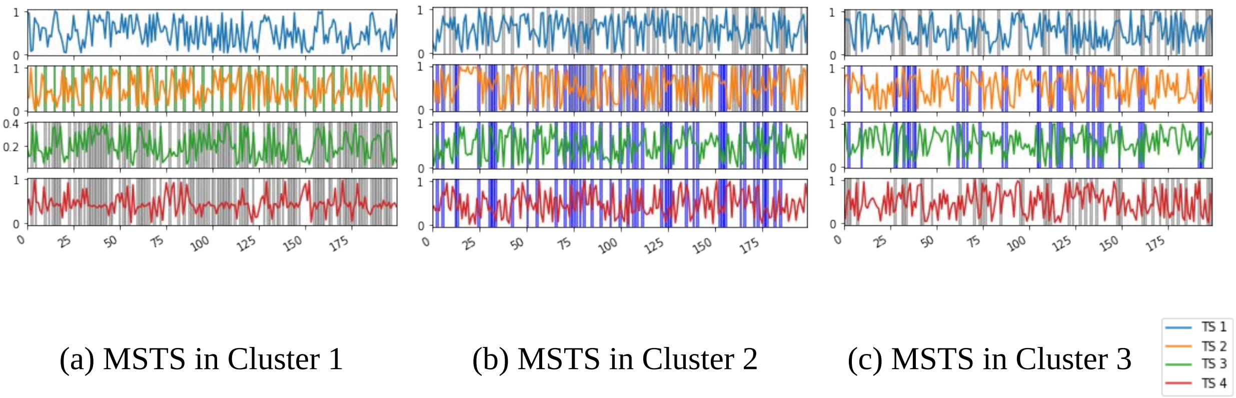
For the temporal association pattern discovery method of each region, the runtime and space complexity is $O(m^{LT} n^{LT+1} t_d^{LT-1})$ where m is the number of time series, n is the number of time points, LT is the level of the associations, and t_d is the time delay. Now, for the spatio-temporal association pattern discovery, we have the STPM for all regions $R = \{r_1, \dots, r_\mathrm{N}\}$ forming a $R \times M'$ matrix and let k be the number of distinct cluster labels. To compute the contingency table for all $\binom{M'}{2}$ pairs of discretized attributes, the computational complexity is $O(M'^2N)$. For $i > 2$, to generate i^{th} order spatio-temporal patterns, the computational complexity is $O(M'^iN)$. For higher order patterns, the candidate set is greatly reduced by previous iterations as the algorithm only considers growing statistically significant patterns, so all pattern candidate set should be much less than all possible combinations of i^{th} order patterns, i.e. $|PC_i| < M'^i$. If we predefine the number of discretized attributes, the overall computational complexity is linear in terms of N. The initial clustering scans through the entire feature matrix once, so the computational complexity of calculating the pair-wise similarity is $O(M'N^2)$. For the re-clustering, it calculates the weight of evidence of all individual interesting patterns supporting or refusing the classification of an object into a class. Given Q interesting patterns and k distinct cluster labels, it calculates kQ times. Each time scans through the entire database of N regions. Consequently, it takes QNk operations to obtain all weights of evidence between each interesting pattern and each class. Hence, its computational complexity is $O(kNQ)$. The classification of an unseen MSTS requires the generation of M' discretized attributes from its MTS and match them against Q interesting patterns to look for the highest total weight of evidence to support the classification into a certain class. As a result, it takes $kM'Q$ operations in total for which the computational complexity is $O(kM'Q)$ for the prediction of class.

5.8 Experimental Results

In this experiment section, we have performed both an experimental and case study using the proposed algorithms. Extensive experiments on synthetic data, real data and case studies have been conducted. Partial results described in this chapter have been published in Wu and Chan (2018c) and submitted to a journal (Wu & Chan, 2018b). In all of the below experiments and case studies, we set $|\varepsilon| = 10$, t_d to 6, $max_{LT} = 2$, and $max_{LS} = 5$ unless we re-specify these values. These values are chosen based on the domain knowledge and experiment setting. Other parameters will be specified in each experiment section. The proposed algorithms in the system are implemented in the programming language Python 3.6 with NumPy, a package for scientific computing, and Pandas, a software library for data manipulation. The experiments are carried out on a personal computer (MacBook Pro) with 2.9 GHz Intel Core i5 processor and 16 Gb RAM running macOS version 10.13.

5.8.1 Synthetic data set.

In this experiment, we generated MSTS for each of 60 regions, i.e. 60 MSTS, which belong to 3 clusters $\{C_1, \ldots, C_3\}$ each of which contains 20 MTS. For MSTS in the same cluster, they share common borders, i.e. $\{W\}_{i,j} = 1$ for region r_i and r_j sharing a common border and in the same cluster or 0 for region r_i and r_j in a different cluster. For each location, there exists at least 1 MSTS. Each MTS consists of 4 TS, i.e. $\{TS_1, \ldots, TS_4\}$. For each TS in each cluster, intra-patterns and inter-patterns are embedded so that MTS within the same cluster contains more similar patterns than those from others. Each TS contains 200 time points and can take on real values from 0 to 1. For C_1, TS_1's values are uniformly distributed within [0, 1]; TS_2's values are uniformly distributed within [0.18, 0.44] in every 5 time point interval stochastically or otherwise uniformly distributed within [0, 1]; TS_3 takes on values uniformly distributed within [0.03, 0.39]; TS_4 takes on values uniformly distributed within [0.32, 0.49] if TS_3's value falls into [0.18, 0.44]. For C_2, TS_2's values are uniformly distributed within [0, 1]; if TS_2's value falls into [0.18, 0.44], the corresponding value in TS_1 is uniformly distributed within [0.53, 0.86]; if TS_2's value falls into [0.40, 0.66], the corresponding value in TS_4 is uniformly distributed within [0.23, 0.41] at the next time point in 60% chance and the corresponding value in TS_3 is uniformly distributed within [0.05, 0.49] at the next time point in 40% chance. For C_3, values of TS_1 and TS_3 are uniformly distributed within [0, 1]; if TS_3's value falls into [0.03, 0.22], the corresponding value in TS_2 is uniformly distributed within [0.53, 0.92]; if TS_1's value falls into [0.70, 0.90] at every six time points, TS_4 takes on a value uniformly distributed within [0.13, 0.47] at the next time point. Figure 15 shows the MSTS of the 3 clusters with the implanted patterns highlighted. In this experiment, we set min_o to 10 based on the observation of the generated data.

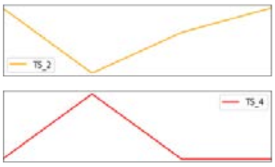
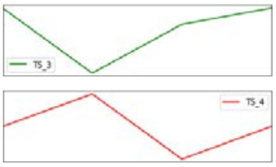
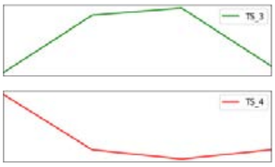
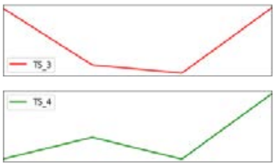
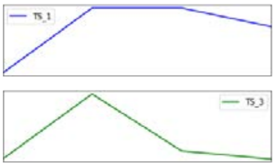
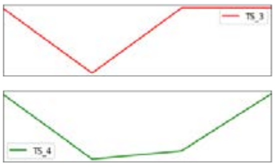
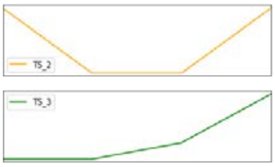
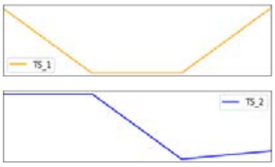
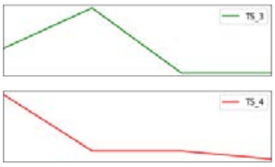


(a) MSTS in Cluster 1 (b) MSTS in Cluster 2 (c) MSTS in Cluster 3

Figure 15. Synthetic MSTS with Implanted Patterns Highlighted.

Table 17

Top Ranked Spatio-Temporal Patterns in Synthetic Data

Pattern			
Location	58, 59	1, 4	4, 8
Statistical Significance	7.935	7.91	7.28
Cluster	3	1	1
Pattern			
Location	7, 18	52, 57	40, 56
Statistical Significance	7.255	6.905	5.58
Cluster	1	3	3
Pattern			
Location	25, 28	47, 49	25, 39
Statistical Significance	5.085	5.075	3.255
Cluster	2	3	2

This set of experiments aims at demonstrating the ability in recovering the embedded spatio-temporal association patterns. A good pattern discovery algorithm should produce, not too many, high-quality patterns that can find the implanted associations. Therefore, to compare the number of generated association patterns, we use the approach developed by Tatavarty, Bhatnagar, and Young (2007), namely TARM, to generate temporal association patterns. With these generated temporal patterns, the STPM is constructed so we can discover spatio-temporal patterns for comparison. Secondly, we also believe that a good representation of the transformed data can lead the clustering algorithms to produce promising clustering results so we applied 3 different model-based approaches to transform the MSTS to STPM for cluster analysis.

