Chapter 3
Temporal Data Mining: Similarity-Profiled Association Pattern

Jin Soung Yoo

Department of Computer Science, Indiana University-Purdue University, Fort Wayne, Indiana, USA
yooj@ipfw.edu

Abstract. Temporal data are of increasing importance in a variety of fields, such as financial data forecasting, Internet site usage monitoring, biomedicine, geographical data processing and scientific observation. Temporal data mining deals with the discovery of useful information from a large amount of temporal data. Over the last decade many interesting techniques of temporal data mining were proposed and shown to be useful in many applications. In this article, we present a temporal association mining problem based on a similarity constraint. Given a temporal transaction database and a user-defined reference sequence of interest over time, similarity-profiled temporal association mining is to discover all associated itemsets whose prevalence variations over time are similar to the reference sequence. The temporal association patterns can reveal interesting association relationships of data items which co-occur with a particular event over time. Most works in temporal association mining have focused on capturing special temporal regulation patterns such as a cyclic pattern and a calendar scheme-based pattern. However, the similarity-based temporal model is flexible in representing interesting temporal association patterns using a user-defined reference sequence. This article presents the problem formulation of similarity-profiled temporal association mining, the design concept of the mining algorithm, and the experimental result.

1 Introduction

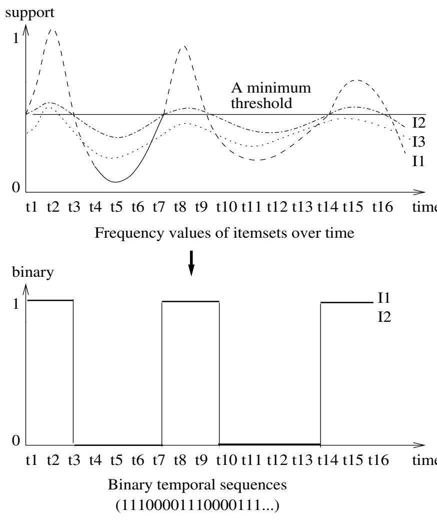
Recent advances in data collection and storage technology have made it possible to collect vast amounts of data every day in many areas of business and science. *Data mining* is concerned with analyzing large volumes of data to automatically discover interesting regularities or relationships which in turn lead to better understanding of the underlying processes. The field of *temporal data mining* deals with such analysis in the case of ordered data with temporal interdependencies [33,25]. Over the last decades many interesting techniques of temporal data analysis were proposed and shown to be useful in many applications. For example, the most common type of temporal data is time series data, which consist of real values sampled at regular time intervals. Time series analysis has quite a long history. Weather forecasting, financial or stock market prediction and automatic process control have been of the oldest and most studied applications of time series analysis [12]. Temporal data mining is of a more recent origin with somewhat different objectives. One of major differences between temporal data mining and classical time series analysis lies in the kind of information what we want to estimate or unearth from the data. The scope of temporal data mining extends beyond the standard forecast applications of time series analysis. Very often, in data mining applications, one may be interested in knowing which variables in the data are expected to exhibit any correlations or causal relationships over time. For example, a timestamped list of items bought by customers lends itself to data mining analysis that could reveal which combinations of items tend to be frequently consumed together, and whether they tend to show particular behaviors over time.

Temporal data mining tasks can be grouped as follows: (i) prediction, (ii) classification, (iii) clustering, (iv) search & retrieval and (v) pattern discovery [33]. Of the five categories listed above, algorithms for pattern discovery in large temporal databases, however, are of more recent origin and are mostly discussed in data mining literature. Temporal pattern mining deals with the discovery of *temporal patterns of interest* in temporal data, where the interest is determined by the domain and the application. The diversity of applications has led to the development of many temporal pattern models. The three popular frameworks of temporal pattern discovery are *sequence mining(or frequent sequence pattern discovery)*, *frequent episode discovery* and *temporal association rule discovery* [33]. Association rule mining is concerned with the discovery of inter-relationships among various data items in transactional data [5]. An example of association rule is

