University of Alberta

MINING STATISTICALLY SIGNIFICANT TEMPORAL ASSOCIATIONS IN MULTIPLE EVENT SEQUENCES

by

Han Liang

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

©Han Liang
Spring 2013
Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Abstract

We propose a two-phase method, called Multivariate Association Discovery (MAD), to mine temporal associations in multiple event sequences. It is assumed that a set of event sequences has been collected from an application, where each event has an id and an occurrence time. The goal is to detect temporal associations of events whose frequencies in the data are statistically significant. The motivation of our work is the observation that in practice many associated events in multiple temporal sequences do not occur concurrently but sequentially. In an empirical study, we apply MAD to tackle two problems originating from different application domains. The experimental results show that our method performed better than other related methods in these domains.

Acknowledgements

Many thanks to my supervisor Dr. Jörg Sander for his consistent help and guidance throughout this research study. Special thanks to my wife and my parents for their support and encouragement. I also would like to thank all the people who gave me help and advice.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the advance of technology and science, more and more organizations have begun to use information systems to assist their business processes and a lot of temporal data in the form of event sequences have been generated. Our work assumes that a set of event sequences has been collected, where each event occurrence has an event id and an occurrence time. Detecting associated events in multiple event sequences has become an issue that attracted more and more attention in the past few years. Our work is motivated by the observation that in practice many associated events in multiple event sequences do not occur concurrently but with a temporal lag. There exist many practical applications that require mining such temporal relations. For example, in network monitoring, where people are interested in the analysis of packet and router logs, different types of events occurring sequentially can be recorded in a log file. The goal is to discover the temporal associations of these events, which indicate the performance of the network. In human-computer interaction modeling, an event sequence represents actions taken by users during a period of time and the goal is to capture aspects such as user intent and interaction strategy by understanding causative chains of connections between actions. A further example is from neuroscience, where analyzing multi-neuron spike data is a challenging problem. With the availability of large amounts of data representing the simultaneous activity of hundreds of neurons, discovering significant patterns of coordinated spiking activity among neurons helps in interpreting the underlying connectivity structure in the neural tissue and relating it to the function of the nervous system.

In this thesis, we design a general, statistical method to detect temporal associations from multiple event sequences. The proposed method can be applied to a large range of application domains and the detected temporal associations can help people in interpreting the behaviors or making predictions for an information system.

1.1 Problem Description

The problem we address in this thesis is to find temporal associations from multiple event sequences. This is not a trivial problem. Assume, for example, we have three sensors, each attached to one of three motors controlling the movements (meaning hoist power, crowd power and swing power) of a shovel dipper. Each sensor continuously records the power consumed by a motor in some time interval, generating a time series of measurements. Such data can be used to analyze the activities, such as the dig-cycles, of a shovel. A typical dig-cycle of a shovel is characterized as one complete cycle of digging the surface, lifting the dirt and loading it on a truck. It is likely in such a data set that events (e.g., defined as subsequences of measurements with certain characteristics) are temporally associated with each other. Ideally, if events in two different sequences are generated by repeating the same coordinated activity, they should occur each time sequentially with a similar temporal lag, whenever the activity occurs. However, things become more complicated in reality: two associated events can occur simultaneously (e.g., lifting, swinging and dumping contents of shovel into a truck); the event, which is supposed to occur later, occurs first (e.g., lifting a shovel full of dirt and then digging the surface again). Furthermore, there may be additional independent occurrences of events on each sequence (e.g., a lifting of the shovel without a following swing, e.g., when re-positioning the shovel). For an associated pair of event occurrences, it is reasonable to assume that the temporal distance of these two occurrences falls within a range around a mean temporal distance, following a distribution, e.g., uniform or Gaussian. However, we may obtain different sets of pairs of event occurrences from two event sequences by assuming that the temporal distance between two associated events follows distinct

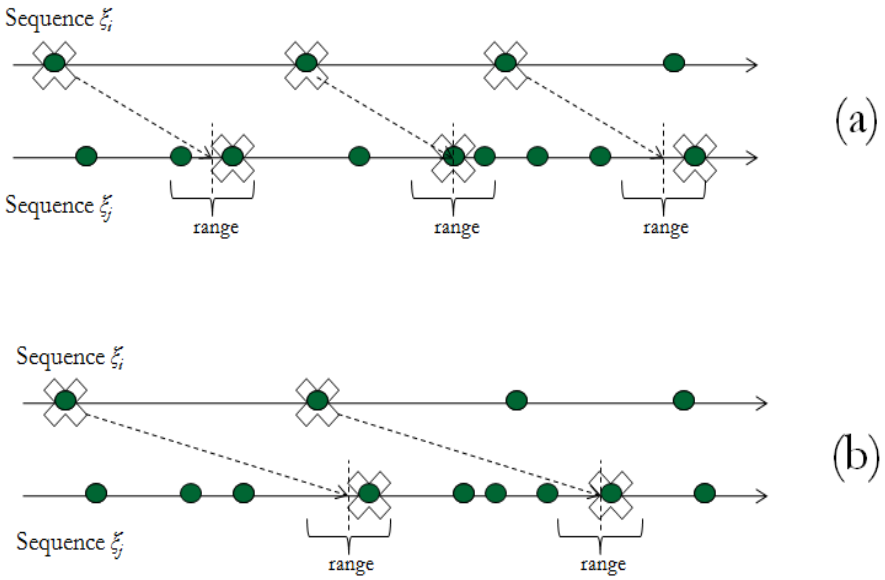

Figure 1.1: (a) Obtaining a set of associated pairs of event occurrences assuming the temporal distance of two associated events follows a uniform distribution with mean = 5 time units and range = 2.5 time units. (b) Obtaining another set of associated pairs assuming the temporal distance follows a uniform distribution with mean = 10 time units and range = 2.5 time units.

distributions. For example, Figure 1.1 shows that we can have two unique sets of associated pairs of event occurrences, given that the temporal distance between two associated events follows different uniform distributions. In the figure, the events marked by crosses denote the associated pairs. A method should be developed to verify that the events in each pair are really temporally associated caused by the same coordinated activity, not just randomly occurring one after the other with a similar time lag. This problem can be solved by employing the methodology of statistical hypothesis testing, where an alternative hypothesis that sample observations are influenced by some non-random mechanism, is compared to a null hypothesis that sample observations result purely from chance.

1.2 Contributions of the Thesis

In this thesis, we propose a two-phase method, called *Multivariate Association Discovery* (MAD). In the first phase, we search for bivariate associations from pairs of

event sequences by comparing the observed distribution of the temporal distances of their event occurrences with a theoretically derived null distribution. A bivariate association will be reported if there exists in the observed distribution a region that has a statistically significant higher count of temporal distances than expected. Two approaches have been designed in this thesis to detect a statistically significant region. In the first approach, we estimate the observed distribution by using the histogram of forward distances and applying a state-of-the-art binning technique to learn a proper bin size for this histogram. A statistically significant region will be retrieved if there exists a bin whose frequency is statistically significant assuming the expected null distribution. In the second approach, given that the temporal distance between two associated events follows a Gaussian distribution, we estimate the observed distribution by using an effective kernel density estimation technique. A theoretical distribution function is derived from the analysis of individual distributions of three categories of temporal distances generated by events of distinct sequences. We treat bivariate association discovery as a least squares curve-fitting problem, where we adjust the parameters of the Gaussian components in the theoretical function to optimally fit the curve of the observed distribution. A statistically significant region will be identified if there exists a bell-shaped portion in the observed distribution showing a statistically significant deviation from the expected null distribution. In the second phase, based on a bivariate association graph, we search each path in the graph for a multivariate association with the requirement that its frequency must be statistically significant in the data set.

To evaluate the usability of our method, we applied it to two application domains. Firstly, we applied MAD to detecting multivariate motifs from multivariate time series data. Existing methods of multivariate motif discovery are all limited by assuming explicitly or implicitly that the univariate elements of a multivariate motif occur completely or approximately synchronously. This assumption does not hold in many real-world applications. We compared MAD with the currently most effective related work on both synthetic and real-world data sets. The experimental results indicate that our method can not only discover synchronous motifs as the other method does, but also successfully find non-synchronous multivariate motifs.

Secondly, we applied our method to detect frequent episodes from event streams. An episode can be understood as a temporally ordered set of event types. Current methods on frequent episode discovery are all limited by requiring users to either provide possible lengths of frequent episodes or specify an inter-event time constraint for every pair of successive event types in an episode. We compared MAD with the most recent work on frequent episode discovery by using simulation data generated by a mathematical model of spiking neurons. The empirical results show that our method is very effective in detecting episodes with variable lengths automatically.

1.3 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 outlines current work about multivariate motif discovery and provides an overview of methods on frequent episode discovery. Chapter 3 gives basic definitions. Chapter 4 presents an overview of our method. Chapter 5 describes the first stage of our method - detecting bivariate associations from two event sequences. Chapter 6 depicts the second stage of our method - discovering multivariate associations based on a bivariate association graph. Chapter 7 gives an empirical study on our bivariate association mining approaches. Chapter 8 presents an experimental evaluation of the proposed method for multivariate motif discovery. Chapter 9 describes the experimental settings and results of our method on frequent episode discovery. Finally, we summarize our work and outline future research directions in Chapter 10.

Chapter 2

Related Work

2.1 Multivariate Motif Discovery

In this research work, since we aim at mining temporal associations between events, which are represented by their occurrence times, of different sequences, our method is related to the work in multivariate motif discovery. In a univariate time series, a *motif* is a set of time series subsequences that exhibit high similarity and occur frequently (according to some measure, e.g., frequency above a threshold) in the whole time series [10]. Typically, the occurrence of a motif corresponds to some meaningful aspect of the data, such as a characteristic action in on-body sensor data. Figure 2.1 shows a motif retrieved from a time series of some industrial process measurements. Considering each motif occurrence as an event, the set of motif occurrences, which are retrieved from the same univariate time series, can be transformed into an event sequence.

In recent years, multivariate time series data are collected widely in many scientific fields, ranging from meteorology to health science. For example, in an environmental monitoring system, the data are often gathered over time at different locations, leading to a geographically indexed multivariate time series. In a d-dimensional multivariate time series containing d univariate time series with corresponding time points, a n-dimensional *multivariate motif* ($n \leq d$) is a set of n-dimensional tuples of univariate elements, where the univariate elements from different dimensions have a temporal association, i.e., they occur concurrently as a *synchronous* multivariate motif (e.g., motif 1 in Figure 2.2) or sequentially as a

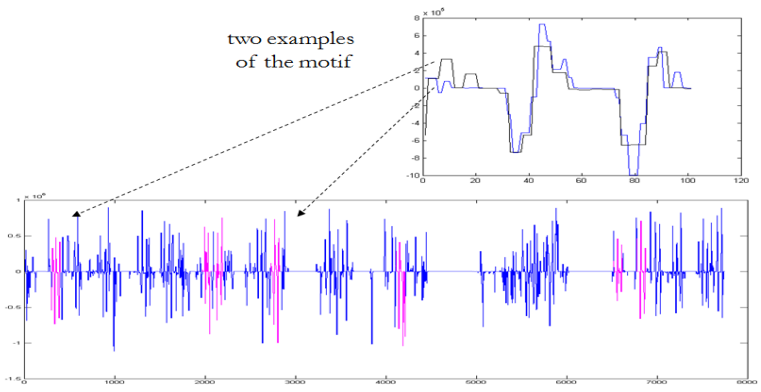

Figure 2.1: A segment of a complex and noisy industrial data set where a motif occurs seven times. The subsequences with pink color indicate the positions of these motif occurrences. A zoomed-in view reveals how similar two examples of these occurrences are to each other.

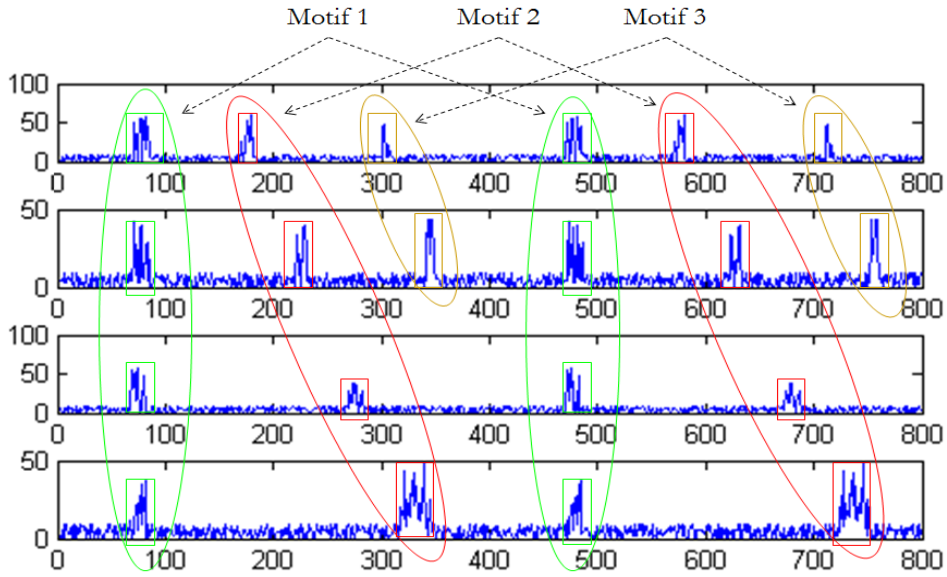

Figure 2.2: Illustration of three multivariate motifs. An ellipse represents a multivariate motif occurrence, and a rectangle denotes a univariate element.

non-synchronous multivariate motif (e.g., motif 2 and motif 3 in Figure 2.2). A multivariate time series can be transformed into multiple event sequences by transforming motifs of univariate time series. A multivariate motif is a special case of a multivariate association.

Based on their approaches of handling multivariate time series data, existing methods of multivariate motif discovery can be classified into three categories.

(1) Representing a multivariate time series as a set of multi-dimensional points. The mining algorithms in this group treat each univariate time series as a dimension and retrieve a set of d-dimensional points from d equal-length univariate time series. Minnen *et al.* proposed a method to detect multivariate motifs that are sparsely distributed in activity data [23]. This method represents the data points symbolically based on a vector quantization, and the result strings are processed by a suffix tree to locate motif seeds (two strings that are most similar to each other). Based on these motif seeds, *hidden markov models* (HMMs) are used to retrieve other motif occurrences. In their later work, another method, which treats the motif discovery problem as locating regions of high density in the space of multivariate time series subsequences, was designed [22]. Dense regions are found by using k-nearest neighbor search (combined with a dual-tree algorithm to reduce the computational complexity). Subsequences representing local density maxima in the space, are determined as motif seeds. An HMM is then learned from each motif seed and its k nearest neighbors. Based on these motif seeds and their trained HMMs, greedy mixture learning is used to discover other motif occurrences. Wang *et al.* proposed another method, which first scans the entire multivariate data to construct a list of candidate motifs using a modified suffix tree that can handle raw data directly, then the list is used to populate a sparse self-similarity matrix for further processing to generate the final selections [36]. Although the methods described in this category handle the original data directly and work well in several example data sets, the high computational complexity and parameter setting make these methods not applicable in many situations. Furthermore, the resulting multivariate motifs must span all of the dimensions and the univariate elements in a multivariate motif must be equally sized.