After TARM is run and then the construction of the STPM, we detect the generated spatio-temporal association patterns and compare it with the one generated by the proposed algorithms. The proposed algorithm generated 73,587 temporal associations and completely captures the implanted temporal associations. The discovered temporal associations by the proposed approach and TARM are ranked by statistical significance and confidence respectively. TARM generated 1,177,424 temporal associations and only up to the 74,976th ranked association captures 12 occurrences of the implanted association. Therefore, the STPM based on the patterns generated from TARM is very sparse and is not feasible for further cluster analysis. The proposed method outperforms TARM by generating a smaller number of temporal associations with a better ranking of the discovered temporal patterns that cover the embedded ones. We go on searching the spatio-temporal associations from the STPM. For the proposed approach, 710 spatio-temporal association patterns are detected. Ranked by the statistical significance, the top 9 patterns are depicted in Table 17 for reference. For the top 399 out of 710 patterns, it is interesting to note that none of them are associated with different clusters so they are likely to describe the characteristics and the cluster relationship without relying on too much on the spatial contiguity information. For the STPM prepared by TARM's temporal patterns, it detected 11,360 spatio-temporal associations. Among these patterns, only up to the 224th ranked pattern is associated with the same cluster.

To evaluate the performance by clustering the generated STPM, given known cluster membership, we can calculate i) F1-measure, defined as $F\left(C_x, C_y\right) = \frac{2R\left(C_x,C_y\right)P\left(C_x,C_y\right)}{R\left(C_x,C_y\right)+P\left(C_x,C_y\right)}$, where $R\left(C_x, C_y\right) = \frac{cnt_{C_x,C_y}}{cnt_{C_x}}$, $P\left(C_x, C_y\right) = \frac{cnt_{C_x,C_y}}{cnt_{C_y}}$, cnt_{C_x,C_y} is the count of records with cluster label C_x in assigned cluster C_y, cnt_{C_x} is the count of records with cluster label C_x, and cnt_{C_y} is the count of records in the assigned cluster C_y , and ii) clustering accuracy, defined as $CA = \frac{\sum_{i=1}^{k} cnt_{c^i}}{Total\ number\ of\ records}$, where cnt_{c^i} is the count of records in *ith* cluster and k is the number of clusters.

To compare to the state of the art algorithms, we applied 3 model-based approaches to transform the MSTS to STPM, namely a) PCA and ARMA, b) PCA and equal frequency binning and c) lift ratio, and then perform clustering using *k*-means and HMM. PCA is used for feature extraction. These features are a) modeled by ARMA to calculate the top 8 LPC coefficients or b) discretized into 3 intervals to form STPM. STPM is clustered by *k*-means (k = 3) using Euclidean distance or by HMM using *log-likelihood* value. We also applied d) a deep learning model based on CNN, which does not require data transformation, described and implemented in section 4.8.1 to classify the MSTS by inputting the cluster label given the known cluster membership in this synthetic data set. Neural network classifier can be modified to fit a clustering model if a clustering algorithm such as *k*-means generates the cluster label initially and be it fed into the neural network's input layer. We split the data along the spatial dimension into 90 percent training set and 10 percent testing set. The results reported in Table 18 show the proposed approach outperforms the others. The first 2 approaches result in poor performance due to high information loss after dimensionality reduction. In c), the relationship between time series is considered so the hidden intra and inter patterns are revealed, significantly boosting the performance. In d) given the ground truth cluster label as an input for CNN to learn a classifier, it is very robust to capture the temporal relationship that can yield better accuracy than a) and b). However, there is still a room for improvement such as integrating different clustering methods into the network architecture while most existing deep learning based clustering techniques have separate feature learning (via deep learning) and clustering (with traditional clustering methods). A recent attempt (Tian, Zhou & Guan, 2017) is to simultaneously learn feature representation and does cluster assignment under the same deep learning framework on handwritten digit data sets and text data sets. However, it is not yet able to apply on multivariate spatial time series data. The proposed approach in e) not only considers important local associations between patterns temporally but also characterizes the space using only significant patterns globally, to produce high-quality matrix representation. Therefore, the noise from less significant patterns is filtered effectively and Pearson correlation coefficient, rather than Euclidean distance, is used for more effective clustering. Utilizing the information provided by the spatial penalty term, it also optimizes the proposed clustering algorithm to cluster the objects that share the common boundaries. Also, patterns hidden in each cluster are explicitly revealed and presented for easy interpretation even by a layperson.

Table 18

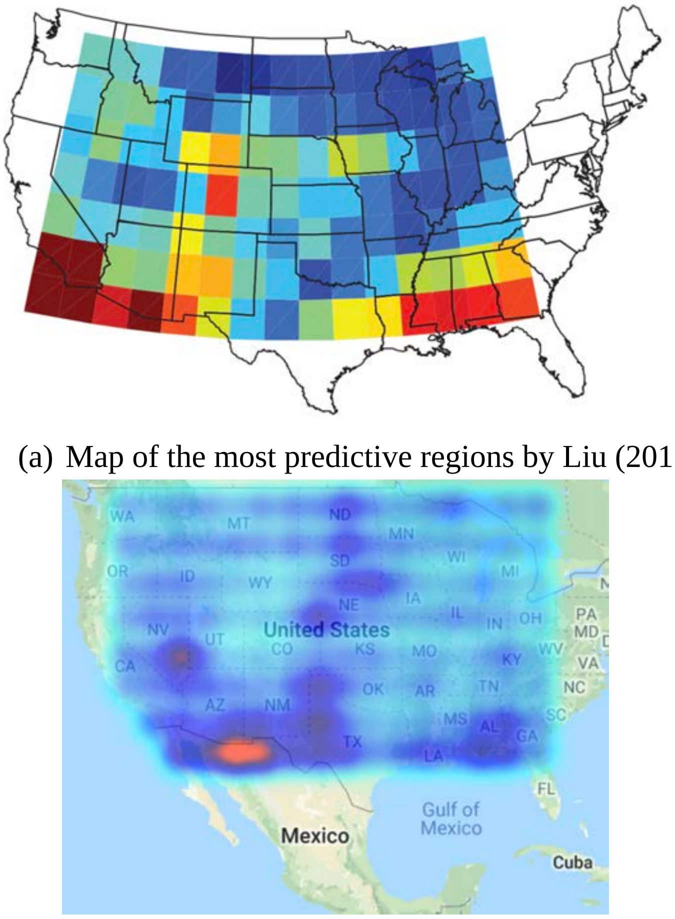
Modeling and Clustering Comparison

Approach	**F1-measure**	CA
a) PCA + ARMA + k-means	0.503	51.1%
b) PCA + binning + HMM	0.428	46.67%
c) Lift ratio + k-means	0.792	86.67%
d) CNN	0.761	73.89%
e) Proposed	0.833	92.38%

5.8.2 North America comprehensive climate data set.

Comprehensive climate data set (CCDS) is a collection of climate records with 125 observation locations using a 2.5 × 2.5 degree grid for latitudes in (30.475, 50.475) and longitudes in (−119.75, −79.75) from North America that contains monthly observations of 18 variables, including carbon dioxide (CO_2), methane (CH_4), carbon monoxide (CO), hydrogen (H_2), wet days (WET), cloud cover (CLD), vapor (VAP), precipitation (PRE), frost days (FRS), diurnal temperature range (DTR), minimum temperature (TMN), average temperature (TMP), maximum temperature (TMX), global solar radiation (GLO), extraterrestrial radiation (ETR), extraterrestrial normal radiation (ETRN), direct solar radiation (DIR), UV aerosol index (UV), spanning from 1990 to 2002. The data set can be obtained from Liu (2015) and is a representative data set used in the spatial data mining field. In particular, the predictability of these variables has been revealed in the literature. These variables can be categorized as primary climate variables, and some human and natural variables such as solar irradiance and greenhouse gases as these are known to affect the climate. For further detail of these variables, please refer to this article (Lozano, Li, Niculescu-Mizil, Liu, Perlich, Hosking & Abe, 2009).

In this experiment, we would like to extract the spatio-temporal patterns to characterize the CCDS by the STPM and then plot it on map (Figure 16) by the discovered patterns' statistical significance in order to demonstrate the ability to visualize them on different locations according to the strength of these patterns hidden in their climate measurements. Based on domain knowledge, we understand that past values of climate measurements in some specific locations to predict the future values of other time series are more predictive than the others. As a qualitative study, we will compare the strength of the discovered patterns by the proposed approach with the map of most predictive regions produced by Liu (2015). For the parameter setting, we set min_o to 5 based on the size of the data set. We applied the proposed algorithms to form the STPM and extracted 31,256 spatio-temporal patterns to represent the locations. To plot the map of most discriminative regions, we also define the aggregate pattern statistical significance of each region as $\mathrm{d}'(i) = \sum_{j}^{|SP|^i} \left| d_{SP\,j} \right|$ where $|SP|^i$ is the total number of patterns belonging to region i and $d_{SP\,j}$ is the statistical significance of the j^{th} pattern in region i. Based on the ground truth, two regions are believed to be the most predictive regions that are the southwest and the southeast. The southwest region reflects the impact of the Pacific Ocean. Both color maps show the color level of the southwest region, deep red in (a) and deep blue in (b), is stronger than other neighborhood regions. The southeast region frequently experiences relative sea level rise, hurricanes, and storm surge in the Gulf of Mexico. Again, on both color maps, the color level of southeast region suggests the past values are able to predict the future values of climate measurements and its discovered patterns are strongly discriminative than the neighborhood regions. The proposed approach can also reveal the interesting region in Colorado with both color maps showing high intensity of the values. It is because the Rocky Mountain valleys act as a funnel for winds from the west to provide locally divergent wind patterns. In short, this experiment can confirm the findings from the literature. It indicates the meaningfulness of the discovered spatio-temporal patterns that quantifies the temporal dependence between variables across different locations.