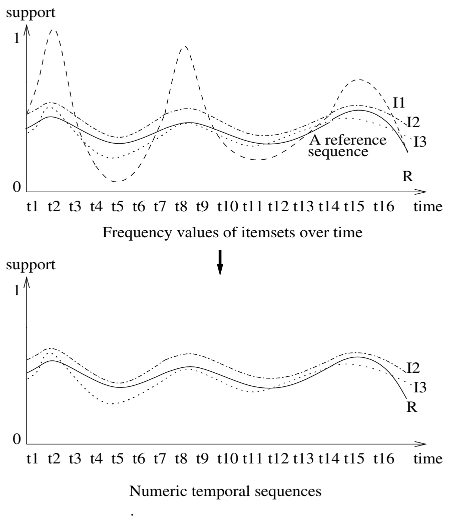
$$\textit{wine} \implies \textit{cheese} \text{ (support=10\%, confidence =80\%).}$$

This rule says that 10% of customers buy *wine* and *cheese* together, and those who buy *wine* also buy *cheese* 80% of the time. The process of association rule mining is to find all *frequent* itemsets that exceed a user-specified support threshold, and then generate association rules from the frequent itemsets which satisfies a user-given confidence threshold. Following the work of [5], the discovery of association rules has been extensively studied in [23,22,36,38]. In particular, [35,28,37] have paid attention to temporal information which is implicitly related to transaction data, and proposed periodic temporal association mining problems.

Temporal association pattern mining is an important extension of association pattern mining as it can be used to mine the behavior aspects of data over time rather than just states at a point in time. Ozden et al. [35] introduced the idea of cyclic association rules which can discover periodicity time information of data. An association rule is said to be cyclic if it holds with a fixed periodicity along the entire length of the sequence of time intervals. For example, a periodic association pattern may state that wine and cheese are sold together primarily in the weekends. A temporal association pattern can be explained with a binary sequence where the time axis is broken down into equally spaced user-defined



(a) Temporal regulation patterns



(b) Temporal similarity patterns

Fig. 1. A comparison of temporal association patterns

time intervals and association rules that hold for the transactions in each of these time intervals are considered. In the binary sequence, 1's correspond to the time intervals in which the association pattern is valid (i.e., the rule satisfies minimum support threshold and confidence threshold.), and the 0's correspond to the time intervals in which it is not valid. For instance, when a defined time interval is day, '10000001000000...' represents the binary sequence of a repetitive temporal pattern on Monday. Fig 1 (a) illustrates an example of periodic temporal association patterns. It shows the support values of three itemsets I_1, I_2 and I_3 over time, and the binary temporal sequences of I_1 and I_2 under a fixed frequent threshold (e.g., support threshold=0.5).

There are many ways of defining what constitutes a pattern. Fig 1 (a) says that I_2 shows the same temporal pattern with I_1 (i.e., a periodic pattern with the binary sequence, 111000111000...), even if their actual measure strengths are quite different. In contrast, Fig 1 (b) illustrates temporal association patterns based on similarity. It shows that I_2 and I_3 have very similar behaviors over time. As a guidance to find similar itemsets, this model uses a reference sequence. The reference sequence can represent the change of interest measure values of a specific event (item) (e.g., a specific product in market basket data, a stock exchange in the stock market, a climate variable such as temperature or precipitation, and a scientific phenomenon), or a user guided sequence pattern showing special shapes (e.g., a seasonal, emerging or diminishing pattern over time). Current periodic temporal association mining methods cannot reveal this kind of temporal association pattern. In this article, we present a temporal association pattern mining problem based on a similarity constraint [42].

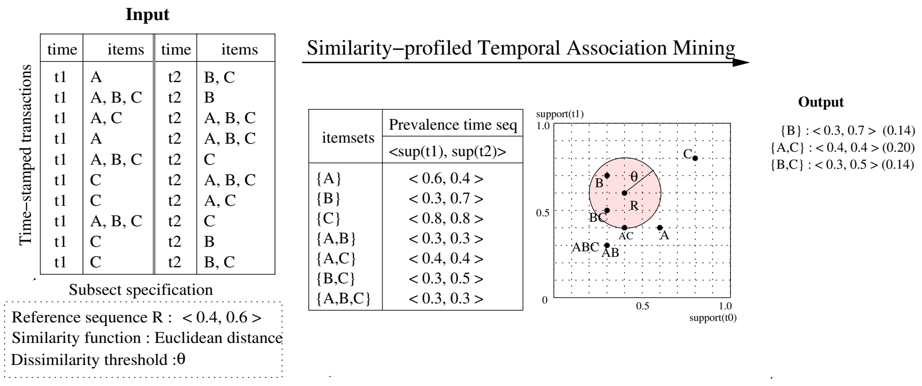
The remainder of the article is organized as follows. Section 2 presents the problem statement of similarity-profiled temporal association mining. The design concept of the mining algorithm is described in Section 3. Section 4 shows the experimental result. The related work is described in Section 5. Section 6 discusses some future direction of the work.