(2) Transforming a multivariate time series into a univariate time series. Tanaka *et al.* presented a method of detecting synchronous multivariate motifs [33]. In this method, the authors first used *principal component analysis* (PCA) to transform a multivariate time series into a univariate time series, and a set of equal-length univariate time series subsequences were formed by sliding a fixed-length window over the projected time series. Each subsequence is further symbolized by using

symbolic aggregate approximation (SAX) [16] - a local quantization method that divides the subsequence into several equal-sized segments, computes the average for each segment and replaces the segment with a symbol. The SAX algorithm assigns a symbol to each segment by consulting a table of pre-computed breakpoints that divide the data range into equiprobable regions assuming an underlying Gaussian distribution. Since every SAX symbol subsequence represents a part of the behavior of the projected time series, the authors further replaced the subsequence by a single unique symbol, called "*behavior symbol*". For example, let '*bcba*' denote a SAX symbol subsequence, where '*c*' means the range of high values, '*b*' means the range of middle values and '*a*' represents the range of low values. This subsequence can be assigned to letter '*B*' that conceptually means "*the time series starts in the middle range, reaches one high peak and then decreases*". The resulting sequence of behavior symbols is called "*behavior symbol sequence*". Finally, a *minimum description length* (MDL) principle was applied to extract motifs from a set of equal-length behavior symbol subsequences, which were obtained by sliding a fixed-length window over the behavior symbol sequence. In order to detect motifs with different lengths, the authors iteratively increased the length of a sliding window and generated different sets of behavior symbol subsequences. This method shows high efficiency compared to other related work for several data sets and successfully finds multivariate motifs that are intuitive. This method, however, explicitly assumes that all of the univariate elements in a multivariate motif occur completely synchronously. Furthermore, since it is possible that some important information gets lost in the process of dimensionality reduction, the resulting multivariate motifs may not be meaningful in some of the original dimensions. Other related work in this category extend Chiu's algorithm to handle multivariate time series data. Chiu's algorithm was proposed as one of the most efficient methods to detect motifs from univariate time series [10]. In this algorithm, a set of equal-length subsequences are formed by sliding a fixed-sized window over the time series. SAX is applied on each subsequence to reduce its dimensionality and the generated SAX symbol sequence is called "*string*". Motif seeds are identified by using a random projections algorithm [8], which builds a collision matrix via a number

of iterations of random projections. Each iteration involves selecting two strings randomly, choosing a subset of string positions at random and building a hash table with the corresponding SAX symbols in the strings. After all strings are hashed, collisions (i.e., equivalent projections from different strings) are taken as evidence of similarity and the corresponding places in the collision matrix are incremented. Minnen *et al.* developed a method that automatically determines the neighborhood radius for each multivariate motif [24]. The method applies SAX on each of the univariate time series independently and concatenates strings from each dimension occurring together within a sliding window into longer single strings. Each single string actually corresponds to a multivariate time series subsequence. A random projections algorithm is applied on these single strings to search for multivariate motif seeds. Once a pair of multivariate motif seeds is found, for every single string the method computes the Euclidean distance from this string to the closer of the two seeds using the original, real-valued data, and assigns a score to the string using the distance value. After that, all single strings are sorted incrementally in terms of their scores. Estimating the neighborhood radius of a multivariate motif is equivalent to searching for an inflection point in the distribution of these scores. The resulting multivariate motifs still have to span all of the dimensions and the univariate elements of a multivariate motif still have to be equally sized. The most recent work from the same authors deals with the problem of sub-dimensional multivariate motif discovery [21], i.e., detection of multivariate motifs that do not necessarily span all dimensions. Similar to their previous work [24], the authors symbolize a multivariate time series by applying SAX on each of its univariate time series and concatenating strings from each dimension in the same sliding window. Multivariate motif seeds can be identified by using a random projections algorithm. Given a pair of multivariate motif seeds, two different methods were proposed to solve the problem of dimension relevance for this multivariate motif. In the first method, given that multivariate motifs are defined by a fixed, user-specified neighborhood radius, the method determines the dimensions of relevance for a multivariate motif by simply identifying those dimensions that do not cause the Euclidean distance between the seeds of this multivariate motif to exceed the given radius. Specially, this

method sorts the dimensions by increasing distance and then incrementally adds dimensions until the seed distance grows too large. The second method is designed for the case when the neighborhood radius have to be estimated automatically. In this method, the authors first estimate the distribution over distances between random subsequences for each dimension by sampling from the data set. Then, given the distribution and a pair of multivariate motif seeds to analyze, the authors evaluate the probability that a value smaller than the seed distance will arise randomly by calculating the corresponding value of the cumulative distribution function. If this value is large, the dimension is believed to be irrelevant because it is likely to arise at random, otherwise, it likely indicates a relevant dimension. However, the univariate elements in different dimensions of a multivariate motif still must be completely synchronous.

(3) Combining a set of univariate motifs into a multivariate motif. The algorithms of this category apply a univariate motif discovery method on each univariate time series and combine some of the discovered univariate motifs into a multivariate motif by detecting their temporal associations. Vahdatpour *et al.* constructs a coincidence graph based on the temporal relations of discovered univariate motifs [34]. A graph is initially built, where a vertex represents a univariate motif and the weight of an edge between two vertices indicates the frequency that the occurrences of the two univariate motifs, which are denoted by these vertices, temporally overlap. Starting from the motif with the highest occurrences, a graph clustering algorithm iteratively detects "*normal activities*" as multivariate motifs by comparing the weights of edges connected to this motif in the graph to a user-defined threshold. Two univariate motifs are believed to be involved in an activity together if the weight of their connecting edge is greater than the threshold. After all occurrences of an activity have been identified, the graph is updated by eliminating the univariate motif occurrences that are associated to this activity. Another method was developed by the same authors to detect "*abnormal activity occurrences*", as multivariate subsequences with at least one univariate element missing compared to their normal activity occurrences [35]. The authors first used Chiu's algorithm to extract univariate motifs from each dimension independently and applied their

previous work on these univariate motifs to detect normal activities. Given a normal activity, a new algorithm was proposed to find the abnormal occurrences of this activity. The input to the algorithm is the list of all univariate motifs discovered from the data and one normal activity. The algorithm first removes the univariate motifs that do not participate in the normal activity from the list. By scanning the remaining univariate motifs in the list, the algorithm then identifies the abnormal occurrences of this activity with the consideration of the fact that a univariate motif occurrence participates in two occurrences of this activity, while only one of them being a legal activity occurrence. Although the methods in this category still assume that the univariate elements of a multivariate motif should temporally overlap with each other, compared with previous methods, they allow the univariate motifs to have different lengths and frequencies, and provide the flexibility that discovered multivariate motifs can span any subsets of dimensions.

Although existing methods on multivariate motif discovery can successfully retrieve synchronous multivariate motifs from some data sets with particular properties (e.g, ECG data), they are all limited by assuming explicitly or implicitly that the univariate elements of a multivariate motif occur completely or approximately synchronously, which results in their poor performances in applications where non-synchronous multivariate motifs exist. In Section 8, we compare MAD with the currently most effective related work on multivariate motif discovery using both synthetic and real-world data.

2.2 Frequent Episode Discovery

Our method is also related to the work in frequent episode discovery. Frequent episode discovery [17] is a popular framework for detecting temporal patterns in symbolic temporal data, with applications in many domains, such as manufacturing [15], telecommunication [18], biology [7], finance [26] *etc*. In this framework, the input data is typically a sequence of event occurrences with each event occurrence characterized by an event type and an occurrence time. For example, an event sequence with five occurrences can be represented as follows: $<$

$(A, 1.6)$,$(E, 4.9)$,$(B, 5.1)$,$(D, 6.6)$, $(C, 10.5) >$. The temporal patterns detected from the data, referred to as *episodes*, are essentially small, temporally ordered sets of event types. For example, $(A \rightarrow B \rightarrow C)$ stands for an episode, where event type A is followed (some time later) by a B and a C, in that order. When event occurrences of appropriate types appear in the sequence, in the same order as in an episode, these event occurrences are said to constitute an occurrence of the episode. For instance, in the example event sequence, $(A \rightarrow B \rightarrow C)$ occurs once. Depending on different types of temporal orders over their event types, episodes can be classified into two categories: serial episodes and parallel episodes. A *serial episode* requires its event types to occur in a sequential order, e.g., $(A \rightarrow B \rightarrow C)$. In contrast, a *parallel episode* is similar to an unordered set of event types and does not require any specific ordering of the event types, e.g., (DE), where event type D can happen before or after a E. An episode is considered interesting if it occurs more often than a threshold in the data.

Based on their different learning goals, current methods of frequent episode discovery can be classified into two categories.

(1) Mining serial and parallel episodes using an Apriori-style procedure. In this category, the methods use an Apriori-style procedure to detect serial and/or parallel episodes. The discovery process consists of two stages: candidate generation and counting frequencies of candidate episodes. In candidate generation, two episodes of size n can be merged to generate a candidate episode of size $(n+1)$ if they share $(n-1)$ event types and the temporal orders among these $(n-1)$ event types in these two episodes are identical. The frequency of an episode is computed as the number of fixed-sized sliding windows in which the episode occurs. Mannila *et al.* first introduced the framework of frequent episode discovery [17]. Two obvious drawbacks of the proposed framework are that: a) the window size has to be fixed by the user and it remains unchanged throughout the whole mining process, which limits the lengths of discovered episodes; b) an occurrence of an episode may be contained in several successive windows if the window size is larger than the length of the episode, which will fraudulently increase the frequency of this episode. Furthermore, it is hard to set an appropriate window size in some applications and

different episodes may vary in length. To solve these issues, Casas-Garriga introduced a method that requires each consecutive pair of event types (e.g., given an episode $A \rightarrow B \rightarrow C$, the pair $A \rightarrow B$ and the pair $B \rightarrow C$ are consecutive) in an episode have a user-defined maximum gap max_gap of time delay [9]. Every episode that is candidate to be frequent, will be searched in sliding windows whose sizes are dynamically adjusted according to the number of event types in the episode, e.g., a candidate episode with m event types is searched in all windows of size $(m-1) \times max\_gap$ time units. For each episode, the authors still counted the number of sliding windows containing the episode as its frequency. Laxman *et al.* proposed another work to discover serial and parallel episodes by using a new frequency measurement, which counts the number of non-overlapped occurrences for an episode [14]. Two occurrences of an episode are said to be non-overlapped if no event in one occurrence appears in between events in the other occurrence. Based on the new frequency definition, the proposed algorithms are better in terms of both time and space complexities compared with previous methods. The same authors presented two more algorithms in their extension work to detect serial and parallel episodes, with consideration of time durations of event types [13]. The authors defined their so-called "*generalized episode*", which associates each event type in the episode with a set of time intervals. The so-called "*principal episode*" was also introduced in this work. An episode is said to be *principal* if every time interval of an event type in the episode has a positive contribution to the episode's frequency. The new algorithms adopted an Apriori-style mining process, except that detected frequent episodes need to be verified as principal episodes. A unified view of all the Apriori-based discovery methods for serial episodes under different definitions of frequencies were proposed in [4]. This unified view allows one to gain insights into different frequencies by exploring their quantitative relationships. Note that all of the methods in this category detect frequent episodes based on a user-defined frequency threshold, which is hard for a user to determine without much guidance on how to do it.

(2) Mining statistically significant episodes. Achar *et al.* proposed a method to discover "*injective episodes*", in each of which the event types are unique and

the temporal order over these event types can be serial or parallel [3]. The method also adopted an Apriori-style procedure to search for frequent episodes. The proposed method computes the frequency of an injective episode by counting its non-overlapped occurrences. Other than the user-defined frequency threshold, an additional measurement, called "*bidirectional evidence*", was proposed to select interesting injective episodes, based on the principle that any pair of event types in an injective episode, which are not constrained by the episode's temporal orders, should appear in either order sufficiently often. An injective episode is believed to be *interesting* if its frequency and its result on the new measurement are respectively above the user-defined thresholds. Sastry *et al.* described a method to mine statistically significant serial episodes by using a set of user-defined inter-event time delays [29]. After using an Apriori-style mining scheme to detect frequent episodes, the authors designed a statistical test to determine the significance level of these detected episodes, based on the intuition that the interaction between two event sequences can be captured by the conditional probability of observing an event from one sequence after a time delay given that an event has occurred on another sequence. The major drawback of this work is that the proposed method needs us to specify an inter-event time delay for any of two event sequences in order to conduct the statistical test. Without understanding the mechanism of data generation, these parameters are not intuitive for people to determine. Patnaik *et al.* presented a different approach to find temporal associations between events by learning an optimal *dynamic bayesian network* (DBN) structure from event sequences [27]. In their work, a specialized class of DBNs, called "*excitatory network*", was proposed. In an excitatory network, nodes denote event types and edges represent *excitatory* influences among nodes, i.e., a set of nodes in the network exert excitatory influences on node A, if occurrence of events corresponding to the nodes in the set increases the probability of occurrence of A. The authors also defined their so-called "*fixed-delay episode*", where the time delays between event types are fixed. For example, if $(A \xrightarrow{5} B \xrightarrow{10} C)$ denotes a fixed-delay episode, every of its occurrences must comprise an A, followed by a B exactly after 5 time units later, which in turn is followed by a C exactly 10 time units later. To obtain the marginal proba-

bilities in an excitatory network, the frequencies of fixed-delay episodes, which are computed by counting their non-overlapped occurrences, are used to compute the joint probabilities and the inclusion-exclusion formula is used to compute the conditional probabilities for each node given different assignments to its parent nodes. Although this work presented a novel approach to mine temporal associations by using DBNs, due to the strong limitation on the time delays between event types in a fixed-delay episode, the proposed method can only be applied on some particular applications. Gwadera *et al.* proposed another method to find statistically significant serial episodes [12]. In this work, an episode is regarded as a subsequence of the input event stream within a window of a given fixed size. A set of candidate frequent episodes is first discovered by using an Apriori-style procedure. To determine the significance of a candidate, a reference model is created either by using a memoryless Bernoulli model or a markov model. The authors used $\Omega^{\exists}(n, w, m)$, which represents the number of windows of length w containing at least one occurrence of episode S of length m when sliding the window along n consecutive events of the input event stream. The authors proved that the normalized $\Omega^{\exists}(n, w, m)$ approximately follows a Gaussian distribution. Given the reference model and a significance level, the proposed method computes an expected frequency for each candidate frequent episode. If the observed frequency of the candidate is statistically significantly larger than its expected frequency, it indicates that this candidate is highly unlikely generated by the reference model and can be regarded as a statistically significant episode. Note that our approach is different from the work in this category, as we use statistical tests to guide the discovery process and the resulting temporal patterns must be statistically significant, but all of the methods in this category just use their designed statistical tests to verify the significance level of temporal patterns.

Although current methods on frequent episode discovery perform well in some data sets, they are all limited by requiring users to either provide possible lengths of frequent episodes or specify an inter-event time constraint for every pair of successive event types in an episode, which results in poor performance in applications where people have little knowledge about the data. Furthermore, the result

of these methods is not well founded in statistics, as most of them evaluate the "significance" of discovered episodes by simply comparing their frequencies to a user-defined frequency threshold. In Section 9, we compare our method with the most recent method on frequent episode discovery using simulated spike train data.

Chapter 3

Background and Definitions

3.1 Event Sequence

An *event sequence* $\xi =< e_1, e_2, \ldots, e_m >$ is an ordered set of events. Each e_i in ξ denotes a tuple (e_id, t_i), where e_id represents the event id and t_i is the occurrence time of the event. All event occurrences in ξ are assumed to be of the same type and the events are ordered by their occurrence time. For example, the event sequence in Figure 3.1 can be represented as follows: $\xi =< (1, 2.3), (2, 5.0), (3, 10.4), (4, 19.8), (5, 24.9) >$.