(a) Map of the most predictive regions by Liu (2015)

(b) Map of the most discriminative regions by the proposed algorithm

Note. In (a), red indicates highly predictive whereas blue indicates lowly predictive. In (b), deep light blue indicates weekly discriminative whereas deep blue and red indicate strongly discriminative.

Figure 16. Map of the Most Predictive Regions by Liu (2015) and Map of the Most Discriminative Regions by the Proposed Algorithm from CCDS.

5.8.3 Case study on Greater Bay Area (GBA) meteorological data set.

Meteorological data set (MET) is a collection of meteorological records taken from 5 different surface stations in Greater Bay Area (GBA) in China that contains hourly observations over a one-year-long period of 8 attributes, including total cloudiness, lower cloudiness, dry bulb temperature, dew point temperature, relative humidity, site pressure, wind direction and wind speed. These 5 surface stations denoted by the alphabets S = {A, B, C, D, E} are located in the great urban region of GBA in China as shown in *Figure 17*. Station A, B, C, D, and E is Guangzhou metropolis, Foshan city, Shenzhen city, Dongguan city and Zhongshan city respectively. The description of data collected by each station and some example data is listed in Table 19 and Table 20 respectively. Based on the domain knowledge from Wong et al. (2010), all those meteorological variables have an internal relationship according to the geographic location of the surface stations and might be governed by local terrain and land use.

Table 19

Data Description of MET

Attribute	Name	Type
S1	Total Cloudiness	Discrete
S2	Lower Cloudiness	Discrete
S3	Dry Bulb Temperature	Continuous
S4	Dew Point Temperature	Continuous
S5	Relative Humidity	Continuous
S6	Site Pressure	Continuous
S7	Wind Direction	Continuous
S8	Wind Speed	Continuous

Note. S = {A, B, C, D, E} corresponds to a set of 5 surface stations:

A = Guangzhou Metropolis; B = Foshan City; C = Shenzhen City;

D = Dongguan City; E = Zhongshan City.

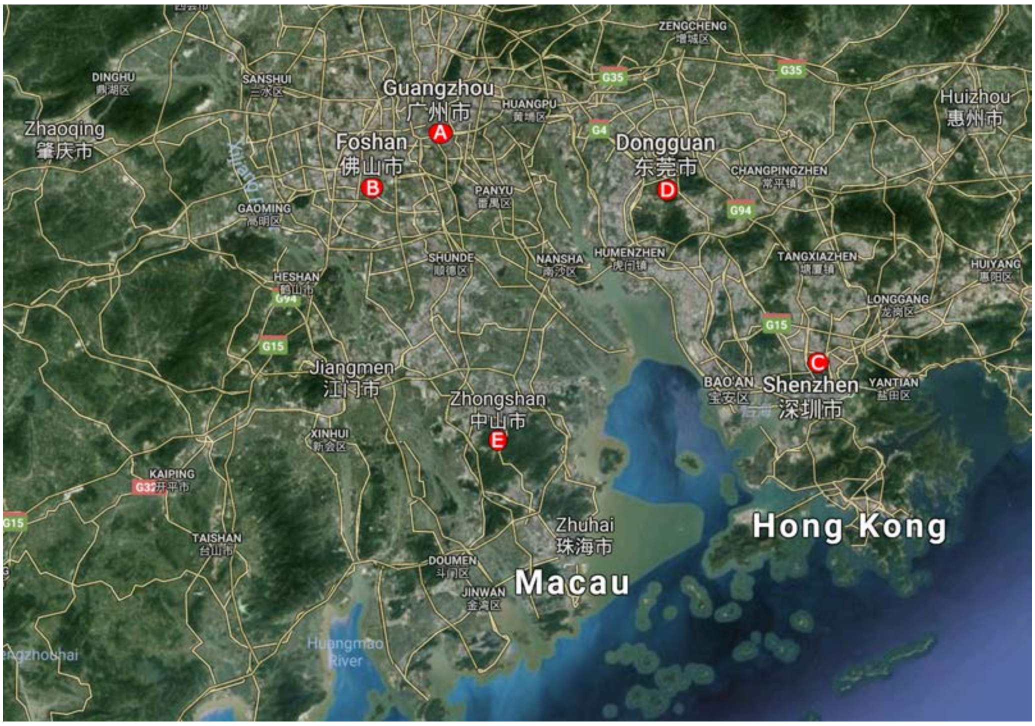


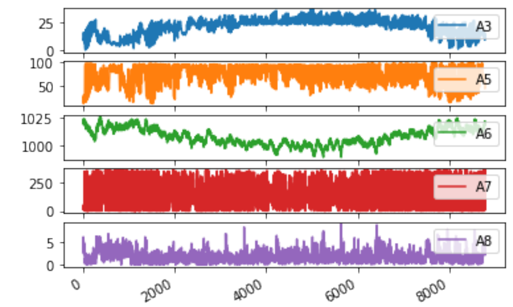
Figure 17. Great Urban Region of Greater Bay Area (GBA) in China.

Table 20

Some Example Data of MET

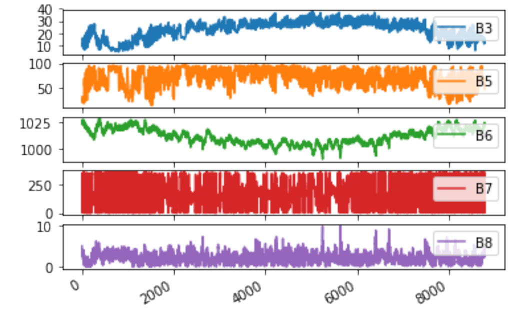
Surface Station A (Guangzhou)

A1	A2	A3	A4	A5	A6	A7	A8
J	J	10	9999	22	1021	23	3
J	J	9.6	9999	19	1021	24	3.8
J	J	9.3	9999	20	1021	20	3.4



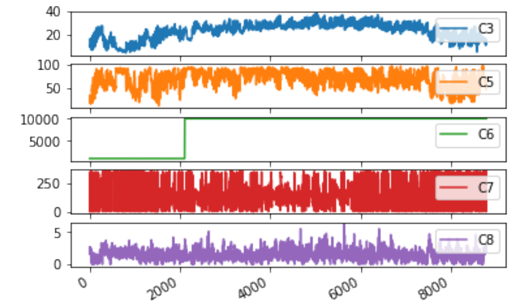
Surface Station B (Foshan)

B1	B2	B3	B4	B5	B6	B7	B8
A	A	11.2	9999	30	1024	17	5
A	A	10.8	9999	27	1024	6	4.5
A	A	10.2	9999	26	1024	9	3.6



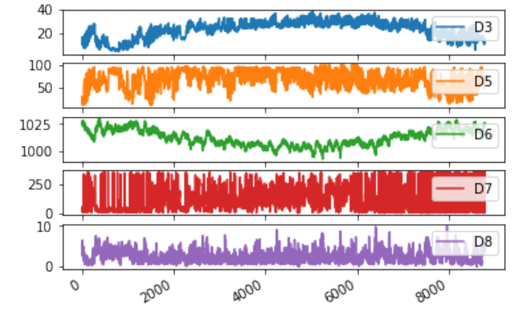
Surface Station C (Shenzhen)

C1	C2	C3	C4	C5	C6	C7	C8
D	D	10.2	9999	35	1024	0	1.7
D	D	9.7	9999	32	1022	6	1.8
D	D	9.5	9999	29	1021	2	2.4



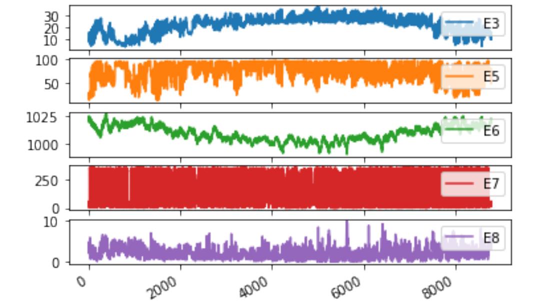
Surface Station D (Dongguan)

D1	D2	D3	D4	D5	D6	D7	D8
J	J	10.9	9999	31	1025	35	2.9
J	J	10.4	9999	30	1025	43	3
J	J	10.2	9999	31	1025	31	3.3



Surface Station E (Zhongshan)

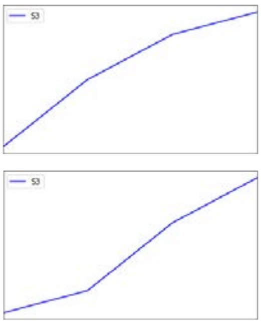
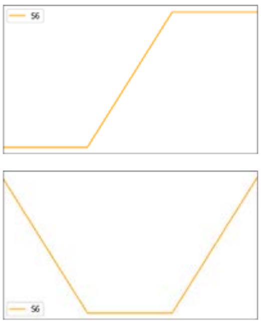
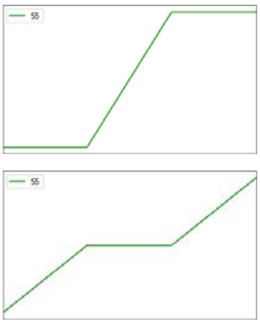
E1	E2	E3	E4	E5	E6	E7	E8
A	A	8.8	9999	28	1022	17	2.5
A	A	8.4	9999	28	1022	23	2.8
A	A	8.2	9999	28	1022	26	3.2