2 Similarity-Profiled Temporal Association Pattern

Given a timestamped transaction database and a user-defined reference sequence of interest over time, *similarity-profiled temporal association mining* is to discover all associated itemsets whose prevalence(frequency) variations over time are similar to the reference sequence under a threshold. Similarity-profiled temporal association mining can reveal interesting relationships of data items which co-occur with a particular event over time. For example, weather-to-sales relationship is a very interesting problem in retail analysts [2,4]. Wal-Mart discovered a surprising customer buying pattern during hurricane season in a region. Not only did survival kits (e.g., flashlights, generators, tarps) show similar selling patterns with bottled water (which is one of important items in emergency), but so did the sales of Strawberry Pop-Tarts which is an unexpected snack item [4]. The similarity-profiled temporal association mining may help finding such item sets whose sales similarly change to that of a specific item(event) for a period of time. The mining results can improve supply chain planning, and retail decision-making for maximizing the visibility of items most likely to be in high demand during a special time period. As another example in business domain, consider an online web site. Weather.com offers weather-related lifestyle information including travel, driving, home & garden and sporting events as well as weather itself information [3]. According to the web site's report, almost 40% of weather.com visitors shop home improvement products with increase of temperature [3]. The web site may attract more advertisers if it can analyze the relationships of visited web sites through weather.com with changes of weather. Consider a scientific application domain. Earth scientists have been interested in the behavior of climates in a region which are often influenced with the El Niño phenomenon, an abnormal warming in the eastern tropical Pacific Ocean [34]. If we consider the El Niño related index values over last 10 years, e.g., the Southern Oscillation Index(SOI) [34], as a reference sequence, one example of similarity-profiled temporal association might be a climate event pattern of low precipitation and low atmospheric carbon dioxide in Australia whose co-occurrence over time is similar to the fluctuation of the El Niño index sequence.

2.1 Problem Statement

The formal problem statement of similarity-profiled temporal association mining follows below. Fig. 2 shows a simple illustration of the pattern mining.



time	items	time	items
t1	A	t2	B, C
t1	A, B, C	t2	B
t1	A, C	t2	A, B, C
t1	A	t2	A, B, C
t1	A, B, C	t2	C
t1	C	t2	A, B, C
t1	C	t2	A, C
t1	A, B, C	t2	C
t1	C	t2	B
t1	C	t2	B, C

Reference sequence R : < 0.4, 0.6 >
Similarity function : Euclidean distance
Dissimilarity threshold :θ

itemsets	Prevalence time seq
	<sup(t1), sup(t2)>
{A}	< 0.6, 0.4 >
{B}	< 0.3, 0.7 >
{C}	< 0.8, 0.8 >
{A,B}	< 0.3, 0.3 >
{A,C}	< 0.4, 0.4 >
{B,C}	< 0.3, 0.5 >
{A,B,C}	< 0.3, 0.3 >

{B} : < 0.3, 0.7 > (0.14)
{A,C} : < 0.4, 0.4 > (0.20)
{B,C} : < 0.3, 0.5 > (0.14)

(a) (b) (c)

Fig. 2. An example of similarity-profiled temporal association mining (a) Input data (b) Generated support time sequences, and sequence search (c) Output itemsets