3.2 Bivariate Associations

We introduce a *bivariate association* A^d_{ab} $(a \neq b)$, between two event sequences ξ_a and ξ_b, as a subset of the Cartesian product of ξ_a and ξ_b, as following:

Definition 1 *Let ξ_a and ξ_b be two event sequences. A set $A^d_{ab} \subseteq \xi_a \times \xi_b$ is called a bivariate association in (ξ_a, ξ_b) with mean temporal distance d if for all $(e, e') \in A^d_{ab} : t \leq t' \wedge t' - t \sim \Phi(\cdot) \wedge E(t' - t) = d$, and there is a one-to-one correspondence between the sets $\{e | \exists e' : (e, e') \in (A^d_{ab})\}$ and $\{e' | \exists e : (e, e') \in (A^d_{ab})\}$, where t (resp. t') is the occurrence time of event e (resp. e'), $\Phi(\cdot)$ denotes a a known*

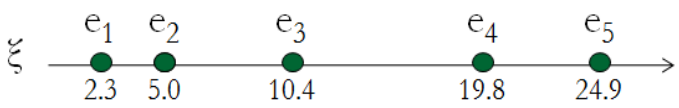

Figure 3.1: Event Sequence

distribution (e.g., uniform or Gaussian) that the temporal distance between two associated events follows, and $E(t' - t) = d$ is the expected temporal difference between associated events in A^d_{ab}.

3.3 Multivariate Associations

A *multivariate association* $MA^{d_1 \ldots d_{k-1}}_{1 \ldots k}$ between k event sequences $\xi_1, \ldots, \xi_k$ is defined as:

Definition 2 *Let $\xi_1, \ldots, \xi_k$ be k different event sequences. A set $MA^{d_1 \ldots d_{k-1}}_{1 \ldots k} \subseteq \xi_1 \times \ldots \times \xi_k$ is called a multivariate association in $(\xi_1, \ldots, \xi_k)$ if for all $(e^1, \ldots, e^k) \in MA^{d_1 \ldots d_{k-1}}_{1 \ldots k} : (e^i, e^{i+1})$ is an instance of a bivariate association in (ξ_i, ξ_{i+1}) with mean temporal distance d_i for all $1 \leq i \leq k - 1$.*

Because of the one-to-one correspondence of associated events in a bivariate association, a multivariate association has this property between any pair of its associated events.

3.4 The Poisson Process and its Properties

In many real-world applications, especially in applications where the event sequences are the result of the superimposition of many low intensity arbitrary (i.e., non-Poisson) point processes [20], the collected event sequences can be modeled as Poisson processes. One example is from neuronal spike train analysis, where a spike train can be viewed as a stochastic point process and is usually assumed to be a Poisson process [11]. For our approach, we assume that the event sequences collected from an application can be modeled as Poisson processes. We define a Poisson process by adopting the approach from [5]:

Definition 3 *A Poisson process, with uniform intensity $\lambda > 0$, is a point process such that:*

- *for every bounded interval $(j, k]$, the count $N_{(j,k]}$ has a Poisson distribution with mean $\theta = \lambda(k - j)$;*

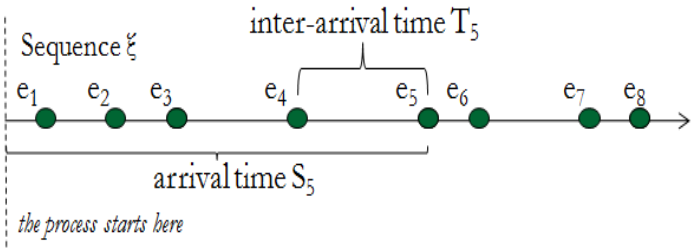

Figure 3.2: Two Properties of a Poisson Process

- *if* $(j_1, k_1], \ldots, (j_m, k_m]$ *are disjoint bounded intervals, the counts* $N_{(j_1,k_1]}, \ldots, N_{(j_m,k_m]}$ *are independent random variables.*

Properties of a Poisson process include:

- the **inter-arrival times** T_i between consecutive event occurrences are independent and follow an exponential distribution with rate $\mu = 1/\lambda$:

$$P(T_i) = \lambda e^{-\lambda T_i}, \tag{3.1}$$

$$F(T_i \leq x) = 1 - e^{-\lambda x}, x > 0; \tag{3.2}$$

- the ith **arrival times** S_i, i.e., the time until the ith event occurrence from the starting point of the process, have a Gamma distribution with shape parameter $\alpha = i$ and scale parameter $\beta = \lambda$:

$$P(S_i) = \lambda e^{-\lambda S_i} \frac{(\lambda S_i)^{i-1}}{(i-1)!}, \tag{3.3}$$

$$F(S_i \leq x) = 1 - \sum_{j=0}^{i-1} e^{-\lambda x} \frac{(\lambda x)^j}{j!}. \tag{3.4}$$

$P(\cdot)$ denotes probability density functions and $F(\cdot)$ denotes cumulative distribution functions. Figure 3.2 illustrates these notions, where e_5 denotes the fifth event occurrence of sequence ξ and S_5 represents its arrival time. Based on the two properties, S_5 follows a Gamma distribution with shape parameter $\alpha = 5$ and scale parameter $\beta = \lambda$. T_5 is the inter-arrival time between the events e_4 and e_5, following an exponential distribution with mean $\mu = 1/\lambda$.

Chapter 4

Our Methodology

We propose a two-phase method to detect multivariate associations from multiple event sequences. In the first phase, we detect bivariate associations from two event sequences by comparing the observed distribution of the temporal distances of their event occurrences with an expected null distribution theoretically derived from the properties of a Poisson process assuming that the event occurrences are randomly and independently positioned. A bivariate association is reported if there exists in the observed distribution a region that has a statistically significant higher count of temporal distances than expected. Two approaches are proposed in this thesis to search for a statistically significant region. In the first approach, we estimate the observed distribution by using the histogram of forward distances and utilizing the state-of-the-art binning technique to learn a proper bin size for this histogram. A statistically significant region is retrieved if there exists a bin in the histogram whose frequency is statistically significant assuming the expected null distribution. In the second approach, given that the temporal distance between two associated events follows a Gaussian distribution, we estimate the observed distribution by using an effective kernel density estimation technique. A theoretical distribution function is derived from the analysis of individual distributions of three categories of temporal distances generated by events of distinct sequences. We treat bivariate association discovery as a least squares curve-fitting problem, where we adjust the parameters of the Gaussian components in the theoretical function to optimally fit the curve of the observed distribution. A statistically significant region is identified if there exists a bell-shaped portion in the observed distribution showing a statisti-

cally significant deviation from the expected null distribution. In the second phase, we use a bivariate association graph to search for multivariate associations with the requirement that their frequencies should also be significant in a statistical sense.

Chapter 5

Detecting Bivariate Associations

5.1 Forward Distances

To determine whether two event sequences ξ_a and ξ_b are temporally associated, we analyze what we define as *forward distances*. A forward distance can be understood as the difference in time between an event $e \in \xi_a$ and one of the events $e' \in \xi_b$ occurring after e. The set of forward distances between events of sequence ξ_a and sequence ξ_b is defined as following:

Definition 4 *Let ξ_a and ξ_b be two event sequences. $FD_{ij} = \{dist | \exists e \in \xi_a \exists e' \in \xi_b, t \leq t' \wedge dist = t' - t\}$, where t (resp. t') denotes the occurrence time of e (resp. e').*

5.2 The Expected Null Distribution of Forward Distances

We can describe the forward distances between two event sequences ξ_a and ξ_b in the following way: to compute the forward distances for an event on sequence ξ_a, we can think of projecting the event onto sequence ξ_b and denoting the projected position as h. The forward distance from h to its right nearest event on sequence ξ_b can be denoted as Z_1. Since we compute Z_1 for each event of sequence ξ_a, Z_1 can be treated as a random variable and we can derive the distribution of this random variable from the properties of a Poisson process. The distribution is characterized

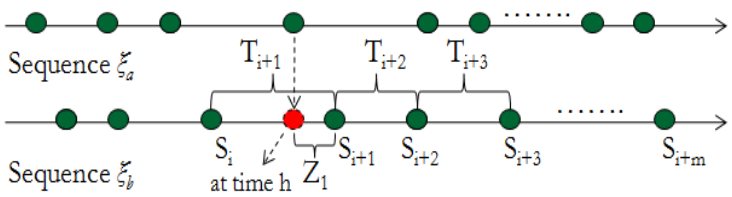

Figure 5.1: Waiting Time Paradox

by the so-called *Waiting Time Paradox* for Poisson processes.[1]

Theorem 1 *Waiting Time Paradox: Suppose that h is a time stamp randomly selected from an event sequence following a Poisson process with intensity λ. Let the arrival time of the event right after h be denoted as S_{i+1}. Let $Z_1 = S_{i+1} - h$, and each inter-arrival time after h be denoted as $T_{i+j}, j = 2, 3, \ldots, m$. Then the random variables $Z_1, T_{i+2}, \ldots, T_{i+m}$ are independent and identically distributed, with an exponential distribution with rate $\mu = 1/\lambda$.*

Figure 5.1 illustrates the theorem. In the figure, we select an event at random from sequence ξ_a with intensity λ_a and project it onto sequence ξ_b with intensity λ_b. T_{i+1} represents the inter-arrival time that contains the projected event (i.e., the dashed circle). m denotes the number of events that occur after time h. S_{i+j} denotes the arrival time of the jth event after time h. The theorem tells us that Z_1 follows the same exponential distribution as T_i, the inter-arrival time on sequence ξ_b, with mean $\mu = 1/\lambda_b$. If we use Z_j to denote the forward distance from time h to the jth event after h, based on the properties of a Poisson process, Z_j is exactly the arrival time of the jth event in the Poisson process starting at time h and follows a Gamma distribution with shape parameter $\alpha = j$ and scale parameter $\beta = \lambda_b$.

Knowing the distribution of the forward distances to the first right neighbor, to the second right neighbor, and so on, we can express the distribution of all forward distances from all events on a finite sequence ξ_a to all events on a finite sequence ξ_b as a mixture (weighted sum) of these individual distributions:

$$f_n(x) = \sum_{j=1}^{N} W_j \times g(x, j, \lambda_b), \tag{5.1}$$

[1] Meester gives a mathematical proof for this theorem [20]

where $f_n(x)$ is the expected null distribution of forward distances, i.e., the distribution under the assumption that there is no temporal association in the data. N is the number of individual distribution components, which equals the number of forward distances the first event on sequence ξ_a has. The term $g(x, j, \lambda_b)$ is the Gamma distribution that Z_j follows. W_j represents the weight of the jth Gamma distribution component. Note that these Gamma distribution components have different frequencies. Because of its temporal position, an event e on sequence ξ_a may not have a forward distance to its jth right neighbor if there are fewer than j events on ξ_b to the right of the projected position of e. The weight for each component density can be estimated from the properties of the involved Poisson processes, as described next. If we use $F_{g(j,\lambda_b)}(x)$ to denote the cumulative distribution function of Z_j, the cumulative distribution function of $f_n(x)$ can be written as:

$$F_n(X \leq x) = \sum_{j=1}^{N} W_j \times F_{g(j,\lambda_b)}(x). \tag{5.2}$$

Figure 5.2 illustrates how the expected values of the weights $W_j (1 \leq j \leq N)$ can be determined. In the figure, S_n is the arrival time of the last event on sequence ξ_b and there are k events on sequence ξ_a that occur before S_n. S_{n-j+1} denotes the arrival time of the jth last event on sequence ξ_b. Every event on ξ_a that occurs before S_{n-j+1} will have all forward distances to events on ξ_b up to and including their jth right neighbor, i.e., they contribute a distance to Z_j. However, every event on ξ_a after S_{n-j+1} will not have a distance to their jth right neighbor and will not contribute a distance to Z_j. If T_j denotes the time interval between S_{n-j+1} and S_n, its expected length $E(T_j)$ is $(j-1)/\lambda_b$. The expected number of events on sequence ξ_b that are in time interval $E(T_j)$, i.e., the expected number of events that do not contribute to distances in Z_j, can be estimated by $(j-1)\lambda_a/\lambda_b$. Let N_j represent the number of distances in Z_j; its expected number $E(N_j)$ can be estimated as $k-[(j-1)\lambda_a/\lambda_b]$. Hence, we can estimate each weight W_j by $E(N_j)/\sum_{i=1}^{N} E(N_i)$.

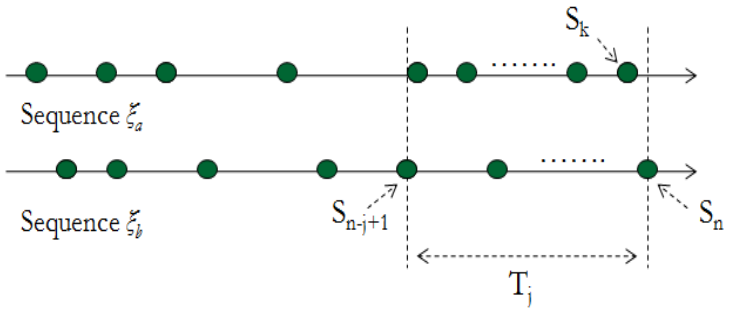

Figure 5.2: Determining the Weights of Individual Gamma Distribution Components

5.3 Limiting the Search Scope

To make our approach more efficient, we can consider only a limited number m of events on sequence ξ_b when computing forward distances for an event on ξ_a. While we may not know in many applications how many forward distances we should consider for an event in order to capture the temporal association of two event sequences, it is often possible to indicate a range of time (e.g., 10 minutes) after which we do not expect an associated event to occur on sequence ξ_b. To exploit this information and save computations, we suggest a practical method to estimate m: for each event on sequence ξ_a, we use the given maximum time to determine the number of forward distances that occur within this time after the event and set m as the maximum observed number of such forward distances.

5.4 A General Approach to Discover Statistically Significant Regions in the Observed Distribution of Forward Distances

In this section, we present a general approach to detect statistically significant regions from the distribution of forward distances generated by events of different sequences. In the proposed approach, we estimate the observed distribution of forward distances by creating a histogram of these distances and apply the state-of-the-art binning technique to learning a proper histogram bin size. A statistically significant region will be reported if there exists a bin in the histogram whose fre-

quency is statistically significant assuming the expected null distribution.

5.4.1 Using Histograms to Estimate the Observed Distribution

In order to detect whether events from two event sequences are temporally associated, we estimate the actual, observed distribution of forward distances. A simple approach is to compute the forward distance for each pair of event occurrences from different event sequences and generate a histogram of these distances. Determining a proper histogram bin size, however, is not trivial. A simple solution is to use one of the existing heuristic binning techniques to determine the bin size automatically. However, some binning techniques (e.g., Sturge's rule [32]) tend to generate a large bin size, resulting in a large variance of forward distances in each bin, which makes it difficult to detect bins with an unusually high count (compared to the expected null distribution). In our research work, we use Shimazaki's method [31], which selects the bin size that minimizes an estimated L^2 risk function. Suppose a set of forward distances are divided into several bins of width $\triangle$. The number of distances in the ith bin is denoted as k_i. The risk function is defined as:

$$C(\triangle) = \frac{2\bar{k} - v}{(\triangle)^2}, \tag{5.3}$$

where $\bar{k}$ and v represent respectively the mean and variance of k_i. The only parameter that Shimazaki's method introduces is the number of iterations for its method. The larger this parameter becomes, the more candidate bin sizes this method can try for learning an optimal bin size for a histogram of forward distances.

It is reasonable to assume that the variance of the distances that a bivariate association follows is not extremely large (otherwise it would be hard to argue even for the existence of such an association). That means that an event e on sequence ξ_a is typically followed by an associated event e' on sequence ξ_b within a relatively small amount of time around the expected mean distance in this association (the distance itself could be large).