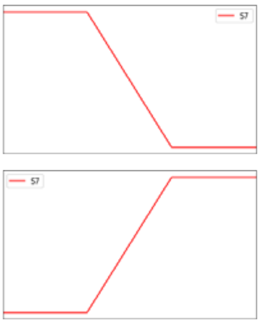
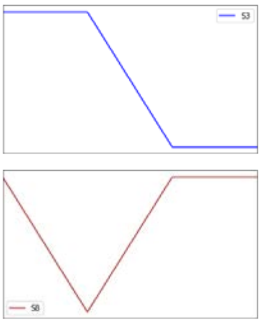


We first removed attribute 2 for all surface stations as it is a redundant attribute of attribute 1. Attribute 4 for all surface stations are also removed as all of them have identical values. In this case study, we set min_o to 300 based on the observation of the data set. The proposed algorithms extracted 31,955 spatio-temporal patterns. Ranked by their statistical significance, we are able to reveal the patterns that are consistent to the attribute clustering from Wong et al. (2010) in which the top ranked patterns can distinguish the attribute cluster items. The attribute clustering by Wong et al. (2010) and Wu, Chan and, Wong (2011) is able to identify the interdependence between attributes but is not able to make the temporal dependence hidden in the attributes across spaces explicit. The proposed algorithms can further uncover the movement of the values in the spatio-temporal dimension. Table 21 shows the top ranked spatio-temporal patterns for each attribute group formed according to Wong et al. (2010). From the clusters labeled by domain experts, we understand that the grouping is based on the interdependence among similar characteristics of attributes within each cluster formed. The 2nd, 3rd and 4th patterns in which the location and attribute are exactly matched with the attribute clustering items reflect the regional (global) characteristics of the correlated meteorological parameters. For instance, an intra pattern in “A7” sensor for detecting wind direction in Guangzhou station significantly co-occurs in that in Foshan station. The 1st and 5th patterns reflect the local characteristics that are significantly influenced by the local geographical feature such as land use and land coverage. The 5th pattern is an inter pattern that consists of 2 different sensors for detecting dry bulb temperature and wind speed in Dongguan city and Zhongshan city. Comparing the values of statistical significance of the top 5 patterns, stations A, B and, C are in a relatively stronger position than station D and E for the weather condition analysis. We normalize the aggregate pattern statistical significance defined in section 5.8.2 by the number of regions that also contain the patterns from region i, that is normalized aggregate pattern statistical significance $\widetilde{d'(i)} = \frac{d'(i)}{n^i}$ where $d'(i)$ is the aggregate pattern statistical significance which is the sum of all patterns' statistical significance in region i and n^i is the number of regions that also contain the patterns from region i. The normalized aggregate pattern statistical significance for stations A, B, C, D, and E is 17.51, 17.51, 15.95, 14.87, and 14.87 respectively. This finding is corresponding to the claim in the work by Wong et al. (2010) that the representative attributes in these attribute clusters are from only stations A, B and C.

Table 21

Top Ranked Spatio-Temporal Patterns for each Attribute Group of MET

Pattern			
Location	A, B, C, D, E	A, B, D, E	A, B, C, D, E
SS	23.744	22.195	14.548
ACI	**A3**, A4, C6, **B3**, **C3**, **D3**, **E3**, A8, B8, C8, D8, E8 (Dry Bulb Temperature & Wind Speed)	**A6**, **B6**, **D6**, **E6** (Site Pressure)	**B5**, **A5**, **C5**, **D5**, **E5** (Relative Humidity)

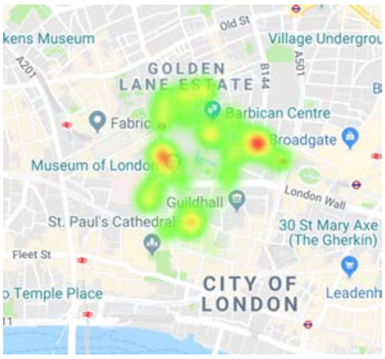
Pattern		
Location	A, B, C, D, E	D, E
SS	9.554	4.3
ACI	**C7**, **A7**, **B7**, **D7**, **E7** (Wind Direction)	A3, A4, C6, B3, C3, **D3**, **E3**, A8, B8, C8, **D8**, **E8** (Dry Bulb Temperature & Wind Speed)

Note. SS – Statistical Significance. ACI – Attribute Cluster Items by Wong et al. (2010). Bold items indicate the spatio-temporal pattern covering the attribute cluster items.

5.8.4 Case study on London crime data set.

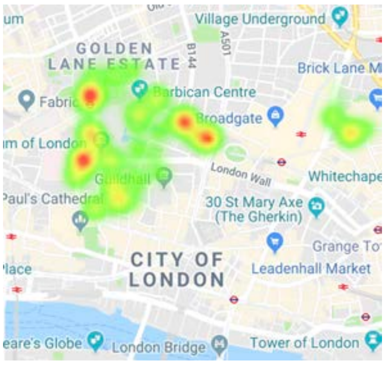
This section presents a case study on how our proposed approach has been used to assist police officers to analyze crime data. In the United Kingdom, police reported crimes are made publicly available via the police.uk website. The London crime data set consists of monthly occurrences of 14 crime types, including anti-social behavior, bicycle theft, burglary, criminal damage and arson, drugs, other crime, other theft, possession of weapons, public order, robbery, shoplifting, theft from the person, vehicle crime, violence and sexual offences, across 35 locations for latitudes in (51.499, 51.531) and longitudes in (−0.227, 0.554) in City of London spanning from September 2012 to August 2017. These 35 locations are formed according to the Lower Layer Super Output Area (LSOA) which is a geographic area built from groups of contiguous clusters of adjacent unit postcodes in the United Kingdom and is automatically generated to be as consistent in population size as possible. The LSOA codes and names are maintained by the Office for National Statistics in the United Kingdom. We will use the LSOA name to label the area in this case study.

In this case study, we set min_o = 5 and $|\varepsilon|$ = 3 based on the size of the data set. The proposed algorithms discovered 9,041 significant spatio-temporal association patterns in the study area. Ranked by the patterns' statistical significance, we plot the color map of the top 5 ranked patterns based on the values of the aggregate pattern statistical significance in Figure 18. The intensity of the color reflects the strength of the discovered patterns in the respective areas. From the color plot, it is obvious to identify some strong patterns as indicated by the red and yellow color. To further demonstrate the insight, we plot the transformed time series and highlight the association for the 1st ranked spatio-temporal pattern in Figure 19. This pattern consists of 2 crime types, violence and sexual offences as well as vehicle crime, across 4 LSOAs, namely City of London 001A, City of London 001B, City of London 001C and Tower Hamlets 015B. It reveals an interesting phenomenon that in these areas a moderate number of occurrences of vehicle crime and a low number of occurrences of violence and sexual offences are significantly correlated. By looking into the figures ending June 2017 from a report published by the Office for National Statistics ("Crime in England and Wales", 2017), we found that vehicle-related thefts have increased in the City of London but sexual offences declined. This kind of insight suggests police force work closely with partners in these areas with enforcement activity.



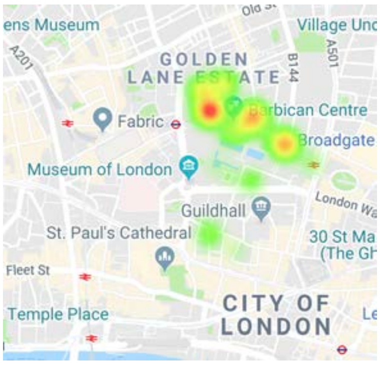
(a) Crime type: violence and sexual offences, vehicle crime

LSOA: City of London 001A, City of London 001B, City of London 001C, Tower Hamlets 015B



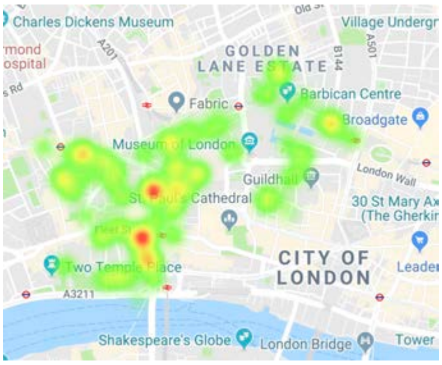
(b) Crime type: other theft, vehicle crime

LSOA: City of London 001A, City of London 001B, City of London 001C, Tower Hamlets 015B



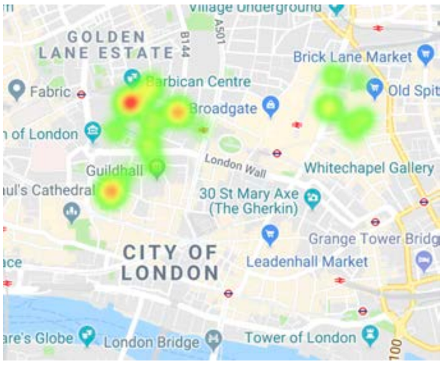
(c) Crime type: theft from the person, vehicle crime

LSOA: City of London 001B, City of London 001C



(d) Crime type: public order, vehicle crime

LSOA: City of London 001B, City of London 001C, City of London 001G



(e) Crime type: criminal damage and arson, vehicle crime

LSOA: City of London 001B, Tower Hamlets 015B

Note. Red color indicates the pattern occurrence is significantly frequent and the spatio-temporal association pattern is strongly discriminative. Green color indicates the pattern occurrence is also significantly frequent but the pattern is weakly discriminative. Yellow color indicates the pattern's strength is between those in red and green.