Given
1) A finite set of items $\mathcal{I}$
2) An interest time period $\mathcal{T}=t_1 \cup \ldots \cup t_n$, where t_i is a time slot by a time granularity, $t_i \cap t_j = \emptyset, i \neq j$
3) A timestamped transaction database $\mathcal{D}=\mathcal{D}_1 \cup \ldots \cup \mathcal{D}_n$, $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset, i \neq j$. Each transaction $d \in \mathcal{D}$ is a tuple $<$ *timestamp*, *itemset* $>$ where *timestamp* is a time $\in \mathcal{T}$ that the transaction is executed, and *itemset* $\subseteq \mathcal{I}$. $\mathcal{D}_i$ is a set of transactions included in time slot t_i.
4) A subset specification
  4a) A reference sequence $\boldsymbol{R} =< r_1, \ldots, r_n >$ over time slots $t_1, \ldots, t_n$
  4b) A similarity function $f_{Similarity}(\boldsymbol{X}, \boldsymbol{Y}) \mapsto \mathbb{R}^n$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are numeric sequences.
  4c) A dissimilarity threshold θ
Find A set of itemsets $I \subseteq \mathcal{I}$ which satisfy the given subset specification, i.e., $f_{Similarity}(\boldsymbol{S_I}, \boldsymbol{R}) \leq \theta$, where $\boldsymbol{S_I} =< s_1, \ldots, s_n >$ is the sequence of support values of an itemset I over time slots $t_1, \ldots, t_n$.

Items: We use the standard notion of *items* in traditional association rule mining [5]. Items can be supermarket items purchased by a customer during a shopping visit, product pages viewed in a web session, climate events at a location, stocks exchanged within a hour, etc. Items can be grouped to form an *itemset*. An itemset with k distinct items is referred to as a k *itemset*. The size of the itemset space is $2^{|\mathcal{I}|} - 1$, where $|\mathcal{I}|$ is the number of items.

Time period: A time period can be a particular year or any arbitrary period of time. We model time as discrete, and thus, a total time period can be viewed as a sequence of time slots by a certain time granularity [10]. For example, one year period can be divided into monthly unit time slots. The i^{th} time slot is denoted with t_i.

Transaction database: The database $\mathcal{D}$ is a set of timestamped transactions. Each transaction is a set of items over a finite item domain, and has a time point when the transaction is executed. The time point associated with a transaction is called its *timestamp*. The transaction dataset can be partitioned to disjoint groups of transactions by a time granularity. $\mathcal{D}_i$ represents a part of transactions of $\mathcal{D}$ executed in time slot t_i.

Subset specification: A subset specification can be used to represent a set of conditions that itemsets have to satisfy to become interesting patterns. Our subset specification consists of three components: a reference sequence, a similarity function, and a dissimilarity threshold. First, we assume that an arbitrary temporal pattern of interest can be defined as a reference sequence by the user. A reference time sequence is a sequence of interesting values over time slots $t_1, \ldots, t_n$. In the example of Fig. 2, <0.4, 0.6> is given as the reference sequence $\boldsymbol{R}$. Second, we use a distance-based similarity function, a $\mathcal{L}_p$ norm ($p = 1, 2, \ldots, \infty$). Many similarity measures have been discussed in time series database literature [20,18,8,27,16]. A $\mathcal{L}_p$ norm is the most popularly used distance measure in similar time sequence search [41,18,6,26], and can be used as basic building blocks for more complex similarity models as in [7]. When p=1, the $\mathcal{L}_1$ norm is known as a *city-block* or *Manhattan*. When p=2, the $\mathcal{L}_2$ norm is called a *Euclidean distance*, and defined as $\mathcal{L}_2(\boldsymbol{X}, \boldsymbol{Y}) = (\sum_{t=1}^{n} |x_t - y_t|^2)^{\frac{1}{2}}$. We also consider a *normalized Euclidean distance*, $Normalized\_\mathcal{L}_2(\boldsymbol{X}, \boldsymbol{Y}) = (\frac{1}{n})^{\frac{1}{2}} * \mathcal{L}_2(\boldsymbol{X}, \boldsymbol{Y}) = (\frac{\sum_{t=1}^{n} |x_t - y_t|^2}{n})^{\frac{1}{2}}$, where $\boldsymbol{X} = < x_1, \ldots, x_n >$ and $\boldsymbol{Y} = < y_1, \ldots, y_n >$ are time sequences, and n is the number of time slots. Euclidean distance is used for figure examples in this article. The last component of the subset specification is a dissimilarity threshold. It indicates a maximum discrepancy to allow for similarity-profiled temporal association patterns.

2.2 Interest Measure

The similarity-profiled temporal association pattern uses a composite interest measure which describes a discrepancy degree between the reference sequence and the sequence of frequency values of an itemset over time. A support time sequence is used to represent temporally changed frequency values of an itemset.