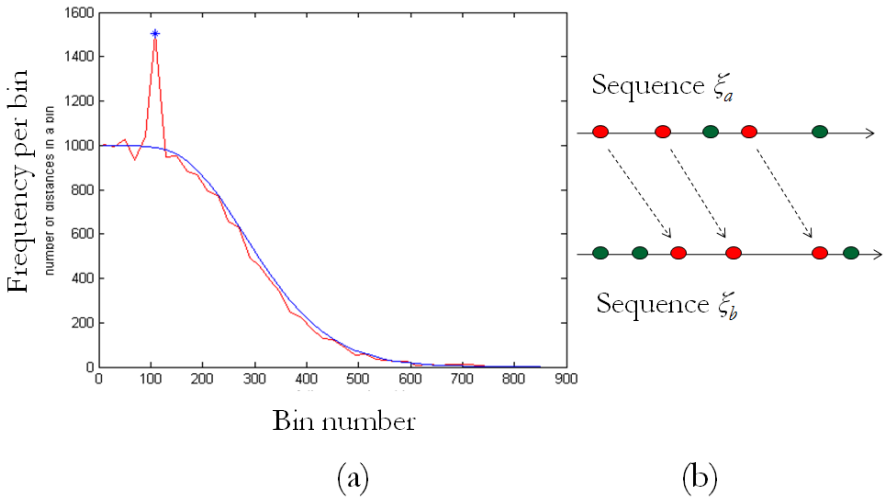

Figure 5.3: (a) Observed distribution (red curve) versus theoretical distribution (blue curve) for a bivariate association with 500 instances. (b) A snapshot of the event sequences used for plotting the left graph. The events linked by the directed dash lines are the associated events determined by our method.

5.4.2 Detecting Statistically Significant Bins in the Observed Distribution

Our method is based on the intuition that when two event sequences have a temporal association, the number of forward distances in a bin in the observed distribution should be larger than expected under the null distribution. Figure 5.3 (a) demonstrates an example of the expected null distribution (the blue curve) and the observed distribution (the red curve) for a bivariate association with 500 instances (i.e., associated pairs of event occurrences), showing clearly a spike in the bin containing distances close to the mean temporal distance between associated event occurrences. The events linked by the directed dashed lines in Figure 5.3 (b) are the associated events determined by our approach.

Let B be a bin in a histogram of observed forward distances, and let $ON(B)$ denote the number of distances in B (ON stands for *Observed Number*). We use the methodology of statistical hypothesis testing to determine the probability that B contains $ON(B)$ distances under the null hypothesis, i.e., under the assumption that all of the distances are distributed according to the null distribution $f_n(x)$. Suppose n is the total number of observed forward distances and P_B is the probability that a

distance falls into B. Then, the probability that a distance does not fall into this bin is $1 - P_B$. If we regard the event that a randomly chosen temporal distance falls into the particular bin B as a Bernoulli experiment, which is repeated independently n times with a success probability equal to P_B, we can use the Binomial distribution as the null hypothesis to compute the probability that we observe $ON(B)$ in bin B. More precisely, the distribution of the test statistic, $ON(B)$, under the null hypothesis is the Binomial distribution with parameters n and P_B, i.e., $ON(B) \sim Binominal(n, P_B)$. P_B can be derived as:

$$\begin{aligned} P_B &= \int_l^u f_n(x)dx \\ &= F_n(u) - F_n(l) \\ &= \sum_{j=1}^{N} W_j \times (F_{g(j,\lambda)}(u) - F_{g(j,\lambda)}(l)), \end{aligned} \tag{5.4}$$

where l and u stand for the lower respectively upper bound of bin B. P_B equals the cumulative distribution of $f_n(x)$ from the bin's left boundary to its right boundary.

Let α_0 be a significance level, and let α be the probability that we observe $ON(B)$, which is computed from $Binomial(n, P_B)$. B is a statistically significant region at significance level α if $\alpha \leq \alpha_0$. Typical values of α_0 for single statistical tests are 0.05 and 0.01. However, since we conduct a larger number of tests, we perform a Bonferroni adjustment [30] of α_0 to avoid a large number of false positives. We adjust the significance level α_0 as $\alpha_0' = \alpha_0/m$, where m denotes the number of bins being tested.

It is possible that the region where a bivariate association places in the observed distribution can span several bins. In order to search for an appropriate region size for this bivariate association, our method merges adjacent bins if they are all statistically significant and tests the merged bin again. If the merged bin is also statistically significant, our method will report it instead of the individual bins.

5.4.3 Modeling Bivariate Associations

If two event sequences are temporally associated, we can estimate the mean temporal distance of their associated pairs of event occurrences and the range of these

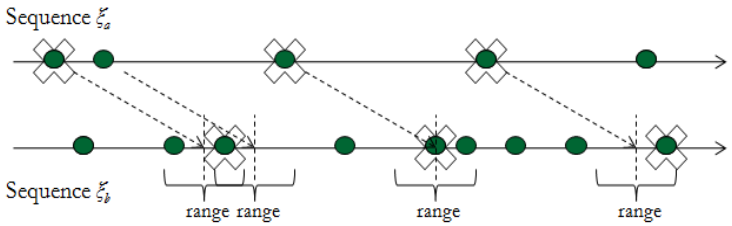

Figure 5.4: Making Associated Pairs of Events

distances by retrieving the bins that show statistically significant deviation from the expected null distribution. We have described in Section 5.4.1 that the temporal distance between two associated events from different event sequences should follow a known distribution with a relatively small range around a mean distance for most meaningful applications. Given a statistically significant bin, the *mean* temporal distance of two associated events in a bivariate association can be estimated as the mean of the distances inside the bin, and the bin size can be used as the *range* of the distances in which we will search for a matching event occurrence.

We retrieve the bivariate association A^d_{ab} from two sequences ξ_a and ξ_b by searching for a set of associated pairs of event occurrences. For an event occurrence on ξ_a, we only consider the event occurrences on ξ_b that are in the time interval $[mean - range/2, mean + range/2]$. There may be more than one event occurrence on ξ_b that can be paired with an event occurrence on ξ_a within that time interval. It is hard to tell in this case which event occurrence is truly associated with the event occurrence on ξ_a. In the current implementation, we adopt a simple approach by selecting the event occurrence on ξ_b that is closest to the mean temporal distance in that time interval. Furthermore, it is possible that two event occurrences on ξ_a can be paired with an event occurrence on ξ_b. In this situation, we will select the event occurrence that happens earlier than the other one on ξ_a to construct the associated pair. Although there are some real-world applications where multiple-to-one correspondence of associated events does exist, in this thesis, we only focus on investigating one-to-one correspondence of associated events. Figure 5.4 demonstrates this procedure. In this figure, the events marked by crosses are the associated pairs determined by our method.

5.5 Another Approach to Find Statistically Significant Regions in the Observed Distribution

Assuming that the temporal distance between two associated events follows a Gaussian distribution, we introduce an approach to look for statistically significant regions in the observed distribution of forward distances. The intuition behind this approach is based on the observation that the mean and the standard deviation of a Gaussian distribution can be estimated by using the zero-crossing points of its second derivative curve. In this approach, we derive a theoretical distribution function from the analysis of individual distributions of three categories of forward distances generated by events from distinct sequences. We treat bivariate association discovery as a least squares curve-fitting problem, where we adjust the means and the standard deviations of the Gaussian components in the theoretical function to optimally fit the curve of the observed distribution. A statistically significant region will be reported if there exists in the observed distribution a bell-shaped portion with a statistically significant higher count of forward distances than expected.

5.5.1 The Theoretical Distribution of Forward Distances

Figure 5.5 illustrates a general scenario of two event sequences where embedded pairs of associated events occur. In this figure, the red points linked by the directed dashed line represent a pair of truly associated events and the green points stand for event occurrences that are randomly and independently distributed on one of the two sequences. There are a total of m event occurrences on sequence ξ_a and n event occurrences on sequence ξ_b. To compute the forward distances for an event occurrence on sequence ξ_a, we project it onto sequence ξ_b and the temporal difference between the projected position h and any of the event occurrences happening after h on sequence ξ_b is the forward distance of the two occurrences.

If there exist some associated pairs of events between two event sequences, only three categories of forward distances can be generated by events from different sequences. The first category of forward distances consists of the distances from an event occurrence that is randomly and independently distributed on sequence ξ_a,

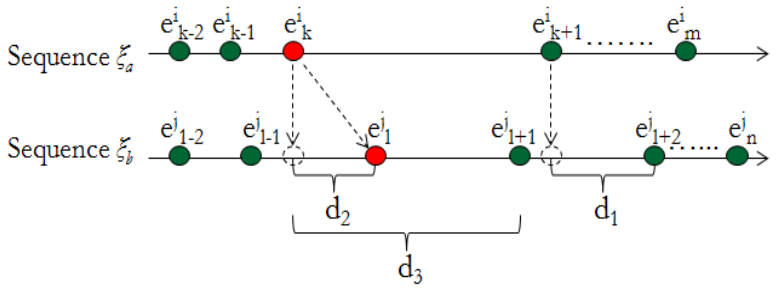

Figure 5.5: Three Categories of Forward Distances Generated From Two Event Sequences

e.g., the distance d_1 between events e^i_{k+1} and e^j_{l+2} in Figure 5.5. Since e^i_{k+1} is not temporally associated with any event on sequence ξ_b, its projected position h will be randomly and independently distributed regarding all of the events on sequence ξ_b. From the *Waiting Time Paradox* for Poisson processes we learn that the forward distance between position h and its right nearest event occurrence on sequence ξ_b follows an exponential distribution with rate $\mu = 1/\lambda_b$, where λ_b is the intensity of sequence ξ_b. Accordingly, we know that the forward distance from h to its jth right nearest event occurrence on sequence ξ_b follows a Gamma distribution with shape parameter $\alpha = j$ and scale parameter $\beta = \lambda_b$. Therefore, the distribution of the forward distances computed at e^i_{k+1} can be described by a mixture of these Gamma distribution components, which is exactly the expected null distribution $f_n(x)$ derived in Section 5.2. Hence, we can use $f_n(x)$ to characterize the distribution of the forward distances in the first category.

The forward distances in the second category are the temporal distances of associated event occurrences, e.g., the distance d_2 between events e^i_k and e^j_l in Figure 5.5. We make the assumption that the temporal distance between two associated events in a bivariate association follows a Gaussian distribution. This assumption holds in many real-world applications. For example, in the application of neuronal spike trains, one type of patterns that scientists are interested in is called *Ordered chains*, which are ordered firing sequences of neurons where times between firing of successive neurons fall within a small range around a mean and are assumed to follow a Gaussian distribution [1]. If there exist more than one bivariate association

between two event sequences, we can describe the distribution of forward distances generated from associated pairs of events as a mixture of Gaussian distributions:

$$f_g(x) = \sum_{p=1}^{N} W_p \times Gaussian(x, \mu_p, \delta_p), \tag{5.5}$$

where N stands for the number of Gaussian distribution components, which equals the number of bivariate associations existing in the data. The term $Gaussian(x, \mu_p, \delta_p)$ denotes the pth Gaussian distribution component with mean μ_p and standard deviation δ_p. This term represents the distribution of the temporal distances of associated pairs of events in the pth bivariate association. W_p denotes the weight of the pth Gaussian component and can be estimated by $k_p / \sum_{p=1}^{N} k_p$, where k_p represents the number of forward distances following this Gaussian distribution.

The forward distances in the third category are from associated events on sequence ξ_a to events that are randomly and independently distributed on sequence ξ_b, e.g., the distance d_3 between events e_k^i and e_{l+1}^j in Figure 5.5. Since e_k^i is only temporally associated with e_l^j on sequence ξ_b, we can consider a new event sequence ξ_b' that contains all of the events on sequence ξ_b but e_l^j. Let λ_b' denote the intensity of this new sequence. We estimate λ_b' by $(n-1)/L$, where L represents the temporal length of sequence ξ_b. Since e_k^i has no temporal association with any event on sequence ξ_b', its projected position h will be randomly and independently distributed regarding all of the events on this sequence. Similar as before, we can infer that the forward distance between position h and its right nearest event occurrence on sequence ξ_b' follows an exponential distribution with rate $\mu = 1/\lambda_b'$ and that the forward distance from h to its jth right nearest event occurrence on sequence ξ_b' follows a Gamma distribution with shape parameter $\alpha = j$ and scale parameter $\beta = \lambda_b'$. Therefore, the distribution of the forward distances computed at event e_k^i can be described by a mixture of these Gamma distribution components. Hence, we can express the distribution of the forward distances in the third category as:

$$f_n'(x) = \sum_{j=1}^{M} W_j \times g(x, j, \lambda_b'), \tag{5.6}$$

where M means the number of individual Gamma distribution components, which equals the number of forward distances the first associated event on sequence ξ_a

has. The term $g(x, j, \lambda'_b)$ stands for the Gamma distribution that the distance to the jth right neighbor follows. W_j is the weight of the jth Gamma component. We can estimate the weights for these Gamma components in the same way as we did for the weights of the Gamma components in the expected null distribution. There are, however, approximately the same amount of forward distances contributing to each Gamma component, if we compute only a limited number c of forward distances at each event on sequence ξ_a. Consequently, we can estimate each weight W_j by $1/(c-1)$.

Based on the previous analysis, we can derive a distribution of all forward distances from all of the events on a finite sequence ξ_a to all of the events on a finite sequence ξ_b as:

$$f_t(x) = (\frac{T - c \times \sum_{p=1}^{N} k_p}{T}) f_n(x) + (\frac{\sum_{p=1}^{N} k_p}{T}) f_g(x) + (\frac{(c-1) \times \sum_{p=1}^{N} k_p}{T}) f'_n(x), \tag{5.7}$$

where T denotes the total number of forward distances computed between the two sequences, N represents the number of bivariate associations and c denotes the number of distances computed at each event occurrence. If k_p denotes the number of temporal distances in the pth bivariate association, we can calculate $c \times \sum_{p=1}^{N} k_p$ distances from the associated events on sequence ξ_a, so approximately $T - c \times \sum_{p=1}^{N} k_p$ distances will follow the expected null distribution $f_n(x)$. Furthermore, there are around $\sum_{p=1}^{N} k_p$ distances generated by the associated pairs of events between the two sequences and we can use $f_g(x)$ to express the distribution of these distances. Finally, $f'_n(x)$ stands for the distribution of the rest of the distances. We determine the normalized weight for each of the distribution components in the theoretical function by dividing the number of distances following this component by the total of distances computed between the two sequences.

5.5.2 Using Kernel Density Estimation to Approximate the Observed Distribution

After computing the forward distance for every pair of event occurrences from different sequences, we approximate the observed distribution of these forward distances by using a kernel density estimation technique. Kernel density estimation is

commonly used as a process for smoothing data, which is accomplished by replacing each data point with a kernel density estimator, such as Gaussian. To compute the density for a data point, a Gaussian kernel estimator takes the observed value of the data point as its mean and assigns its standard deviation to a fixed value. These densities of Gaussian kernel estimators are then summed over all data points, providing a continuous probability density distribution. As compared to data binning techniques, kernel density estimation creates a smoother distribution curve than histograms, which prevents loss of information from a large number of observed values being placed into one bin.

In our research work, we adopt Botev's method [6], which is the state-of-the-art adaptive kernel density estimation method. The key idea of this method is to view the kernel from which the estimator is constructed as the transition density of a linear diffusion process that has a given limiting and stationary probability density. In addition, the method also includes a plug-in non-parametric bandwidth (i.e., the standard deviation of Gaussian kernel estimators) selection algorithm that does not require a preliminary normal model for the data. Experimental results show that this method results in a simple and intuitive kernel estimator with substantially reduced asymptotic bias and mean square error compared to other related work. The input variables of this method are the data set and the number of data points over each of which the density estimate is computed. In our work, we set the second variable of this method to the number of forward distances, which means that the method constructs a Gaussian kernel estimator and computes a density for each forward distance. We use $f_o(x)$ to denote the generated density curve of the observed distribution.