Figure 18. The Color Map Plot of Top 5 Ranked Patterns from London Crime Data.

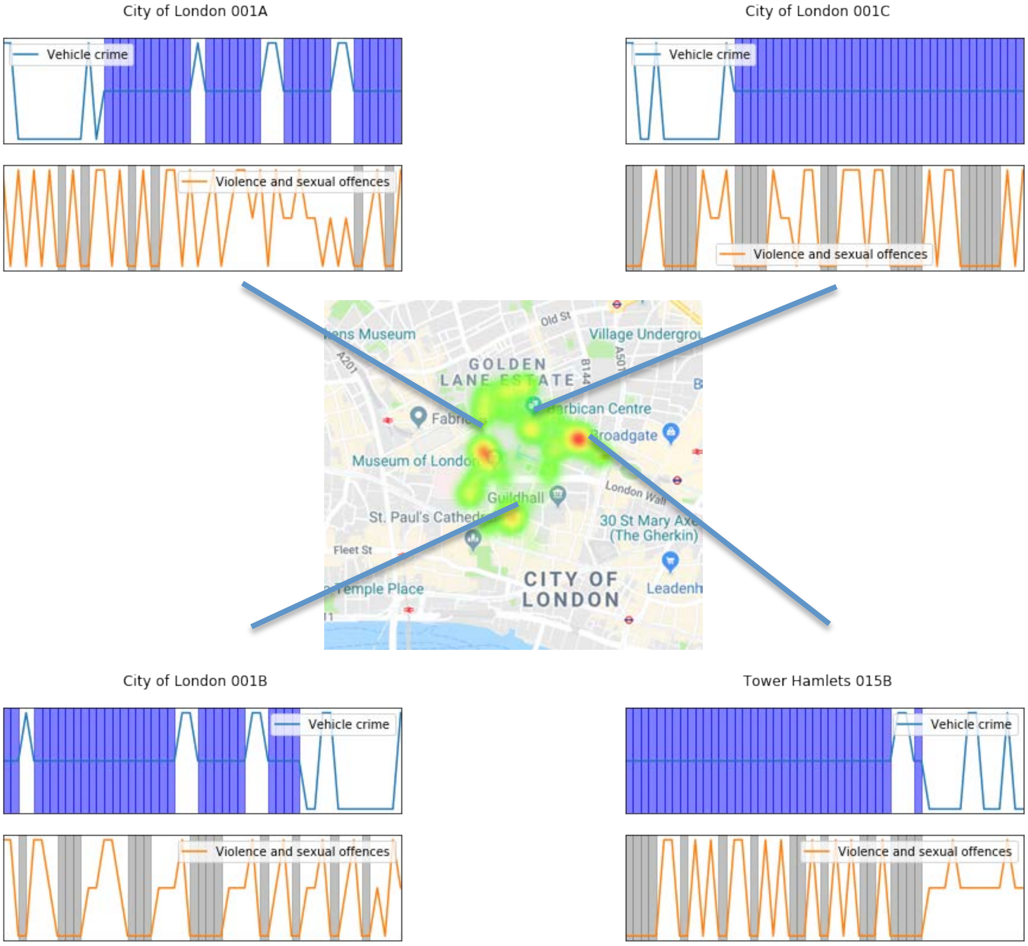


Figure 19. The 1st Ranked Spatio-Temporal Association in London Crime Data.

5.9 Summary of the Chapter

This chapter proposed a pattern mining approach to discover spatio-temporal associations for multivariate spatial time series data. These associations capture patterns occurring sequentially with varying time delays in single and multiple time series across spaces. The matrix representation for the transformed MSTS stores important spatio-temporal pattern information to reveal the statistical significance, available for further clustering and classification. We have performed both an experimental and case study using the proposed approach with a synthetic data set and real data sets. The proposed approach is able to uncover the embedded patterns in the synthetic data set and reveal the clustering relationship effectively. The experimental results using real data sets have been consistent with the findings reported by the literature and show the proposed approach is able to uncover the relations among time series in multiple locations. The findings in the case studies may also provide previously unknown relations that could introduce interesting insight for subject matter experts. In addition, an aggregate statistical significance is introduced for ranking and visualizing the discovered patterns.

This page is intentionally left blank.

6 CONCLUSION

In this thesis, we presented several techniques for spatial trajectory data and multivariate spatial time series data based on pattern discovery approach. Each technique is tested against synthetic and real data sets and compared to the state of the art algorithms for performance analysis. Our experimental results show that the proposed approaches outperform the others most of the time in many situations where class information is unavailable. When class information is available, pre-processing in discretization can utilize the class dependence on the attributes to partition the attributes and subsequently, this can boost the performance of classification in terms of accuracy.

For spatial trajectory, we first presented the feature generation and discretization methods that extract some basic, advanced and derived features and discretize these features for pattern discovery using simple rule-based, unsupervised and supervised discretization techniques. Empirically, we showed that these generated features are able to assist in building good classifiers that can effectively predict the class attribute in terms of accuracy. To discover high order patterns for spatial trajectory, we utilize the generated feature matrix to perform association pattern discovery. Higher order association patterns are generated based on statistically significant lower-order association patterns to avoid the exhaustive search. The significance of the association of patterns is evaluated by a statistical significance test by adjusted residual measure. Since each pattern is assigned an adjusted residual value, we can rank them according to their values and use them to train a classifier based on the rule set that is to use the discovered patterns to predict the class attribute by weight of evidence measure. As each rule is assigned a weight of evidence value, summing the weights of evidence of rules of the same class yields the total weight of evidence for which the class with the highest total weight of evidence to be assigned to the unseen data record. The adjusted residual value and weight of evidence value suggest how statistically significant the occurrence of the discovered patterns is and how much these patterns contribute to the predictiveness of the class attribute. Consequently, top-ranked patterns and rules can be chosen as the selected features. Our extensive experiments on real-world data sets indicate that the proposed techniques outperform other feature selection methods, such as basic feature (BF), advanced feature (AF) and deep neural network (DNN) on trajectory coordinate and time, in terms of the classification accuracy. Moreover, the processing time is linear to the size of the data set.

For multivariate spatial time series, a simple but effective feature matrix representation has been developed. The technique first vectorizes each multivariate time series of all regions by a temporal association pattern discovery algorithm to be used as features for clustering and classification and produces a spatio-temporal pattern matrix (STPM). The temporal association patterns are the statistically significant associations of frequent sequential patterns after multiple time series are transformed by SAX into multiple sequences. Since the association is detected by a statistical significance test based on adjusted residual, the temporal patterns are statistically significant if their values of adjusted residual are greater than the significance threshold at a certain confidence level. Based on the test, the proposed pattern mining algorithm eliminates temporal patterns that fail the test and retains the ones passing the test as the features for clustering and classification. The re-clustering technique summarizes the local clustering information by a classification algorithm that accounts for the global information using the weight of evidence measure to assess how the information of interesting spatio-temporal patterns contributes to the predictiveness of the class attribute. The experimental results on a synthetic data set show that the embedded patterns can be uncovered from the generated feature matrix and the feature matrix that is used for training a cluster model consistently produces high-quality clustering results in terms of accuracy computed after restoring the ground truth. Moreover, the case study results find some important insights that are commensurate to the literature and survey report. Empirically, we showed that the proposed techniques that take spatial and temporal information into consideration during the mining process perform better than other mining techniques that ignore the temporal and spatial dependence in terms of classification accuracy and F1-measure.

6.1 Summary of Contributions

6.1.1 Theoretical contributions.

In this thesis, we have two primary theoretical contributions to the development and practice of the data mining technology in the proposed algorithms.

1. Theoretical framework for pattern discovery for spatial trajectory and multivariate spatial time series (Wu & Chan, 2017, 2018a, 2018b, 2018c): A systematic theoretical foundation is proposed for mining such data types. A transformation method based on feature extraction and selection using pattern discovery techniques is developed to produce a feature matrix to represent and to characterize the very complex original data structure. Thus, the theoretical framework proposed as a unified framework is able to model common data mining tasks, such as clustering and classification, by handling these two forms of spatio-temporal data.
2. Demonstration of the necessity of mining patterns for clustering and classification in big databases of spatial trajectory and multivariate spatial time series (Wu & Chan, 2017, 2018a, 2018b, 2018c): From experimental results on extensive experiments using both synthetic data set and real data sets, the thesis forms a basis with supportive evidence that in big database of these types of data, strongly correlated and frequently co-occurring events across spatio-temporal domains do exist to form statistically significant association patterns. This leads to our thought to utilize the information of how attributes of spatio-temporal types are naturally associated to conduct supervised and unsupervised learning. The revealed effective clustering and/or classification models trained based on the discovered interesting patterns reconfirms the class attribute

and patterns relationship by introducing the spatio-temporal pattern matrix and thus allowing for i) the optimization of the interdependence between attributes of patterns using attribute clustering, ii) further pattern analysis and model learning, and iii) knowledge representation and interpretation. This concept using the interesting patterns that are interpretable and human readable to summarize the complex data structure furnishes the overall solution framework to conduct further predictive analytics using the-state-of-the-art techniques possible. Not only does it contribute to state a new research problem, but it also leads to an area of research to develop computational solutions using a set of new algorithms to deal with these data mining tasks.