Definition 1. *Let $\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_n$ be a disjoint timestamped transaction dataset. The* **support time sequence** *of an itemset I is defined as*

$$\boldsymbol{S_I} =< support(I, \mathcal{D}_1), \ldots, support(I, \mathcal{D}_n) >,$$

where $support(I, \mathcal{D}_t)$ is the support value of itemset I in a transaction set $\mathcal{D}_t$, which is the fraction of transactions that contain the itemset I in $\mathcal{D}_t$ such that $support(I, \mathcal{D}_t) = |\{d \in \mathcal{D}_t | I \subseteq d\}| / |\mathcal{D}_t|$, $1 \leq t \leq n$.

We assume the reference sequence values are in the same scale with the support measure, or can be transformed to the same scale. The interest measure of the similarity-profiled temporal association is defined as following.

Definition 2. *Let I be an itemset and $\boldsymbol{S_I} = < s_1, \ldots, s_n >$ be the support time sequence of I. Given a reference sequence $\boldsymbol{R} = < r_1, \ldots, r_n >$, an interest measure for the similarity-profiled temporal association pattern is defined as $D(\boldsymbol{R}, \boldsymbol{S_I})$ which is a $\mathcal{L}_p$ norm ($p = 1, 2, \ldots, \infty$) based dissimilarity distance between $\boldsymbol{R}$ and $\boldsymbol{S_I}$.*

An itemset I is called a *similar itemset* if $D(\boldsymbol{R}, \boldsymbol{S_I}) \leq \theta$ where θ is a dissimilarity threshold. In Fig. 2, the pattern mining output is {B}, {A, C} and {B, C} since their interest measure values do not exceed the dissimilarity threshold, 0.2.

3 Mining Algorithm

Similarity-profiled temporal association mining presents challenges in computation. The straight-forward approach is to divide the mining process into two separate phrases. The first phrase computes the support values of all possible itemsets at each time point, and generates their support sequences. The second phrase compares the generated support time sequences with a given reference sequence, and finds similar itemsets. In this step, a multi-dimensional access method such as an R-tree family can be used for a fast sequence search [21,18,26,14]. However, the computational costs of first generating the support time sequences of all combinatorial candidate itemsets and then doing the similarity search become prohibitively expensive with increase of items. Thus it is crucial to devise schemes to reduce the itemset search space effectively for efficient computation. We first present the design concept of similarity-profiled temporal association mining algorithm.

3.1 Envelope of Support Time Sequence

It is a core operation to generate the support time sequences of itemsets in a similarity-profiled association mining algorithm. The operation, however, is very data intensive, and sometimes can produce the sequences of all combinations of items. We explore a way for estimating support time sequences without examining the transaction data. Calders in [13] proposed a set of rules for deducing best bounds on the support of an itemset if the supports of all subsets of it are known.

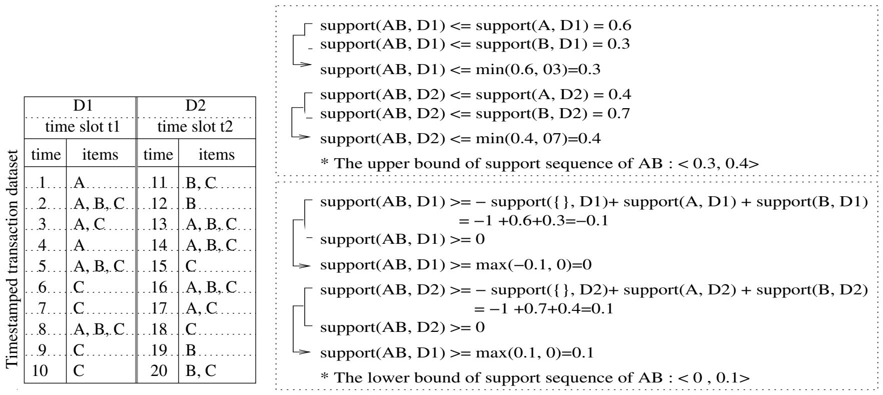
Theorem 1. *Let $\mathcal{D}$ be a transaction dataset, and I be an itemset,*

$$support(I, \mathcal{D}) \in [L(I, \mathcal{D}), U(I, \mathcal{D})]$$

with

$$L(I,\mathcal{D}) = max\{\sigma_I(J,\mathcal{D}), 0 \mid J \subset I \text{ and } |J| \text{ is even }\},$$
$$U(I,\mathcal{D}) = min\{\sigma_I(J,\mathcal{D}) \mid J \subset I \text{ and } |J| \text{ is odd }\}$$

where $\sigma_I(J, \mathcal{D}) = \sum_{J \subseteq J' \subset I} (-1)^{|I-J'|+1} \cdot support(J', \mathcal{D}).$