5.5.3 Zero-crossing Points on the Second Derivative Curve of a Gaussian Distribution

The proposed approach for bivariate association detection is based on the observation that: if we use μ and σ to represent respectively the mean and the standard deviation of a Gaussian distribution, the inflection points on the first derivative curve of this Gaussian are at $\mu \pm \sigma$, causing the two zero-crossing points on its second

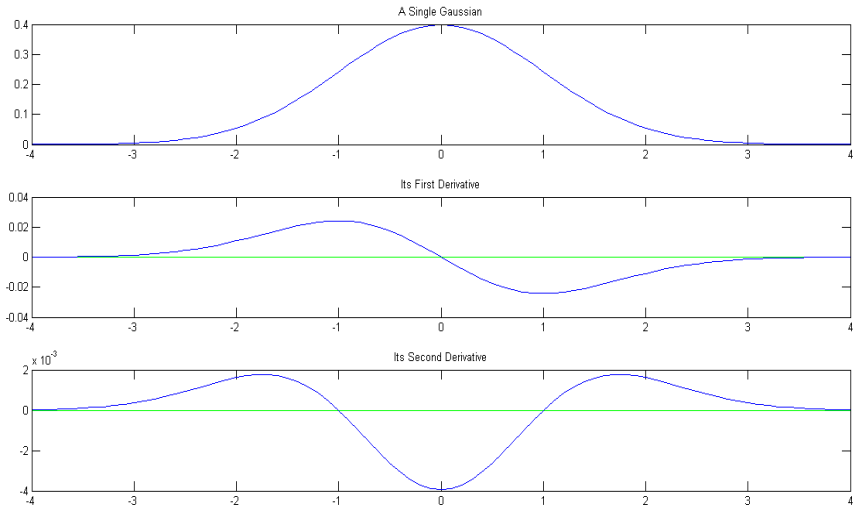

Figure 5.6: A Gaussian Distribution with its First and Second Derivatives

derivative curve to be at $\mu \pm \sigma$ as well [2]. Figure 5.6 presents an example, where a Gaussian distribution with $\mu = 0$ and $\sigma = 1$ is plotted at the top level and its first and second derivative curves are respectively shown at the middle and bottom level. This figure illustrates clearly that the two zero-crossing points on the second derivative curve of this Gaussian are at ± 1. Based on the positions of the zero-crossing points, we can easily estimate the mean and the standard deviation of this Gaussian distribution.

5.5.4 Regarding Statistically Significant Region Detection as a Least Squares Curve-fitting Problem

Based on the analysis of the theoretical distribution $f_t(x)$, we can treat statistically significant region detection as a least squares curve-fitting problem, where we initialize and adjust the parameters of the Gaussian distributions in $f_t(x)$ to optimally fit the curve of the observed distribution $f_o(x)$. Algorithm 1 is used to discover statistically significant regions from the observed distribution. In this algorithm, we first generate the second derivative curve of the observed distribution and obtain the pairs of zero-crossing points from the resulting curve. For each pair of zero-crossing points (x_l, x_r), we use the statistical test described in Section 5.4.2

to check if the number of forward distances falling into the region $[x_l, x_r]$ in the observed distribution is statistically significant, assuming that all the forward distances are distributed according to the expected null distribution $f_n(x)$. Suppose m pairs are "*statistically significant*", each of which indicates a Gaussian distribution. We can, therefore, approximate $f_t(x)$ by using the m Gaussian distributions. For each statistically significant pair (x'_l, x'_r), we estimate the mean of the Gaussian distribution by averaging the values of these two points:

$$\mu = \frac{x'_l + x'_r}{2}. \tag{5.8}$$

Accordingly, we determine the standard deviation of this Gaussian by taking half of the absolute value of the difference between the two points:

$$\sigma = \frac{|x'_l - x'_r|}{2}. \tag{5.9}$$

In this manner, the means and standard deviations of the m Gaussian distributions in $f_t(x)$ can be initially estimated. Once these parameters have been determined, we estimate the number of forward distances following each Gaussian distribution in $f_t(x)$ by using the system of linear equations:

$$\sum_{x_j \in U} f_t(x_j) \equiv \sum_{x_j \in U} f_o(x_j), \tag{5.10}$$

where x_j stands for a forward distance and U denotes the set of all the forward distances computed between two event sequences. After assigning initial values to the Gaussian distributions' parameters in $f_t(x)$, we apply the *Levenberg-Marquardt* (LM) algorithm [19] to adjust these parameters. Starting from an initial set of parameters, the LM algorithm iteratively modifies these parameters until a local minimum is reached in the sum-of-squared error between the observed distribution and the approximating function $f_t(x)$. The measurement the LM algorithm minimizes can be written as:

$$e_n \equiv \sqrt{\sum_{x_j \in U} (f_t(x_j) - f_o(x_j))^2}. \tag{5.11}$$

Finally, a statistically significant region will be reported if the number of forward distances following a Gaussian distribution, whose parameters have been heuristically determined, in $f_t(x)$ is also statistically significant, in contrast to the expected null distribution.

Algorithm 1 Approximation Algorithm (AA)

Input: $f_o(x)$ - the observed distribution of forward distances; $f_n(x)$ - the expected null distribution of forward distances

Output: S - a set of statistically significant regions

1: Generate the second derivative curve of $f_o(x)$ and let $f''_o(x)$ denote the resulting curve;
2: Obtain the pairs of zero-crossing points on $f''_o(x)$;
3: **for** each pair of zero-crossing points (x_l, x_r) **do**
4: Check if the number of forward distances falling into the region [x_l, x_r] in $f_o(x)$ is statistically significant, assuming that all the forward distances are distributed according to $f_n(x)$;
5: Suppose m pairs are "statistically significant", each of which indicates a Gaussian distribution. Estimate the m Gaussian distributions' parameters using Equation 5.8, Equation 5.9 and Equation 5.10;
6: Approximate the theoretical distribution $f_t(x)$ using these Gaussian distributions and their parameters are adjusted by the *Levenberg-Marquardt* algorithm as it minimizes Equation 5.11;
7: **for** each Gaussian distribution $\Phi(\mu, \sigma)$ in $f_t(x)$ **do**
8: Verify if the number of forward distances falling into the region $[\mu-\sigma, \mu+\sigma]$ in $f_o(x)$ is statistically significant, assuming that all the forward distances are distributed according to $f_n(x)$;
9: **if** the region $[\mu - \sigma, \mu + \sigma]$ is "statistically significant" **then**
10: Store pair (μ, σ) into S;
11: **return** S

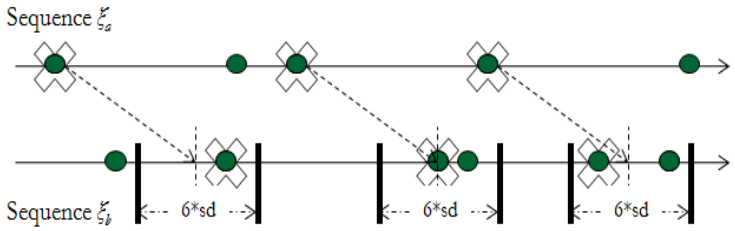

Figure 5.7: Making Associated Pairs of Events

Figure 5.8 provides for us an example of the observed probability density curve (the blue curve) and the theoretical probability density curve (the red curve) for a bivariate association with 5000 instances (i.e., associated pairs of event occurrences), demonstrating how similar the two curves are to each other.

5.5.5 Modeling Bivariate Associations

If two event sequences are temporally associated, we can detect their bivariate associations by identifying from the observed distribution the bell-shaped regions that show statistically significant deviation from the expected null distribution. Each of the regions corresponds to a bivariate association. We have assumed in Section 5.5.1 that the temporal distance between two associated events from different sequences follow a Gaussian distribution with a relatively small standard deviation around a mean distance. Given a statistically significant region, the *mean* temporal distance of associated events in this bivariate association can be estimated as the mean of the distances inside the region, and the *standard deviation* of the temporal distances can be approximated as the standard deviation of the distances within the region.

We retrieve the bivariate association A^d_{ab} from two sequences ξ_a and ξ_b by looking for a set of associated pairs of event occurrences. For an event occurrence on ξ_a, we only think of the event occurrences on ξ_b that are in the time interval $[mean - 3 * sd, mean + 3 * sd]$, where $mean$ and sd denote the mean respectively the standard deviation of this bivariate association, and three standard deviations account for 99.7% of associated event occurrences on sequence ξ_b being contained in this interval. If there are multiple event occurrences on ξ_b that can be paired with an

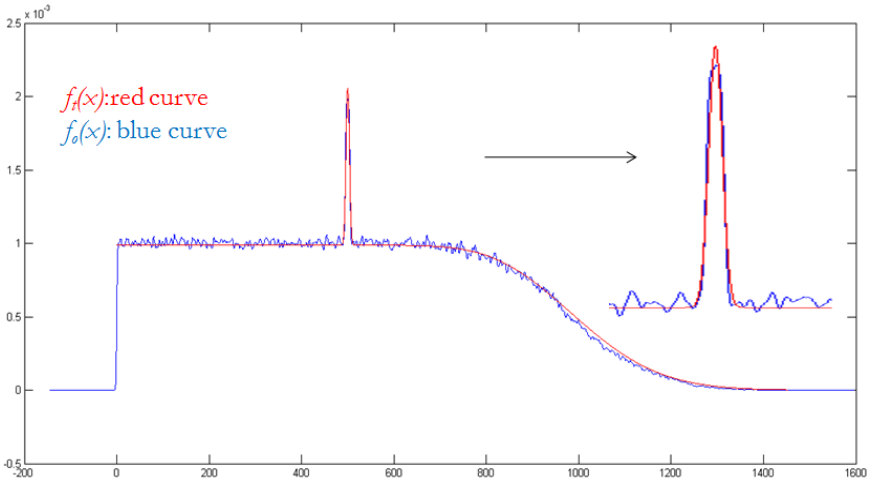

Figure 5.8: Observed probability density curve (blue curve) versus theoretical probability density curve (red curve) for a bivariate association with 5000 occurrences (associated pairs of events). A zoomed-in view reveals how similar the two curves are to each other in the region bounded by the "statistically significant" pair of zero-crossing points.

event occurrence on ξ_a within the time interval, we just select the event occurrence on ξ_b that is nearest to the mean temporal distance in that interval. In addition, if more than one event occurrence on ξ_a can be paired with an event occurrence on ξ_b, we will choose the occurrence on ξ_a that happens earlier than the rest to construct the associated pair. Figure 5.7 demonstrates this process. In this figure, the events marked by crosses are the associated pairs determined by our method.

Chapter 6

Detecting Multivariate Associations

Given a set of bivariate associations detected from multiple event sequences, we build a directed graph by using these bivariate associations. In the graph, a vertex denotes a bivariate association. We add a directed edge to the graph from vertex v_i to vertex v_j if the event sequence where the bivariate association associated with v_i ends is the same sequence where the bivariate association associated with v_j starts. Based on the graph, we use Algorithm 2 to discover multivariate associations. In this algorithm, we first search the graph for a *root*, i.e., a vertex having no incoming edges. There can be more than one root in the graph and we randomly select one to begin our discovery process. In the case where all the vertices in a graph have incoming edges, we regard each vertex as a root. Given a root r_i, we retrieve all of the paths beginning with r_i from the graph and store them in a path set P_i. Starting from a randomly chosen path p in P_i, we search along p for its maximum sub-path sp_{max}, which is verified to be a multivariate association by Algorithm 3. If such a path sp_{max} is found, it will be stored in a temporary set $Temp$. We start our search again with another randomly chosen path in the remaining part of P_i and repeat Step 11-13 until all of the paths in P_i are processed. After iterating all of the roots in R, we copy every detected multivariate association from $Temp$ to a result set S and delete all the vertices used to constitute this multivariate association as well as their associated edges from the graph. Finally, since it is possible that some multivariate associations are actually the sub-paths of others (e.g., $v_a \rightarrow v_b$ and $v_b \rightarrow v_c$ are two sub-paths of $v_a \rightarrow v_b \rightarrow v_c$, but $v_a \rightarrow v_c$ is regarded as an independent association), we remove redundant associations from S. We repeat Step 1-18 until the graph is

empty.

Algorithm 2 Multivariate Association Discovery (MAD)

Input: $G(V, E)$ - bivariate association graph
Output: S - a set of paths, each representing a multivariate association

```
1:  while G is not empty do
2:    for each vertex v_i ∈ G do
3:      if v_i has no incoming edges then
4:        Store v_i in a root set R;
5:    if R is empty then
6:      for each vertex v_i ∈ G do
7:        Store v_i in R;
8:    for each root r_i ∈ R do
9:      Retrieve all of the paths starting at r_i from G and store them in a set P_i;
10:     for each path p ∈ P_i do
11:       Starting from r_i, search along p for its maximum sub-path sp_max, which
          is verified to be a multivariate association by Algorithm 3;
12:       Store sp_max in a temporary set Temp;
13:   for each path p ∈ Temp do
14:     Store p in a result set S;
15:     Remove all the vertices on p as well as their associated edges from G;
16:   for each path p ∈ S do
17:     Remove all the sub-paths of p from S;
18: return S
```

Algorithm 3 is based on the assumption that the frequency of a multivariate association should be statistically significant in the data. That means when there exists a multivariate association on a path, the number of chains of connected associated pairs of events traversing this path should be larger than expected under the assumption that these chains form by chance. Given a path $p = v_1 \dots v_l$ and a significance level α_0, we will search for a multivariate association on p in the following way. Let k be the number of chains of connected associated pairs of events starting from v_1 to v_{i-1} on p, and let x be the number of chains of connected associated pairs of events starting from v_1 to v_i on p. We use m and n to denote the number of associated pairs of events in v_{i-1} and v_i, respectively. If we randomly select an associated pair of events from v_{i-1}, the probability δ that this associated pair is on a chain starting from v_1 to v_i can be estimated by $\delta = k/m$. If we regard the event of choosing an associated pair of events from v_i to extend one of the chains starting

from v_1 until v_{i-1} as a Bernoulli experiment, which is repeated independently n times with the success probability equal to δ, we can use the Binomial distribution as the null hypothesis to compute the probability α that x chains get extended, i.e., $x \sim Binominal(n, \delta)$. Since we conduct multiple hypothesis tests on p, we adjust the significance level α_0 again by using $\alpha_0' = \alpha_0/l$, where l denotes the number of vertices on p. If α is smaller than α_0', we continue to check for the next vertex on p; otherwise, we output the sub-path of p from v_1 to v_{i-1} as a multivariate association.

Algorithm 3 Candidate Association Verification (CAV)

Input: $p = v_1 \dots v_l$ - a path in G; α_0 - significance level
Output: sp_{max} - a multivariate association

index = 1;
for each vertex v_i ($2 \leq i \leq l$) on p **do**
  $k \longleftarrow$ the number of chains of connected associated pairs of events starting from v_1 until v_{i-1} on p;
  $x \longleftarrow$ the number of chains of connected associated pairs of events starting from v_1 until v_i on p;
  $m \longleftarrow$ the number of associated pairs in v_{i-1};
  $n \longleftarrow$ the number of associated pairs in v_i;
  $\delta = k/m \longleftarrow$ the probability that an associated pair in v_{i-1} is on a chain starting from v_1 until v_{i-1};
  $\alpha = Binomial(x, n, \delta) \longleftarrow$ the probability of observing x chains starting from v_1 until v_i;
  if $\alpha \leq \alpha_0/l$ **then**
    index = i;
  else
    break;
return $sp_{max} = v_1 \dots v_{index}$

Chapter 7

Evaluating Bivariate Association Mining Methods on Synthetic Data Sets

To study the generality and robustness of our proposed methods, we conducted four groups of experiments. In each group, we generated two event sequences with a length of 2×10^7 time units and implanted varying numbers of occurrences of a bivariate association and "*noise*" event occurrences (i.e., those that do not participate in the bivariate association) into the two sequences, so that each event sequence had a total of 1×10^4 occurrences. The temporal distance between two associated events in the bivariate association followed a Gaussian distribution. The significance level for detecting implanted bivariate associations was set to 10^{-11}. The number of forward distances computed at each event occurrence was set to 20. We evaluated the performance of our methods using the F-measure, which is computed as:

$$F = 2 \cdot \frac{p \cdot r}{p + r}, \tag{7.1}$$

where p denotes the precision and r represents the recall of a method. In our research study, *p* stands for the number of retrieved "*true*" bivariate association occurrences divided by the total of returned bivariate association occurrences and *r* stands for the number of retrieved "*true*" bivariate association occurrences divided by the total of bivariate association occurrences implanted into the two event sequences. F-measure reaches its best value at 1 and worst score at 0. In the following, we name the method, which uses a histogram of forward distances to search

Table 7.1: BAM-I: % of Bivariate Association Occurrences vs. Mean Temporal Distance

Per. \ Mean	20	50	100	200	500	1000	3000	5000
10%	0.959	0.958	0.958	0.957	0.960	0.961	0.959	0.958
30%	0.982	0.980	0.982	0.981	0.983	0.982	0.982	0.982
50%	0.991	0.992	0.993	0.992	0.992	0.992	0.993	0.992
70%	0.995	0.996	0.995	0.996	0.996	0.996	0.995	0.995
90%	0.998	0.998	0.996	0.997	0.998	0.998	0.997	0.997

for bivariate associations, *BAM-I*; we name the method, which treats bivariate association discovery as a least squares curve-fitting problem, *BAM-II*. In BAM-I, to find a proper bin size for the histogram of forward distances, we set the number of iterations of Shimazaki's method to 50000.