6.1.2 Methodological contributions.

There are two important methodological contributions in this thesis as follows.

1. A statistical approach to extract spatio-temporal patterns in spatial trajectory and multivariate spatial time series (MSTS) based on the statistical significance test (Wu & Chan, 2017, 2018a, 2018b, 2018c): Based on the discretized attributes extracted and transformed from the original data, an association discovery technique is incorporated to detect statistical significant temporal patterns with time delay. The spatio-temporal pattern matrix (STPM) that characterizes the raw data makes conducting clustering and classification analysis possible by a unified pattern discovery framework.
2. A flexible algorithmic approach to pre-process and post-process the transformed data (Wu & Chan, 2017, 2018a, 2018b, 2018c): In both

settings of supervised and unsupervised learning, the experiments have shown the effectiveness and flexibility of the proposed solution to cope with different situations. In practice, spatial trajectory and MSTS data may come with noise, missing data and incorrectly marked labels will mix with the data sets. In this thesis, different discretization methods are employed and implemented for the data pre-processing, i.e. MACA using the mode to drive the discretization in unsupervised learning while in supervised learning, use the class attribute to drive the discretization by optimizing the interdependency between the class attribute and the other attributes. In the post-processing, if data are unlabeled, an initial clustering using the-state-of-the-art clustering technique locally optimizes the assignment of cluster label and then the re-clustering step, or the classification step if data are labeled, makes use of the weight of evidence measure to quantify the information contributed by the interesting patterns of each category to globally predict the class membership.

6.1.3 Application contributions.

In summary, we contribute to the area of the application domain in four major aspects.

1. Discovery of transportation modes of commuters from GPS trajectories in China on a large scale real data set (Wu & Chan, 2018b): Discovering patterns from GPS trajectories is regarded as a classification problem when the transportation mode of each trip are annotated by users and solved as a pattern mining problem by transforming each trip into features for training a classifier. However, since the dimensionality of the spatial trajectory is very high, extracting discriminative and useful

features for further predictive analysis becomes difficult. For a database with a large number of spatial trajectories, we first pre-process each trajectory into a set of basic, advanced and derived features. Then, we apply MACA to break the feature set into groups. In each group, we use the representative feature, one with the highest interdependence with others in the group, to drive the discretization of the values of other features. Treating intervals as discrete events, association patterns can be discovered. To demonstrate the flexibility of the proposed method, we fed the discretized feature matrix into multiple classifiers for performance comparison. The results show the transformed feature matrix representation can provide high-quality data summary to characterize the original spatial trajectory data for building a classifier.

2. Discovery and prediction of driver telematics fingerprint using the mined driving behavior over multiple various length driving trips distinguishing the driver identity (Wu & Chan, 2017): Due to privacy concern, user location data in the moving object trajectory are to be anonymized before publishing. To classify the privacy-preserving driving trips in a set of recorded GPS tracks, an information theoretic approach to characterize them based on their occurrences of frequently detected patterns is developed and applied. The interesting patterns that are discovered through a statistical significance test are treated as driver telematics fingerprints for training a classifier using the weight of evidence measure. This application validates the classificatory power of the discovered interesting patterns. The result indicates the approach is effective and efficient in achieving a good accuracy in the prediction of the class labels

of the different driving trips with varying length based on the transformed set of attributes.

3. Discovery and ranking of meteorological pattern relationship from weather sensors and surface stations over two large study areas, through experiments on a) MET data from Greater Bay Area in China and b) Comprehensive Climate data from North America, rendering subtle relations for regional weather monitoring (Wu & Chan, 2018b): The discovery of the spatio-temporal patterns from the multivariate spatial time series weather data can reflect the regional and global characteristics of the correlated meteorological parameters. Based on domain knowledge, we understand that past values of climate measurements in some specific locations to predict the future values of other time series are more predictive than the others. To demonstrate the ability to visualize the predictability on different locations according to the strength of these patterns hidden in their climate measurements, we defined the aggregate pattern statistical significance of each region for plotting a heat map to show the intensity of the values for the prediction. The result indicates the meaningfulness of the discovered spatio-temporal patterns that quantifies the temporal dependence between variables across different locations. The top-ranked patterns are also found to be the representative characteristics of the correlated meteorological parameters that are consistent with the literature.
4. Discovery and grouping of interesting crime occurrence patterns in multiple London districts revealing criminal characteristics (Wu & Chan, 2018b): This application described how our proposed approach has been

used to assist police officers to analyze crime data. The pattern discovery and grouping experiment on a large set of a publicly available London crime data set yields some important relationships among crime types and groups of contiguous clusters of adjacent unit postcodes in the United Kingdom. From identified spatio-temporal patterns, some interesting phenomena, such as different levels of occurrences of some crime types strongly correlated across regions, are highlighted. These useful findings, including vehicle-related thefts increased in the City of London while sexual offenses declined, correspond to the national statistics report published by the government department. Such findings show the usefulness and effectiveness of the proposed method in revealing subtle crime patterns for the suggestion of police force working closely with partners in these areas with enforcement actions.

6.2 Future Work

There are several promising directions for our future work in developing effective methods for different components of spatial trajectory and MSTS data analysis. The future development of the individual components is listed below.

In spatial trajectory, if a moving object can be labeled as a good or bad object per their properties, what is the characteristic of the movement of them? This can be answered by analyzing the interesting patterns of multiple good objects. One of the current trends in computational movement data analysis is to relate the movement to its embedding context (Laube, 2014; Long, Weibel, Dodge & Laube, 2018). In this direction, we can extend the feature matrix to model also the embedding context for context-aware pattern mining for selected application problems.

After showing our techniques perform better than other techniques, we can explain why our techniques work better in order to explore the theoretical analysis of our techniques. It is anticipated that the next step is to generate models and knowledge for further exploration of the new data type if we get more insights into why our techniques perform well.

Another direction is to generalize for other comparable techniques to handle more forms of spatio-temporal data, such as multimedia spatio-temporal data stream, in particular, to generate the feature matrix structure that can effectively and efficiently represent and retrieve the data without losing too much of information from the raw data for further analysis.

7 References

Agrawal, R., & Srikant, R. (1994, September). *Fast algorithms for mining association rules*. Paper presented at the International Conference on VLDB (pp. 487-499).

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). *Automatic subspace clustering of high dimensional data for data mining applications*. Paper presented at the International Conference of the ACM SIGMOD (pp. 94-105). ACM.

Agrawal, R., Ghosh, S., Imieliński, T., Iyer, B., & Swami, A. (1992, August). *An interval classifier for database mining applications*. Paper presented at the International Conference on VLDB (pp. 560-573).

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, *22*(2), 207-216.

Agresti, A. (2003). *Categorical data analysis*. John Wiley & Sons.

Aleksander, I., & Morton, H. (1990). *An introduction to neural computing*. London: Chapman & Hall.

Alelyani, S., Tang, J., & Liu, H. (2013). Feature selection for clustering: A review. In C. Aggarwal & C. Reddy (Ed.), *Data Clustering: Algorithms and Applications* (pp. 110-121). Chapman & Hall/CRC.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745-6750.

Amiri-Simkooei, A. R. (2009). Noise in multivariate GPS position time-series. *Journal of Geodesy*, *83*(2), 175-187.

Ashbrook, D., & Starner, T. (2002). *Learning significant locations and predicting user movement with GPS*. Paper presented at the Sixth International Symposium on Wearable Computers (pp. 101-108). IEEE.

Au, W. H., Chan, K. C., Wong, A. K., & Wang, Y. (2005). Attribute clustering for grouping, selection, and classification of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*(2), 83-101.

Baecke, P., & Bocca, L. (2017). The value of vehicle telematics data in insurance risk selection processes. *Decision Support Systems*, *98*, 69-79.

Beale, R., & Jackson, T. (1990). *Neural Computing-an introduction*. CRC Press.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems* (pp. 153-160).

Bittorf, M. K. A. B. V., Bobrovytsky, T., Erickson, C. C. A. C. J., Hecht, M. G. D., Kuff, M. J. I. J. L., Leblang, D. K. A., ... & Yoder, M. M. (2015). *Impala: A modern, open-source SQL engine for Hadoop*. Paper presented at the 7th Biennial Conference on Innovative Data Systems Research.

Bonato, P., Mork, P. J., Sherrill, D. M., & Westgaard, R. H. (2003). Data mining of motor patterns recorded with wearable technology. *IEEE Engineering in Medicine and Biology Magazine*, *22*(3), 110-119.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Bu, Y., Howe, B., Balazinska, M., & Ernst, M. D. (2010). HaLoop: efficient iterative data processing on large clusters. *Proceedings of the VLDB Endowment*, *3*(1-2), 285-296.

Cao, Q., Bouqata, B., Mackenzie, P. D., Messier, D., & Salvo, J. J. (2009, October). *A grid-based clustering method for mining frequent trips from large-scale, event-based telematics datasets*. Paper presented at the IEEE International Conference on Systems, Man and Cybernetics (pp. 2996-3001). IEEE.

Chan, K. C., Ching, J. Y., & Wong, A. K. (1992, June). *A probabilistic inductive learning approach to the acquisition of knowledge in medical expert systems*. Paper presented at the Fifth Annual IEEE Symposium on Computer-Based Medical Systems (pp. 572-581). IEEE.