Timestamped transaction dataset

D1		D2	
time slot t1		time slot t2	
time	items	time	items
1	A	11	B, C
2	A, B, C	12	B
3	A, C	13	A, B, C
4	A	14	A, B, C
5	A, B, C	15	C
6	C	16	A, B, C
7	C	17	A, C
8	A, B, C	18	C
9	C	19	B
10	C	20	B, C

support(AB, D1) <= support(A, D1) = 0.6
support(AB, D1) <= support(B, D1) = 0.3
support(AB, D1) <= min(0.6, 03)=0.3

support(AB, D2) <= support(A, D2) = 0.4
support(AB, D2) <= support(B, D2) = 0.7
support(AB, D2) <= min(0.4, 07)=0.4

* The upper bound of support sequence of AB : < 0.3, 0.4>

support(AB, D1) >= – support({ }, D1)+ support(A, D1) + support(B, D1)
= –1 +0.6+0.3=–0.1
support(AB, D1) >= 0
support(AB, D1) >= max(–0.1, 0)=0

support(AB, D2) >= – support({ }, D2)+ support(A, D2) + support(B, D2)
= –1 +0.7+0.4=0.1
support(AB, D2) >= 0
support(AB, D1) >= max(0.1, 0)=0.1

* The lower bound of support sequence of AB : < 0 , 0.1>

(a) An example data

(b) Bounds of supports

Fig. 3. An example of upper and lower bounds of support sequence of itemset AB

$L(I, \mathcal{D})$ means a lower bound of $support(I, \mathcal{D})$, and $U(I, \mathcal{D})$ means an upper bound of $support(I, \mathcal{D})$. The proof of the tight bounds is described in [13]. We adopt this set of rules to derive the tight upper bound and lower bound of support time sequence of an itemset.

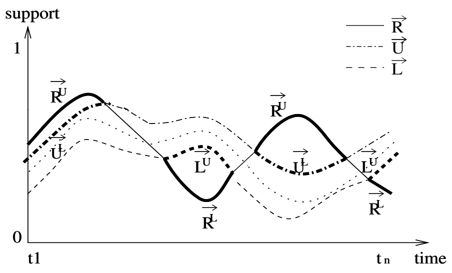
Definition 3. *Let $\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_n$ be a timestamped transaction dataset. The* **lower bound support time sequence** *of an itemset I, $\boldsymbol{L_I}$, and the* **upper bound support time sequence** *of I, $\boldsymbol{U_I}$ are defined as following.*

$$\boldsymbol{L_I} =< l_1, \ldots, l_n >=< L(I, D_1), \ldots, L(I, D_n) >$$
$$\boldsymbol{U_I} =< u_1, \ldots, u_n >=< U(I, D_1), \ldots, U(I, D_n) >$$

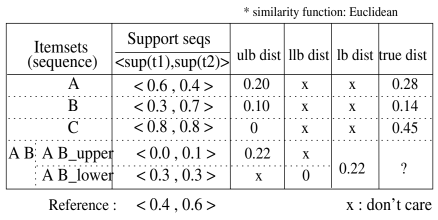
Fig. 3 shows the computation of the lower and upper bound support sequences of an itemset $I = \{A, B\}$.

3.2 Lower Bounding Distance

A lower bounding distance concept is used to find itemsets whose support sequences could not possibly match with a reference sequence under a given threshold. If the lower bounding distance of an itemset does not satisfy the dissimilarity threshold, its true distance also does not satisfy the threshold. Thus the lower bounding distance can be used to prune the itemset without computing its true distance. Our lower bounding distance is defined with upper and lower bound support time sequences. It consists of two parts, *upper lower-bounding distance* and *lower lower-bounding distance.*