In the first group of experiments, we varied both the percentage of occurrences that belong to the implanted bivariate association from 10% to 90% of the total of occurrences on the two event sequences, and the mean temporal distance between two associated events in the range of 20 to 5000 time units. The standard deviation of the temporal distances in the bivariate association was fixed to 2.5 time units. Table 7.1 shows the result of BAM-I, Table 7.2 shows the result of BAM-II. We can see from these tables that our methods constantly performed well, achieving high F-measure scores in all of the cases, which means that the performance of our methods is not affected by the mean temporal distance between two associated events. Both of the methods are also insensitive to noise, as they detected the implanted bivariate association even in the case where the number of "*noise*" event occurrences was 9 times larger than the number of bivariate association occurrences.

In the second group of experiments, we changed both the percentage of bivariate association occurrences from 10% to 90% of the total number of occurrences on the sequences, and the standard deviation of the temporal distances in the bivariate association between 2.5 and 50 time units. This time the mean temporal distance was fixed to 500 time units. Table 7.3 shows the result of BAM-I, Table 7.4 shows

Table 7.2: BAM-II: % of Bivariate Association Occurrences vs. Mean Temporal Distance

Per. \ Mean	20	50	100	200	500	1000	3000	5000
10%	0.985	0.986	0.985	0.985	0.986	0.985	0.985	0.985
30%	0.988	0.989	0.987	0.986	0.988	0.988	0.987	0.988
50%	0.992	0.993	0.993	0.992	0.993	0.992	0.992	0.991
70%	0.995	0.995	0.996	0.995	0.995	0.996	0.995	0.995
90%	0.996	0.996	0.997	0.997	0.998	0.997	0.996	0.996

Table 7.3: BAM-I: % of Bivariate Association Occurrences vs. Standard Deviation of Temporal Distances

Per. \ SD	2.5	5	10	20	30	40	50
10%	0.952	0.886	0.832	0.749	0.641	0.213	0.104
30%	0.982	0.968	0.935	0.902	0.859	0.808	0.771
50%	0.991	0.981	0.970	0.944	0.928	0.913	0.903
70%	0.994	0.989	0.980	0.963	0.957	0.937	0.927
90%	0.997	0.993	0.985	0.972	0.962	0.952	0.943

the result of BAM-II. We observed from these tables that our methods obtained high scores in the cases where the standard deviation of the temporal distances was relatively small. Although the performance of our methods gradually declines as the standard deviation of the implanted bivariate association enlarges, in many real-world applications the standard deviation of a bivariate association will not be extremely large, otherwise, it would be hard to argue even for the existence of such a temporal association. Furthermore, from these experimental results we learned that it was difficult for our methods to detect this implanted bivariate association in the situations where the number of "*noise*" event occurrences was much larger than the number of bivariate association occurrences.

In the third group of experiments, we varied both the percentage of bivariate association occurrences from 10% to 90% of the total number of occurrences on the two event sequences, and the number of forward distances computed at each

Table 7.4: BAM-II: % of Bivariate Association Occurrences vs. Standard Deviation of Temporal Distances

Per. \ SD	2.5	5	10	20	30	40	50
10%	0.985	0.973	0.932	0.884	0.822	0.541	0.475
30%	0.987	0.980	0.960	0.921	0.887	0.838	0.809
50%	0.992	0.987	0.979	0.953	0.946	0.929	0.910
70%	0.995	0.992	0.985	0.974	0.956	0.948	0.931
90%	0.997	0.995	0.989	0.980	0.971	0.967	0.952

Table 7.5: BAM-I: % of Bivariate Association Occurrences vs. # of Forward Distances Computed at Each Event Occurrence

Per. \ #	1	4	8	16	32	64	128
10%	0.948	0.952	0.957	0.957	0.958	0.959	0.958
30%	0.953	0.978	0.981	0.982	0.980	0.982	0.981
50%	0.969	0.988	0.991	0.993	0.991	0.992	0.991
70%	0.983	0.990	0.994	0.995	0.996	0.996	0.994
90%	0.984	0.992	0.996	0.996	0.998	0.997	0.996

event occurrence in the range of 1 to 128. We assigned 500 time units to the mean temporal distance and 2.5 time units to the standard deviation of the implanted bivariate association. Table 7.5 presents the result of BAM-I, Table 7.6 presents the result of BAM-II. The experimental results show that our methods consistently worked well, obtaining high scores in all of the data sets, which indicates that the performance of our methods are not influenced by the number of forward distances computed at each event occurrence, as long as the implanted bivariate association can be completely captured by the generated forward distances.

In the fourth group of experiments, we aim to observe how the performance of BAM-I is affected by the bin size of a histogram of forward distances. Although we adopt Shimazaki's method in the current implementation to learn the optimal bin size for a histogram, other binning techniques can also be applied in our method. Based on the nature of each technique, we can have histograms with different bin

Table 7.6: BAM-II: % of Bivariate Association Occurrences vs. # of Forward Distances Computed at Each Event Occurrence

Per. \ #	1	4	8	16	32	64	128
10%	0.966	0.975	0.984	0.985	0.985	0.984	0.986
30%	0.973	0.981	0.986	0.988	0.988	0.987	0.989
50%	0.979	0.984	0.993	0.992	0.994	0.993	0.992
70%	0.983	0.989	0.995	0.996	0.996	0.995	0.996
90%	0.988	0.992	0.996	0.998	0.997	0.998	0.997

sizes. For example, Sturge's rule [32] tends to generate a large bin size, leading to a histogram with only a few of bins. In this group, we changed both the percentage of occurrences that belong to the bivariate association from 10% to 90% of the total number of occurrences on the sequences, and the bin size, which we use to create a histogram to estimate the observed distribution of forward distances, between 5 and 500 time units. We assigned 500 time units to the mean temporal distance and 2.5 time units to the standard deviation of the implanted bivariate association. Different from what we did in other groups of experiments where we learn the bin size automatically by using Shimazaki's method, the bin size is manually determined in this group. Note that when we assign a small value to the bin size, the region where the implanted bivariate association exists in the observed distribution may span more than one bin. To search for an appropriate region size for the bivariate association, our method merges adjacent bins if they are statistically significant individually, and reports the merged region if it is also statistically significant. Table 7.7 presents that BAM-I performed well while we used a bin size close to the standard deviation of the bivariate association, which indicates that most of the temporal distances between associated events fall into one bin and make this bin statistically significant assuming the expected null distribution. When the bin size is much larger than the standard deviation of the bivariate association, our method becomes ineffective because a substantial number of forward distances, which do not belong to the implanted bivariate association, fall into the bin that contains the temporal distances of associated events. Furthermore, this table also presents that the performance

Table 7.7: BAD-I: % of Bivariate Association Occurrences vs. Bin Size

Per. \ Bin Size	5	10	20	50	100	300	500
10%	0.952	0.948	0.944	0.810	0.704	0.609	0.355
30%	0.983	0.981	0.976	0.943	0.903	0.859	0.647
50%	0.993	0.991	0.987	0.974	0.957	0.939	0.797
70%	0.995	0.993	0.992	0.987	0.977	0.976	0.900
90%	0.997	0.997	0.996	0.995	0.992	0.992	0.967

of BAM-I gradually declined as more and more "*noise*" event occurrences were implanted into the sequences.

Chapter 8

Empirical Study On Multivariate Motif Discovery

We apply our methodology to detect multivariate motifs from multiple time series sequences and compare it with the work of Vahdatpour *et. al.* [34], which is currently the most effective work for multivariate motif discovery. This method adopts an approach that uses the result of a univariate motif discovery algorithm as the input of the second stage in their method and compose multivariate motifs by using graph clustering. In the following sections, we call the multivariate association discovery method, which adopts the approach of using histograms of forward distances to search for bivariate associations, *MAD-I*; we name the other method, which uses the approach that regards bivariate association discovery as a least squares curve-fitting problem, *MAD-II*. We verify the resulting multivariate motifs by comparing them with the ground truth if it is available.

8.1 Collecting Event Sequences from Applications

Depending on specific applications, we can adopt different approaches to transform raw temporal data into event sequences. In the case of multivariate motif discovery, we can ignore the regions in a univariate time series that are not part of any univariate motif occurrence. We further simplify the time series containing univariate motif occurrences by representing a motif occurrence using its starting position in the time series. If we regard each such point as an event, the univariate time series can be transformed into an event sequence (see Figure 8.1).

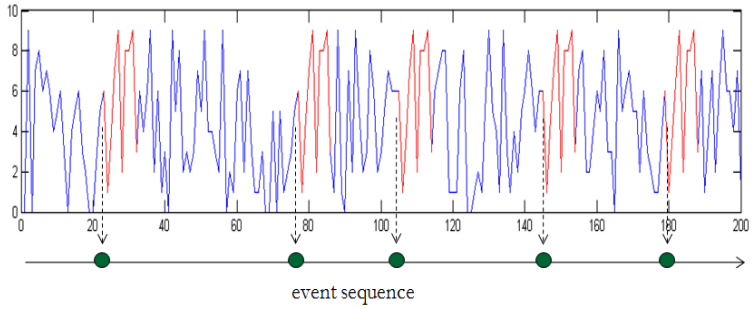

Figure 8.1: Transforming a univariate time series into an event sequence, where a motif occurs five times, indicated by red color.

In the case that we need to search for univariate motifs from the data set, we adopt the MK algorithm [25], which is an improved version of Chiu's algorithm. This algorithm has two strengths: a) it is currently one of the most effective methods in the literature; b) it can reduce the effect of noise in the data via random projections. In our experiments, we generate univariate motifs of variable lengths by assigning distinct sets of parameter values, e.g., motif length, to Chiu's algorithm. We then transform each found univariate motif into an event sequence, as described above. For a fair comparison, we evaluate both our methods and Vahdatpour's method using the same set of event sequences.

8.2 Experiment on Synthetic Multivariate Time Series Data Sets

To evaluate our methods in terms of generality and robustness, we conducted three groups of experiments. In each group, we generated a set of univariate time series with a length of 2×10^7 time units. We implanted varying numbers of occurrences of a multivariate motif and "*noise*" univariate motif occurrences (i.e., those that do not participate in the multivariate motif) into these time series, so that each univariate time series had a total of 1×10^4 occurrences. Both the length of "*noise*" univariate motif occurrences and the length of univariate elements in the multivariate motif equaled 20 time units. Figure 8.2 shows a small snapshot of one of the data sets we used for evaluating the performance of a method. In our methods,

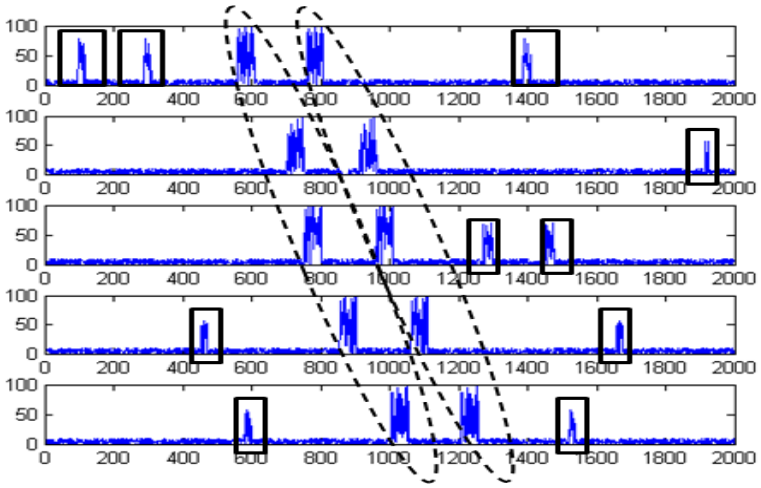

Figure 8.2: A snapshot of the data set used for evaluating the performance of a method. A rectangle represents a "*noise*" univariate motif occurrence and an ellipse denotes a multivariate motif occurrence.

we set the number of forward distances computed at each univariate motif occurrence to 20; to avoid false positives, we set the significance level of statistical tests to 10^{-11}. In MAD-I, we assigned 50000 to the number of iterations used by Shimazaki's method. In Vahdatpour's method, we set the threshold for determining the minimum correlation of two univariate motifs to 0.05, as done by Vahdatpour *et al.* [34]. The performance was evaluated by using the F-measure. In addition, we call a bivariate association found in two univariate time series a *bivariate motif*, and we name a n-variate association detected from n univariate time series a *n-variate motif*.

In the first group of experiments, we created synthetic data sets containing five randomly generated univariate time series, where a 5-variate motif and some "*noise*" univariate motif occurrences were implanted. The 5-variate motif consisted of 4 bivariate motif components, each of which had a fixed standard deviation of temporal distances equal to 2.5 time units. We varied both the percentage of 5-variate motif occurrences from 10% to 100% of the total number of occurrences, and the mean temporal distance between 10 and 5000 time units. Table 8.1 presents the result of MAD-I, Table 8.2 shows the result of MAD-II. From the experimental results we learn that our methods not only detected this multivariate motif when

Table 8.1: MAD-I: % of Multivariate Motif Occurrences vs. Mean Temporal Distance

Per. \ Mean	10	20	200	1000	5000
10%	0.947	0.945	0.941	0.944	0.942
30%	0.952	0.956	0.949	0.953	0.957
50%	0.965	0.960	0.965	0.963	0.967
70%	0.974	0.973	0.977	0.974	0.972
90%	0.987	0.991	0.988	0.985	0.986

Table 8.2: MAD-II: % of Multivariate Motif Occurrences vs. Mean Temporal Distance

Per. \ Mean	10	20	200	1000	5000
10%	0.983	0.984	0.983	0.984	0.985
30%	0.984	0.986	0.985	0.985	0.986
50%	0.985	0.988	0.986	0.988	0.986
70%	0.990	0.989	0.988	0.990	0.988
90%	0.992	0.991	0.992	0.991	0.991

its univariate elements temporally overlap (i.e., the cases when the mean temporal distance equals 10 or 20 time units) but also found it as its univariate elements had varying temporal lags. Our methods are also robust, even in the situation where the number of "*noise*" univariate motif occurrences was 9 times larger than the number of multivariate motif occurrences. The result in Table 8.3 indicates that Vahdatpour's method detected nothing when the univariate elements of this multivariate motif were not synchronous.