Chan, K. C., & Wong, A. K. (1990). APACS: A system for the automatic analysis and classification of conceptual patterns. *Computational Intelligence*, *6*(3), 119-131.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... & Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems*, *26*(2), 4.

Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2012). NbClust Package: finding the relevant number of clusters in a dataset. *UseR! 2012*.

Cheung, D. W., Ng, V. T., Fu, A. W., & Fu, Y. (1996). Efficient mining of association rules in distributed databases. *IEEE Transactions on Knowledge and Data Engineering*, *8*(6), 911-922.

Ching, J. Y., Wong, A. K., & Chan, K. C. C. (1995). Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(7), 641-651.

Chiu, D. K., Wong, A. K., & Cheung, B. (1991). Information discovery through hierarchical maximum entropy discretization and synthesis. *Knowledge Discovery in Databases*. AAAI Press.

Cios, K. J., & Kurgan, L. A. (2004). CLIP4: Hybrid inductive machine learning algorithm that generates inequality rules. *Information Sciences*, *163*(1-3), 37-83.

Cios, K. J., & Kurgan, Ł. A. (2002). Hybrid inductive machine learning: An overview of CLIP algorithms. *New Learning Paradigms in Soft Computing* (pp. 276-322). Physica-Verlag Heidelberg.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*(4), 261-283.

Coppi, R., D'Urso, P., & Giordani, P. (2010). A fuzzy clustering model for multivariate spatial time series. *Journal of Classification*, *27*(1), 54-88.

Crime in England and Wales: year ending June 2017. (2017, October). Retrieved from https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/bulletins/crimeinenglandandwales/june2017

Dasarathy, B. V. (1991). Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press.

Dash, M., Choi, K., Scheuermann, P. & Liu, H. (2002). *Feature selection for clustering - a filter solution*. Paper presented at the Second International Conference on Data Mining (pp. 115-122). IEEE.

Davis, L. (1991). Handbook of genetic algorithms. New York: Academic Press.

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1-38.

Dong, W., Li, J., Yao, R., Li, C., Yuan, T., & Wang, L. (2016). Characterizing driving styles with deep learning. *arXiv preprint arXiv:1607.03611*.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification (second edition). John Wiley & Sons. New York.

Endo, Y., Toda, H., Nishida, K., & Kawanobe, A. (2016, April). *Deep feature extraction from trajectories for transportation mode estimation*. Paper presented at Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 54-66). Springer, Cham.

Eldawy, A., & Mokbel, M. F. (2015, April). *The era of big spatial data*. Paper presented at 2015 31st IEEE International Conference on Data Engineering Workshops (pp. 42-49). IEEE.

Fayyad, U., & Irani, K. (1993). *Multi-interval discretization of continuous-valued attributes for classification learning*. Paper present at International Joint Conference on Artificial Intelligence (pp. 1022-1029). Chambery, France.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27-34.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*(2), 139-172.

Forgey, E. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classification. *Biometrics*, *21*(3), 768-769.

Frenzel, S., & Pompe, B. (2007). Partial mutual information for coupling analysis of multivariate time series. *Physical Review Letters*, *99*(20), 204101.

Fu, T. C. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, *24*(1), 164-181.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., & Cai, J. (2015). Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*.

Haberman, S. J. (1977). *The analysis of frequency data* (Vol. 4). University of Chicago Press.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504-507.

Holland, J. H. (1987). *Genetic algorithms and classifier systems: foundations and future directions* (No. LA-UR-87-1863; CONF-870775-1). Michigan Univ., Ann Arbor (USA).

Holland, J. H., & Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*. Massachusetts: Addison-Wesley.

Houtsma, M., & Swami, A. (1995, March). *Set-oriented mining for association rules in relational databases*. Paper presented at the Eleventh International Conference on Data Engineering (pp. 25-33). IEEE.

Hüsken, M., & Stagge, P. (2003). Recurrent neural networks for time series classification. *Neurocomputing*, *50*, 223-235.

Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68-75.

Kaufman, K. A., & Michalski, R. S. (1999, June). *Learning from inconsistent and noisy data: the AQ18 approach.* Paper presented at the International Symposium on Methodologies for Intelligent Systems (pp. 411-419). Springer, Berlin, Heidelberg.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis* (Vol. 344). John Wiley & Sons.

Kerber, R. (1992, July). ChiMerge: *Discretization of numeric attributes*. Paper presented at the Tenth National Conference on Artificial Intelligence (pp. 123-128). AAAI Press.

Kohonen, T. (2012). *Self-organization and associative memory* (Vol. 8). Springer Science & Business Media.

Krumm, J., & Horvitz, E. (2004, August). *LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths*. Paper present at the International Conference on Mobile and Ubiquitous Systems (pp. 4-13). Boston, Massachusetts.

Kurgan, L. A., & Cios, K. J. (2004). CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering, 16*(2), 145-153.

Kurgan, L. A., & Cios, K. J. (2003). *Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm*. Paper presented at the International Conference on Machine Learning and Applications (pp. 30-36).

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters, 42*, 11-24.

Laube, P. (2014). *Computational movement analysis*. Berlin, Germany: Springer.

Lavielle, M., & Teyssiere, G. (2006). Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, *46*(3), 287-306.

Le Cessie, S., & Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied Statistics*, 191-201.

Lee, H., Ekanadham, C., & Ng, A. Y. (2008). Sparse deep belief net model for visual area V2. In *Advances in neural information processing systems* (pp. 873-880).

Li, G. C. L. (2008). *Association pattern analysis for pattern pruning, pattern clustering and summarization* (Doctoral dissertation, The University of Waterloo).

Liao, L., Fox, D., & Kautz, H. (2004). *Learning and inferring transportation routines*. Paper presented at the Nineteenth National Conference on Artificial Intelligence (AAAI-04). AAAI Press.

Liew, C. S., Wah, T. Y., Shuja, J., & Daghighi, B. (2015). Mining personal data using smartphones and wearable devices: A survey. *Sensors*, *15*(2), 4430-4469.

Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, *15*(2), 107-144.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, *6*(4), 393-423.

Liu, H., & Setiono, R. (1997). Feature selection via discretization. *IEEE Transactions on Knowledge and Data Engineering*, *9*(4), 642-645.

Liu, L., Wong, A. K., & Wang, Y. (2004). A global optimal algorithm for class-dependent discretization of continuous data. *Intelligent Data Analysis*, *8*(2), 151-170.

Liu, Y. (2015). Scalable Multivariate Time-Series Models for Climate Informatics. *Computing in Science & Engineering*, *17*(6), 19-26.

Long, J. A., Weibel, R., Dodge, S., & Laube, P. (2018). Moving ahead with computational movement analysis. *International Journal of Geographical Information Science*, 32(7), 1275-1281.

Lozano, A. C., Li, H., Niculescu-Mizil, A., Liu, Y., Perlich, C., Hosking, J., & Abe, N. (2009, June). *Spatial-temporal causal modeling for climate change attribution*. Paper presented at the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 587-596). ACM.

Ma, P. C., Chan, K. C., & Chiu, D. K. (2005). Clustering and re-clustering for pattern discovery in gene expression data. *Journal of Bioinformatics and Computational Biology*, *3*(02), 281-301.

MacEachren, A. M., Wachowicz, M., Edsall, R., Haug, D., & Masters, R. (1999). Constructing knowledge from multivariate spatiotemporal data: integrating geographical visualization with knowledge discovery in database methods. *International Journal of Geographical Information Science*, 13(4), 311-334.

MacQueen, J. (1967, June). *Some methods for classification and analysis of multivariate observations*. Paper presented at the Berkeley Symposium on Mathematical Statistics and Probability (pp. 281-297).

Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., & Cheung, D. W. (2004, August). *Mining, indexing, and querying historical spatiotemporal data*. Paper presented at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 236-245). ACM.

Mardonova, M., & Choi, Y. (2018). Review of Wearable Device Technology and Its Applications to the Mining Industry. *Energies*, *11*(3), 547.

Melnik, S., Gubarev, A., Long, J. J., Romer, G., Shivakumar, S., Tolton, M., & Vassilakis, T. (2010). Dremel: interactive analysis of web-scale datasets. *Proceedings of the VLDB Endowment, 3*(1-2), 330-339.

Memisevic, R., & Hinton, G. (2007, June). Unsupervised learning of image transformations. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*(pp. 1-8). IEEE.

Michalski, R. S., Mozetic, I., Hong, J., & Lavrac, N. (1986). *The multi-purpose incremental learning system AQ15 and its testing application to three medical domains*. Paper presented at AAAI 1986.

Milligan, G. W. (1980). An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika, 45*(3), 325-342.

Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, *14*(5), 1003-1016.

Oates, T. (1999, August). *Identifying distinctive subsequences in multivariate time series by clustering*. Paper presented at ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 322-326). ACM.

Owsley, L. M., Atlas, L. E., & Bernard, G. D. (1997, April). *Automatic clustering of vector time-series for manufacturing machine monitoring*. Paper presented at 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (pp. 3393-3396). IEEE.

Pascual-Marqui, R. D. (2007). Instantaneous and lagged measurements of linear and nonlinear dependence between groups of multivariate time series: frequency decomposition. *arXiv preprint arXiv:0711.1455*.

Paterson, A., & Niblett, T. B. (1987). ACLS manual. *Edinburgh: Intelligent Terminals, Ltd*.