In the second group of experiments, we again generated synthetic data sets containing five randomly generated univariate time series, where a 5-variate motif and some "*noise*" univariate motif occurrences were implanted. Each bivariate motif components had now a fixed mean of temporal distances equal to 500 time units, and we varied both the percentage of 5-variate motif occurrences from 10% to 100% of the total number of occurrences, and the standard deviation of the bivariate motif components between 10 and 100 time units. Table 8.4 shows the result of MAD-I,

Table 8.3: VAH: % of Multivariate Motif Occurrences vs. Mean Temporal Distance

Per. \ Mean	10	20	200	1000	5000
10%	0.889	0.496	0.0	0.0	0.0
30%	0.959	0.519	0.0	0.0	0.0
50%	0.974	0.531	0.0	0.0	0.0
70%	0.985	0.553	0.0	0.0	0.0
90%	0.990	0.564	0.0	0.0	0.0

Table 8.4: MAD-I: % of Multivariate Motif Occurrences vs. Standard Deviation of Temporal Distances

Per. \ Vari.	2.5	5	10	20	30
10%	0.946	0.923	0.911	0.751	0.592
30%	0.950	0.947	0.938	0.889	0.747
50%	0.966	0.953	0.941	0.926	0.880
70%	0.972	0.964	0.955	0.943	0.904
90%	0.982	0.973	0.965	0.952	0.922

Table 8.5 presents the result of MAD-II. From these two tables we observe that: the larger the standard deviation of the bivariate motif components becomes, the more difficult it is to detect the multivariate motif, since our methods may fail to detect some of the bivariate motif occurrences, which are used to constitute multivariate motif occurrences. We also applied Vahdatpour's method on the same data sets and the result shows it never found this multivariate motif.

In the third group of experiments, we evaluated the performance of our methods as the significance level α_0 used to detect bivariate motif components and multivariate motifs changed. We created synthetic data sets containing five randomly generated univariate time series, where a 5-variate motif and some "*noise*" univariate motif occurrences were implanted. The mean temporal distance of the bivariate motif components was set to 500 time units, and their standard deviations were set to 2.5 time units. We varied both the percentage of 5-variate motif occurrences from 10% to 100% of the total number of occurrences, and the significance level α_0

Table 8.5: MAD-II: % of Multivariate Motif Occurrences vs. Standard Deviation of Temporal Distances

Per. \ Vari.	2.5	5	10	20	30
10%	0.977	0.964	0.918	0.861	0.795
30%	0.986	0.969	0.951	0.899	0.847
50%	0.986	0.976	0.961	0.916	0.880
70%	0.990	0.983	0.972	0.934	0.904
90%	0.992	0.984	0.974	0.948	0.922

Table 8.6: MAD-I: % of Multivariate Motif Occurrences vs. Significance Level α_0

Per. \ α_0	0.01	0.001	10^{-6}	10^{-9}	10^{-11}
10%	0.945	0.942	0.946	0.939	0.943
30%	0.956	0.953	0.952	0.952	0.956
50%	0.962	0.964	0.965	0.967	0.970
70%	0.974	0.972	0.980	0.976	0.971
90%	0.988	0.988	0.989	0.987	0.990

between 0.01 and 10^{-11}. Table 8.6 shows the result of MAD-I, Table 8.7 shows the result of MAD-II. The experimental results indicate our methods work constantly well by achieving high F-measure scores using different significance levels.

Finally, we generated a complex synthetic data set of ten randomly generated univariate time series with a length of 2×10^8 time units, where we implanted five multivariate motifs. Each multivariate motif had 1000 occurrences. We also added 5000 "*noise*" univariate motif occurrences to each dimension in the data set. Table 8.8 lists the properties of the implanted multivariate motifs, showing that both synchronous and non-synchronous multivariate motifs were included in this data set. Table 8.9 illustrates that MAD-I retrieved all of the occurrences of the bi-variate and the 3-variate motifs from the data, leading to a score of 1.0 for each. Although it missed several occurrences, MAD-I still obtained a score of 0.959 for the 5-variate motif, 0.942 for the 8-variate motif and 0.967 for the 10-variate motif. Similar observations can be made from the results of MAD-II. Both of our methods

Table 8.7: MAD-II: % of Multivariate Motif Occurrences vs. Significance Level α_0

Per. \ α_0	0.01	0.001	10^{-6}	10^{-9}	10^{-11}
10%	0.983	0.984	0.982	0.982	0.985
30%	0.983	0.985	0.984	0.984	0.986
50%	0.986	0.988	0.986	0.985	0.986
70%	0.990	0.991	0.990	0.989	0.990
90%	0.991	0.991	0.992	0.990	0.992

Table 8.8: The Properties of Implanted Multivariate Motifs

Properties \ Motifs	bi-variate	3-variate	5-variate	8-variate	10-variate
Mean	2	20	800	3000	5000
Standard Deviation	0.5	2.5	10	10	5
Dimensions	1-2	3-5	1-5	1-8	1-10

detected 88 bivariate motifs and 165 n-variate motifs (n$\geq$3) totally from the data. Vahdatpour's method performed well on the bi-variate and the 3-variate motifs, but it did not detect other non-synchronous multivariate motifs, leading to its scores in these cases being 0.

Figure 8.3 shows how the three methods scale when we increase the dimensionality of a synthetic data set (i.e., the number of univariate time series) with an embedded multivariate motif spanning all of the dimensions. Each univariate time series in the data set has a length of 2×10^7 time units. We set the mean and stan-

Table 8.9: The F-measure Performances of MAD-I, MAD-II and Vahdatpour's method on a Complex Synthetic Data Set

Methods \ Motifs	bi-variate	3-variate	5-variate	8-variate	10-variate
MAD-I	1.0	1.0	0.959	0.942	0.967
MAD-II	1.0	1.0	0.976	0.955	0.980
VAH	0.992	0.853	0.0	0.0	0.0

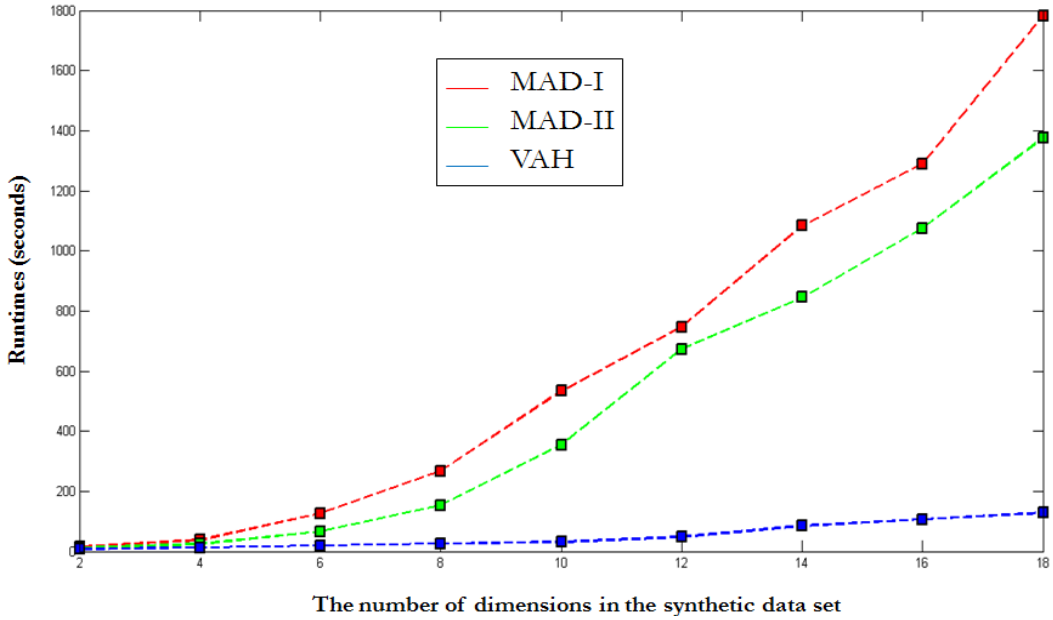

Figure 8.3: The scalability of MAD-I, MAD-II and Vahdatpour's Method with Increasing the Dimensionality of a Synthetic Data Set

dard deviation of the bivariate motif components to 10 respectively 2.5 time units, so both of the methods can find this multivariate motif. We implanted 5000 occurrences of the multivariate motif into the data set and added 5000 "*noise*" univariate motif occurrences to each of the dimensions. As the graph illustrates, MAD-I is computationally more expensive, but still feasible. MAD-I takes around 29 minutes to detect 266 multivariate motifs from this synthetic data set. Our methods need to detect associations from any of two univariate motifs, search for the associated pairs and construct multivariate motifs, while for Vahdatpour's method, increasing the dimensionality of the data set just affects the calculations required for updating the weight of an edge between two univariate motifs in the graph.

8.3 Experiment on Real-world Multivariate Time Series Data Sets

To explore the utility of our methods in real applications, we applied MAD-I, which adopts the approach of using histograms of forward distances to detect bivariate associations, on two real-world data sets. We first tested MAD-I and Vahdatpour's

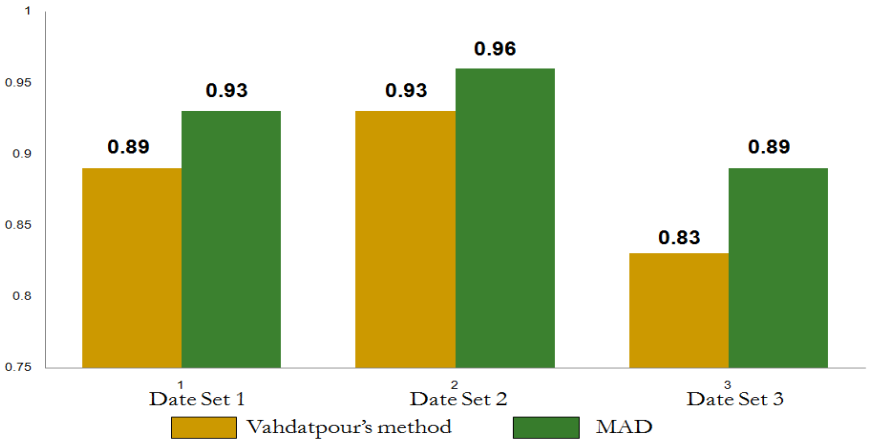

Figure 8.4: The Accuracy of Vahdatpour's method and MAD-I for Three Smart-Cane Data Sets

method on the data collected from a wearable system, called SmartCane [37]. This system is developed as a device to monitor and train senior or impaired people in their assisted walking behavior. Three data sets are generated by the sensors of the system and each has eight univariate time series. There exists a synchronous multivariate motif in these data sets, which corresponds to the normal use of the cane when walking (i.e., normal activity). We evaluated the performance of MAD-I and Vahdatpour's method in terms of accuracy, which is computed by the number of retrieved normal activity occurrences divided by the number of normal activity occurrences observed in the data. The number of forward distances computed at each univariate motif occurrence was set to 10 and the two significant levels were set to 10^{-11}. In Vahdatpour's method, we assigned 0.05 to the threshold of determining the correlation between univariate motifs, as done by Vahdatpour *et. al.* [34]. Figure 8.4 summarizes the accuracy of normal activity discovery by use of the two methods.

We further evaluated MAD-I and Vahdatpour's method using a data set where non-synchronous multivariate motifs may exist. The data set consists of recordings of shovel operations, provided by an oil company. We attempt to detect a variety of patterns, such as dig-cycles. A typical dig-cycle of a mining shovel is defined as one complete cycle for digging the surface, lifting the dug oil-sand, and finally loading it on the truck. Three different motors (i.e., *Crowd*, *Hoist*, and *Swing*) are dedicated

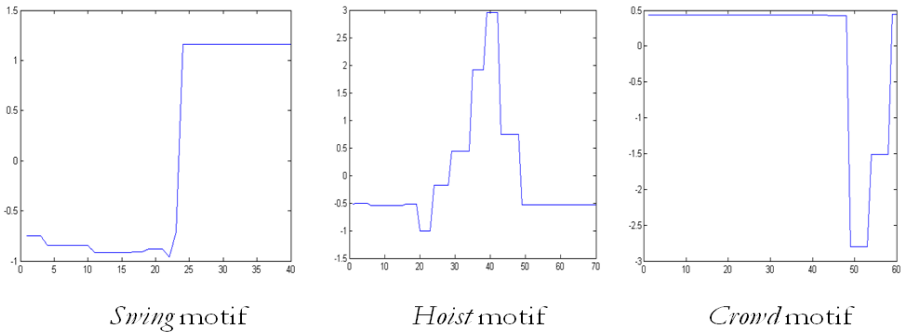

Figure 8.5: A multivariate motif occurrence retrieved by MAD-I from the shovel data.

for digging, lifting, and swinging oil-sand, respectively. The electronic power consumed by these motors (i.e., *Crowd* power, *Hoist* power, and *Swing* power) varies over time and the power profiles of the motors provide information about their activities. We ran our method and Vahdatpour's method using the same parameters as done for the SmartCane data. Figure 8.5 shows one of the multivariate motifs detected by our method involving all three motors. The temporal order of this multivariate motif is: *Swing* motif $\rightarrow$ *Hoist* motif $\rightarrow$ *Crowd* motif. Compared to the result of MAD-I, Vahdatpour's method detected a multivariate motif that consists of only the *Swing* motif and the *Crowd* motif. We are currently in the process of characterizing and interpreting the usefulness of such temporal associations found in this data set.

Chapter 9

Empirical Study On Frequent Episode Discovery

In this section, we apply our methods to discover frequent episodes from neural spike train data and compare them with the work of Patnaik *et. al.* [27], which is the currently most effective method for temporal pattern discovery in spike train data. Patnaik's work uses a depth-first pattern growth algorithm to search for frequent episodes whose time delays between event types are fixed, and it encodes temporal associations between events from different sequences by using a dynamic Bayesian network, where the conditional probabilities of a network node are learned based on the frequencies of discovered frequent episodes. We verify the resulting frequent episodes by comparing them with the ground truth.

We evaluate our methods and Patnaik's method on synthetic data collected from a mathematical model of spiking neurons [28], which simulates the interactions among spiking neurons. In this model, each spike train (a sequence of spikes generated by a neuron) follows an inhomogeneous Poisson process whose firing rate is computed by a function of the stimulus received by the neuron in the recent past. This model allows for temporal associations with variable time delays of associated spikes, which mimic the situation in conduction pathways of real neurons. We use this model to assess the performance of a method in discovering several episodes implanted into a group of synthetic data sets. Figure 9.1 illustrates these episodes, where nodes denote spike trains and directed arcs represent temporal orders of firing spikes among trains. For each episode the values above the directed arc indicate

the range of time delays between associated spikes in this temporal association. Figure 9.1 (a) presents an example of a serial episode, where spike train 3 excites two chains $\{6, 12\}$ and $\{5, 10, 15\}$. Figure 9.1 (b) gives an example of a parallel episode, where spike train 36 approximately synchronously excites three chains $\{37, 42\}$, $\{38, 45\}$ and $\{39, 40\}$. In these two episodes, the time delay between two associated spikes follows a Gaussian distribution. In our experiments, we also consider a fixed-delay episode, where the time delays between associated spikes of distinct trains are fixed. Figure 9.1 (c) gives an example of this special episode, in which spike train 63 excites three chains simultaneously while spike train 69 is activated by trains 64, 65, and 66 together. Each of the generated synthetic data sets consists of 100 spike trains (the spike trains, which are not involved in the implanted episodes, fire independently). Table 9.1 lists the data sets used for our experiments. The first column shows the name of a data set, the second column shows the length of a data set (i.e., the number of time slices in the data sequence), the third column presents the base firing rate $\hat{\lambda}_0$ of neurons used by the mathematical model for data generation, the fourth column presents the activation probability ρ of a neuron (i.e. the conditional probability that the neuron fires given its stimulus received in the recent past). In order to generate these data sets, we set the parameters of the mathematical model to the same values as done by Patnaik *et. al.* [27]. We arrange these data sets into three groups. In the A-group ($A_1 - A_4$), the data set length is 60000 ms, the activation probability ρ is set to 0.9 and the base firing rate $\hat{\lambda}_0$ is varied from 0.01 to 0.025. Similarly, in the B-group ($B_5 - B_8$), the activation probability ρ is varied from 0.8 to 0.95 keeping everything else constant. Finally, in the C-group ($C_9 - C_{11}$), the data set length is varied from 60000 ms to 120000 ms. We measure the performance of a method by using again the F-measure.

First, we summarize the performances of our methods and Patnaik's method on the data sets in the A-group. We created these data sets by changing the base firing rate $\hat{\lambda}_0$ of neurons in the mathematical model. The larger value we assign to $\hat{\lambda}_0$, the more spikes are generated on a train. Table 9.2 presents the result of MAD-I, Table 9.3 shows the result of MAD-II. From the experimental results we learn that our methods successfully detected both serial and parallel episodes by achieving high

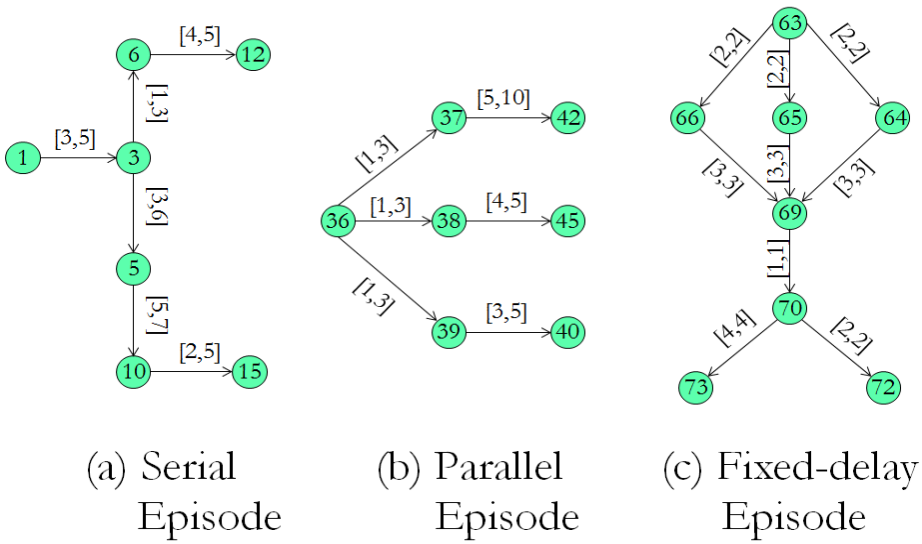

Figure 9.1: Three Episodes Implanted into Synthetic Data Sets

F-measure scores in all of the data sets. Furthermore, our methods also worked well in discovering the fixed-delay episode, especially in the cases where the base firing rate was set to a relatively small value. The result in Table 9.4 indicates that: although Patnaik's method was very effective in finding the fixed-delay episode by retrieving all of its occurrences from the data, this method detected nothing when the time delay between two associated spikes in an episode follows a Gaussian distribution. This is because of the following reason: to reduce the computational complexity, Patnaik's method only searches for fixed-delay episodes to construct a dynamic Bayesian network, which can be used to encode temporal associations among spike trains.

Second, we evaluate the competing methods by using the data sets in the B-group. This time we created the data sets by varying the activation probability ρ of a neuron in the mathematical model. The larger value we set to ρ, the more occurrences of an episode are implanted into the data. Table 9.5 shows the result of MAD-I, Table 9.6 presents the result of MAD-II. From these two tables we observe that our methods successfully discovered these implanted episodes from the data. We also applied Patnaik's method on the same data sets and the result in Table 9.7 shows that it found neither serial nor parallel episodes.

Table 9.1: Data Sets

Name	Length (ms)	Base Firing Rate $\hat{\lambda}_0$	Activation Probability ρ
A_1	60000	0.01	0.9
A_2	60000	0.015	0.9
A_3	60000	0.02	0.9
A_4	60000	0.025	0.9
B_5	60000	0.02	0.8
B_6	60000	0.02	0.85
B_7	60000	0.02	0.9
B_8	60000	0.02	0.95
C_9	60000	0.02	0.9
C_{10}	90000	0.02	0.9
C_{11}	120000	0.02	0.9

Table 9.2: MAD-I: Implanted Episodes vs. Base Firing Rate $\hat{\lambda}_0$

Episode Types \ $\hat{\lambda}_0$	0.01	0.015	0.02	0.025
Serial Episode	0.999	0.998	0.997	0.996
Parallel Episode	0.999	0.999	0.998	0.996
Fixed-delay Episode	1.0	1.0	0.999	0.998

Table 9.3: MAD-II: Implanted Episodes vs. Base Firing Rate $\hat{\lambda}_0$

Episode Types \ $\hat{\lambda}_0$	0.01	0.015	0.02	0.025
Serial Episode	1.0	0.999	0.998	0.997
Parallel Episode	1.0	0.999	0.999	0.998
Fixed-delay Episode	1.0	1.0	0.999	0.998

Table 9.4: PAT: Implanted Episodes vs. Base Firing Rate $\hat{\lambda}_0$

Episode Types \ $\hat{\lambda}_0$	0.01	0.015	0.02	0.025
Serial Episode	0.0	0.0	0.0	0.0
Parallel Episode	0.0	0.0	0.0	0.0
Fixed-delay Episode	1.0	1.0	1.0	1.0

Table 9.5: MAD-I: Implanted Episodes vs. Activation Probability ρ

Episode Types \ ρ	0.8	0.85	0.9	0.95
Serial Episode	0.997	0.996	0.997	0.997
Parallel Episode	0.998	0.998	0.998	0.997
Fixed-delay Episode	1.0	1.0	0.999	0.999

Table 9.6: MAD-II: Implanted Episodes vs. Activation Probability ρ

Episode Types \ ρ	0.8	0.85	0.9	0.95
Serial Episode	0.999	0.999	0.998	0.998
Parallel Episode	1.0	1.0	0.999	0.998
Fixed-delay Episode	1.0	0.999	1.0	0.999

Table 9.7: PAT: Implanted Episodes vs. Activation Probability ρ

Episode Types \ ρ	0.8	0.85	0.9	0.95
Serial Episode	0.0	0.0	0.0	0.0
Parallel Episode	0.0	0.0	0.0	0.0
Fixed-delay Episode	1.0	1.0	1.0	1.0

Table 9.8: MAD-I: Implanted Episodes vs. Data Set Length L

Episode Types \ L	60000	90000	120000
Serial Episode	0.998	0.998	0.997
Parallel Episode	0.999	0.998	0.999
Fixed-delay Episode	0.999	0.999	0.999

Table 9.9: MAD-II: Implanted Episodes vs. Data Set Length L

Episode Types \ L	60000	90000	120000
Serial Episode	0.999	0.998	0.998
Parallel Episode	0.999	0.999	0.999
Fixed-delay Episode	1.0	1.0	0.999

Finally, we evaluate the performances of these methods as the data set length was varied. Table 9.8 presents the result of MAD-I, Table 9.9 presents the result of MAD-II. The experimental results show that our methods worked constantly well by achieving high scores in all of cases, indicating that the performances of our methods are not affected by the data set length. The result from Table 9.8 shows that Patnaik's method still failed to detect either serial or parallel episodes.

Table 9.10: PAT: Implanted Episodes vs. Data Set Length L

Episode Types \ L	60000	90000	120000
Serial Episode	0.0	0.0	0.0
Parallel Episode	0.0	0.0	0.0
Fixed-delay Episode	1.0	1.0	1.0

Chapter 10

Conclusion

10.1 Summary

In this thesis, we studied the problem of extracting temporal associations of events from multiple event sequences. We presented a two-phase method, called *Multivariate Association Discovery* (MAD). In the first phase, we discover bivariate associations from two event sequences by comparing the observed distribution of the forward distances of their event occurrences with a theoretically derived null distribution. A bivariate association is retrieved if there exists in the observed distribution a region with a statistically significant higher count of forward distances than expected. Two approaches are proposed in this thesis to search for a statistically significant region. In the first approach, we estimate the observed distribution by using the histogram of forward distances and applying an effective binning technique to learn a proper bin size for this histogram. A statistically significant region is identified if there exists a bin whose frequency is statistically significant assuming the expected null distribution. In the second approach, given that the temporal distance between two associated events follows a Gaussian distribution, we estimate the observed distribution by using a state-of-the-art kernel density estimation technique. A theoretical distribution function is derived from the analysis of individual distributions of three categories of temporal distances generated by events of distinct sequences. We treat bivariate association discovery as a least squares curve-fitting problem, where we adjust the parameters of the Gaussian components in the theoretical function to optimally fit the curve of the observed distribution.

A statistically significant region is found if there is a bell-shaped portion in the observed distribution showing a statistically significant deviation from the expected null distribution. In the second phase, we use a bivariate association graph to search for multivariate associations with the requirement that their frequencies should also be significant in a statistical sense.

To validate our method, we applied it to two different application domains. Firstly, we used MAD to detect multivariate motifs from multivariate time series data. Existing methods of multivariate motif discovery are all limited by assuming explicitly or implicitly that the univariate elements of a multivariate motif occur completely or approximately synchronously. This assumption does not hold in many real-world applications. We empirically compared MAD with the currently most effective related work on both synthetic and real-world data sets. The experimental results indicate that our method can not only discover synchronous motifs as the other method does, but also successfully find non-synchronous multivariate motifs. Secondly, we applied our method to detect frequent episodes from event streams. An episode can be understood as a temporally partially ordered set of event types. Current methods on frequent episode discovery are all limited by requiring users to either provide possible lengths of frequent episodes or specify an inter-event time constraint for every pair of successive event types in an episode, which results in poor performance when users have little knowledge about the data. We compared MAD with the most recent work on frequent episode discovery by using simulated spike train data. The empirical results show that our method can effectively detect episodes with variable lengths.

10.2 Future Work

Since there exist a substantial number of real applications where individual event sequences do not follow a Poisson process, we are unable to use the properties of a Poisson process to derive an expected null distribution of forward distances. In the future research, we will investigate a general method that can detect bivariate associations when event sequences do not follow Poisson processes. In addition,

we can work on reducing the runtime of our method and further evaluating MAD on real-world spike train data sets.

Bibliography

[1] M. Abeles and I. Gat. Detecting precise firing patterns in experimental data. *Journal of Neurosci Methods*, 107:141–154, 2001.

[2] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth dover printing, tenth gpo printing edition, 1964.

[3] Avinash Achar, Srivatsan Laxman, V. Raajay, and P. S. Sastry. Discovering general partial orders in event streams. *CoRR*, abs/0902.1227, 2009.

[4] Avinash Achar, Srivatsan Laxman, and P. S. Sastry. A unified view of automata-based algorithms for frequent episode discovery. *CoRR*, abs/1007.0690, 2010.

[5] Adrian Baddeley. Spatial point processes and their applications. 1892:1–75, 2007.

[6] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38(5):2916–2957, 2010.

[7] Bouchra Bouqata, Christopher D. Carothers, Boleslaw K. Szymanski, and Mohammed Javeed Zaki. Vogue: A novel variable order-gap state machine for modeling sequences. In *Proceedings of the 10th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–54, 2006.

[8] Jeremy Buhler and Martin Tompa. Finding motifs using random projections. In *Proceedings of Research in Computational Molecular Biology - 5th Annual International Conference*, pages 69–76, 2001.

[9] Gemma Casas-Garriga. Discovering unbounded episodes in sequential data. In *Proceedings of the 7th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 83–94, 2003.

[10] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 493–498, 2003.

[11] F. Gabbiani and C. Koch. Principles of spike train analysis. *Methods in Neuronal Modeling*, pages 313–360, 2005.

[12] Robert Gwadera, Mikhail J Atallah, and Wojciech Szpankowski. Reliable detection of episodes in event sequences. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 67–74, 2003.

[13] Srivatsan Laxman, P. S. Sastry, and K. P. Unnikrishnan. Discovering frequent generalized episodes when events persist for different durations. *IEEE Transactions on Knowledge Data Engineering*, 19(9):1188–1201, 2007.

[14] Srivatsan Laxman, P. S. Sastry, and K. P. Unnikrishnan. A fast algorithm for finding frequent episodes in event streams. In *Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 410–419, 2007.

[15] Srivatsan Laxman, Basel Shadid, P. S. Sastry, and K. P. Unnikrishnan. Temporal data mining for root-cause analysis of machine faults in automotive assembly lines. *CoRR*, abs/0904.4608, 2009.

[16] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 2–11, 2003.

[17] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovering frequent episodes in sequences. In *Proceedings of the 1st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 210–215, 1995.

[18] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.

[19] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[20] Ronald Meester. *A Natural Introduction to Probability Theory*. Birkhauser Verlag, new edition, 2004.

[21] David Minnen, Charles Isbell, Irfan Essa, and Thad Starner. Detecting subdimensional motifs: an efficient algorithm for generalized multivariate pattern discovery. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 601–606, 2007.

[22] David Minnen, Charles L. Isbell, Irfan Essa, and Thad Starner. Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. In *Proceedings of the 21th Conference on Artificial Intelligence*, pages 615–620, 2007.

[23] David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. Discovering characteristic actions from on-body sensor data. In *Proceedings of the 5th International Semantic Web Conference*, pages 11–18, 2006.

[24] David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. Improving activity discovery with automatic neighborhood estimation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2814–2819, 2007.

[25] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, and Sydney Cash. Exact discovery of time series motifs. In *Proceedings of the 9th SIAM Conference on Data Mining*, pages 473–484, 2009.

[26] Anny Ng and Ada Wai-Chee Fu. Mining frequent episodes for relating financial events and stock trends. In *Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 27–39, 2003.

[27] Debprakash Patnaik, Srivatsan Laxman, and Naren Ramakrishnan. Discovering excitatory networks from discrete event streams with applications to neuronal spike train analysis. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 407–416, 2009.

[28] V. Raajay. Frequent episode mining and multi-neuronal spike train data analysis. *Master's thesis, IISc, Bangalore*, 2009.

[29] P. S. Sastry and K. P. Unnikrishnan. Conditional probability-based significance tests for sequential patterns in multineuronal spike trains. *Neural Computation*, 22(4):1025–1059, 2010.

[30] J. P. Shaffer. Multiple hypothesis testing. *The Annual Review of Psychology*, 46:561–584, 1995.

[31] Hideaki Shimazaki and Shigeru Shinomoto. A method for selecting the bin size of a time histogram. *Neural Computation*, 19(6):1503–1527, 2007.

[32] H.A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21:65–66, 1926.

[33] Yoshiki Tanaka, Kazuhisa Iwamoto, and Kuniaki Uehara. Discovery of time-series motif from multi-dimensional data based on mdl principle. *Machine Learning*, 58(2-3):269–300, 2005.

[34] Alireza Vahdatpour, Navid Amini, and Majid Sarrafzadeh. Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *Proceedings of the 21th International Joint Conference on Artificial Intelligence*, pages 1261–1266, 2009.

[35] Alireza Vahdatpour and Majid Sarrafzadeh. Unsupervised discovery of abnormal activity occurrences in multi-dimensional time series, with applications in wearable systems. In *Proceedings of the 10th SIAM Conference on Data Mining*, pages 641–652, 2010.

[36] Lei Wang, Eng Siong Chng, and Haizhou Li. A tree-construction search approach for multivariate time series motifs discovery. *Pattern Recognition Letters*, 31:869C875, 2010.

[37] Winston Wu, Lawrence Au, Brett Jordan, Thanos Stathopoulos, Maxim Batalin, William Kaiser, Alireza Vahdatpour, Majid Sarrafzadeh, Meika Fang, and Joshua Chodosh. The smartcane system: an assistive device for geriatrics. In *Proceedings of the 7th International Conference on Body Area Networks*, pages 1–4, 2008.