Patterson, D. J., Liao, L., Fox, D., & Kautz, H. (2003, October). *Inferring high-level behavior from low-level sensors*. Paper presented at the International Conference on Ubiquitous Computing (pp. 73-89). Springer, Berlin, Heidelberg.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226-1238.

Platt, J. C. (1999). 12 fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*, 185-208.

Poultney, C., Chopra, S., & Cun, Y. L. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems* (pp. 1137-1144).

Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, *38*, 48.

Quinlan, J. R. (1987, August). *Generating production rules from decision trees*. Paper presented at IJCAI (pp. 304-307).

Rabiner, L. R., & Juang, B. H. (1986). An introduction to hidden Markov models. *ieee assp magazine*, *3*(1), 4-16.

Ratcliffe, J. H. (2004). The hotspot matrix: A framework for the spatio-temporal targeting of crime reduction. *Police practice and research*, *5*(1), 5-23.

Rosén, C., & Yuan, Z. (2001). Supervisory control of wastewater treatment plants by combining principal component analysis and fuzzy c-means clustering. *Water Science and Technology*, *43*(7), 147-156.

Rousseeuw, P. J., & Kaufman, L. (1990). Finding groups in data. *Series in Probability & Mathematical Statistics 1990*, *34 (1)*, 111-112.

Sankoff, D. (1983). Time warps, string edits, and macromolecules. *The Theory and Practice of Sequence Comparison, Reading*.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

Shumway, R. H. (2014). Discrimination and clustering for multivariate time series. *Wiley StatsRef: Statistics Reference Online*.

Singhal, A., & Seborg, D. E. (2005). Clustering multivariate time-series data. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *19*(8), 427-438.

Sohn, T., Varshavsky, A., LaMarca, A., Chen, M. Y., Choudhury, T., Smith, I., ... & De Lara, E. (2006, September). *Mobility detection using everyday GSM traces*. Paper presented at the International Conference on Ubiquitous Computing (pp. 212-224). Springer, Berlin, Heidelberg.

Stonebraker, M., Abadi, D. J., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., ... & O'Neil, P. (2005, August). *C-store: a column-oriented DBMS*. Paper presented at the 31st International Conference on Very Large Data Bases (pp. 553-564). VLDB Endowment.

Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, *16*(1), 30-34.

Smyth, P., & Goodman, R. M. (1992). An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and data engineering*, *4*(4), 301-316.

Su, C. T., & Hsu, J. H. (2005). An extended chi2 algorithm for discretization of real value attributes. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 437-441.

Talavera, L. (1999, June). Feature selection as a preprocessing step for hierarchical clustering. Paper presented at *ICML* (pp. 389-397).

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, 37.

Tay, F. E., & Shen, L. (2002). A modified chi2 algorithm for discretization. *IEEE Transactions on Knowledge & Data Engineering*, (3), 666-670.

Tian, K., Zhou, S., & Guan, J. (2017, September). DeepCluster: A General Clustering Framework Based on Deep Learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 809-825). Springer, Cham.

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411-423.

Tou, J. T., & Gonzalez, R. C. (1974). Pattern recognition principles. *Reading, MA: Addison-Wesley*.

Tsai, C. J., Lee, C. I., & Yang, W. P. (2008). A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences*, *178*(3), 714-731.

Tsay, R. S., Peña, D., & Pankratz, A. E. (2000). Outliers in multivariate time series. *Biometrika*, *87*(4), 789-804.

Vedaldi, A., & Fulkerson, B. (2010, October). *VLFeat: An open and portable library of computer vision algorithms*. Paper presented at the 18th ACM International Conference on Multimedia (pp. 1469-1472). ACM.

Von Bünau, P., Meinecke, F. C., Király, F. C., & Müller, K. R. (2009). Finding stationary subspaces in multivariate time series. *Physical Review Letters*, *103*(21), 214101.

Wang, D. C. C., & Wong, A. K. (1979). Classification of discrete data with feature space transformation. *IEEE Transactions on Automatic Control*, *24*(3), 434-437.

Wang, Y., & Wong, A. K. (2003). From association to classification: Inference using weight of evidence. *IEEE Transactions on Knowledge and Data Engineering*, *15*(3), 764-767.

Wang, Y., & Wong, A. K. (1996, August). Representing Discovered Patterns Using Attributed Hypergraph. Paper presented at KDD (pp. 283-286).

Wong, A. K., & Chiu, D. K. (1987). Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6).

Wong, A. K., Chiu, D. K., & Huang, W. (2001). A discrete-valued clustering algorithm with applications to biomolecular data. *Information Sciences*, *139*(1-2), 97-112.

Wong, A. K., & Li, G. C. (2008). Simultaneous pattern and data clustering for pattern cluster analysis. *IEEE Transactions on Knowledge and Data Engineering*, *20*(7), 911-923.

Wong, A. K., & Wang, Y. (1997). High-order pattern discovery from discrete-valued data. *IEEE Transactions on Knowledge and Data Engineering*, *9*(6), 877-893.

Wong, A. K., & Wang, Y. (2003). Pattern discovery: a data driven approach to decision support. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 33*(1), 114-124.

Wong, A. K., Wu, B., Wu, G. P., & Chan, K. C. (2010, October). *Pattern discovery for large mixed-mode database*. Paper presented at the 19th ACM International Conference on Information and Knowledge Management (pp. 859-868). ACM.

Wu, G. P., & Chan, K. C. (2018a). *Clustering driving trip trajectory data based on pattern discovery techniques*. Paper presented at 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China.

Wu, G. P., & Chan, K. C. (2018b). *Discovery of spatio-temporal patterns in multivariate spatial time series*. Manuscript submitted for publication.

Wu, G. P., & Chan, K. C. (2018c). *Mining spatio-temporal patterns in multivariate spatial time series*. Paper presented at 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China.

Wu, G. P., & Chan, K. C. (2017). *Privacy-preserving trajectory classification of driving trip data based on pattern discovery techniques*. Paper presented at 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA.

Wu, G. P., Chan, K. C., & Wong, A. K. (2011, December). Unsupervised fuzzy pattern discovery in gene expression data. *BMC Bioinformatics,* 12(5), S5. BioMed Central.

Wu, G. P., Chen, Y. C., Zhu, W. Y., & Chan, K. C. (2013). *An Intelligent System for Effective Mobile Application Advertising*. Paper presented at 2013

Conference Technologies and Applications of Artificial Intelligence (TAAI), Taiwan.

Wu, Q., Bell, D., McGinnity, M., Prasad, G., Qi, G., & Huang, X. (2006, September). *Improvement of decision accuracy using discretization of continuous attributes*. Paper presented at the International Conference on Fuzzy Systems and Knowledge Discovery (pp. 674-683). Springer, Berlin, Heidelberg.

Xing, Z., Pei, J., & Keogh, E. (2010). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, *12*(1), 40-48.

Xu, Y. & Jin, Y. (2002). *U.S. Patent No. 6,401,027*. Washington, DC: U.S. Patent and Trademark Office.

Yang, K., & Shahabi, C. (2004, November). *A PCA-based similarity measure for multivariate time series*. Paper presented at the 2nd ACM International Workshop on Multimedia Databases (pp. 65-74). ACM.

Yoon, H., Yang, K., & Shahabi, C. (2005). Feature subset selection and feature ranking for multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, *17*(9), 1186-1198.

Yuan, G., Sun, P., Zhao, J., Li, D., & Wang, C. (2017). A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review*, *47*(1), 123-144.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., ... & Stoica, I. (2012, April). *Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing*. Paper presented at the 9th

USENIX Symposium on Networked Systems Design and Implementation (pp. 15-28). USENIX Association.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. *HotCloud*, *10*(10-10), 95.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996, June). BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Record*, 25(2), 103-114.

Zhao, W., Ma, H., & He, Q. (2009, December). *Parallel k-means clustering based on MapReduce*. Paper presented at the IEEE International Conference on Cloud Computing (pp. 674-679). Springer, Berlin, Heidelberg.

Zheng, Y. (2015). Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *6*(3), 29.

Zheng, Y., Chen, Y., Li, Q., Xie, X., & Ma, W. Y. (2010). Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web (TWEB)*, *4*(1), 1.

Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W. Y. (2008, September). *Understanding mobility based on GPS data*. Paper presented at the 10th International Conference on Ubiquitous Computing (pp. 312-321). ACM.

Zheng, Y., Liu, L., Wang, L., & Xie, X. (2008, April). *Learning transportation mode from raw GPS data for geographic applications on the web*. Paper presented at the 17th International Conference on World Wide Web (pp. 247-256). ACM.

Zhou, P. Y., & Chan, K. C. (2014, May). *A model-based multivariate time series clustering algorithm*. Paper presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 805-817). Springer, Cham.

Zhou, P. Y., & Chan, K. C. (2015, May). *A feature extraction method for multivariate time series classification using temporal patterns*. Paper presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 409-421). Springer, Cham.

Zhu, Y., Zheng, Y., Zhang, L., Santani, D., Xie, X., & Yang, Q. (2012). Inferring taxi status using GPS trajectories. *arXiv preprint arXiv:1205.4378*.

Zhuang, D. E., Li, G. C., & Wong, A. K. (2014). Discovery of temporal associations in multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, *26*(12), 2969-2982.

Zuur, A. F., Fryer, R. J., Jolliffe, I. T., Dekker, R., & Beukema, J. J. (2003). Estimating common trends in multivariate time series using dynamic factor analysis. *Environmetrics: The official Journal of the International Environmetrics Society, 14*(7), 665-685.