(a) Subsequences for a lower bounding distance



* similarity function: Euclidean

Itemsets (sequence)		Support seqs <sup(t1),sup(t2)>	ulb dist	llb dist	lb dist	true dist
A		< 0.6 , 0.4 >	0.20	x	x	0.28
B		< 0.3 , 0.7 >	0.10	x	x	0.14
C		< 0.8 , 0.8 >	0	x	x	0.45
A B	A B_upper	< 0.0 , 0.1 >	0.22	x	0.22	?
	A B_lower	< 0.3 , 0.3 >	x	0		

Reference : < 0.4 , 0.6 >   x : don't care

(b) Lower bounding distance of AB

Fig. 4. An example of lower bounding distances

Definition 4. *For a reference sequence* $\boldsymbol{R}$ *and the upper bound support sequence* $\boldsymbol{U}$ *of an itemset, let* $\boldsymbol{R^U} =< r_1, \ldots, r_k >$ *be a subsequence of* $\boldsymbol{R}$*, and* $\boldsymbol{U^L} =< u_1, \ldots, u_k >$ *be a subsequence of* $\boldsymbol{U}$ *where* $r_t > u_t$*,* $1 \leq t \leq k$*. The* **upper lower-bounding distance** *between* $\boldsymbol{R}$ *and* $\boldsymbol{U}$*,* $D_{Ulb}(\boldsymbol{R}, \boldsymbol{U})$*, is defined as* $D(\boldsymbol{R^U}, \boldsymbol{U^L})$*.*

The upper lower-bounding distance between $\boldsymbol{R}$ and $\boldsymbol{U}$ is a dissimilarity distance between a subsequence of $\boldsymbol{R}$, $\boldsymbol{R^U}$, and a subsequence of $\boldsymbol{U}$, $\boldsymbol{U^L}$, in which each element value r_t in $\boldsymbol{R^U}$ is greater than the corresponding element value u_t of $\boldsymbol{U^L}$. For example, when Euclidean distance is the similarity function, $D_{Ulb}(\boldsymbol{R}, \boldsymbol{U}) = D(\boldsymbol{R^U}, \boldsymbol{U^L}) = (\sum_{t=1}^{n} f(r_t, u_t)))^{\frac{1}{2}}$, where if $r_t > u_t$, $f(r_t, u_t) = |r_t - u_t|^2$; otherwise, $f(r_t, u_t) = 0$.

Definition 5. *For a reference sequence* $\boldsymbol{R}$ *and the lower bound support time sequence* $\boldsymbol{L}$ *of an itemset, let* $\boldsymbol{R^L} =< r_1, \ldots, r_k >$ *be a subsequence of* $\boldsymbol{R}$*, and* $\boldsymbol{L^U} =< l_1, \ldots, l_k >$ *be a subsequence of* $\boldsymbol{L}$ *where* $r_t < l_t$ *,* $1 \leq t \leq k$*. The* **lower lower-bounding distance** *between* $\boldsymbol{R}$ *and* $\boldsymbol{L}$*,* $D_{Llb}(\boldsymbol{R}, \boldsymbol{L})$*, is defined as* $D(\boldsymbol{R^L}, \boldsymbol{L^U})$*.*

The lower lower-bounding distance between a reference sequence $\boldsymbol{R}$ and a lower bound support sequence $\boldsymbol{L}$ is a dissimilarity distance between $\boldsymbol{R^L}$ and $\boldsymbol{L^U}$, in which each element value r_t in $\boldsymbol{R^L}$ are less than the corresponding element value l_t of $\boldsymbol{L^U}$.

Definition 6. *For a reference sequence* $\boldsymbol{R}$*, and the upper bound support time sequence* $\boldsymbol{U}$ *and lower bound support time sequence* $\boldsymbol{L}$ *of an itemset, the* **lower bounding distance***,* $D_{lb}(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{L})$ *is defined as* $D_{Ulb}(\boldsymbol{R}, \boldsymbol{U}) + D_{Llb}(\boldsymbol{R}, \boldsymbol{L})$*.*

Fig. 4 (a) gives an example of subsequences, $\boldsymbol{R^U}$, $\boldsymbol{U^L}$, $\boldsymbol{R^L}$ and $\boldsymbol{L^U}$. As shown, the subsequences do not need to be a continuous sequence. Fig. 4 (b) shows an example of lower bounding distances of {A, B} computed from the upper bound support sequence and lower bound support sequence of {A, B}.

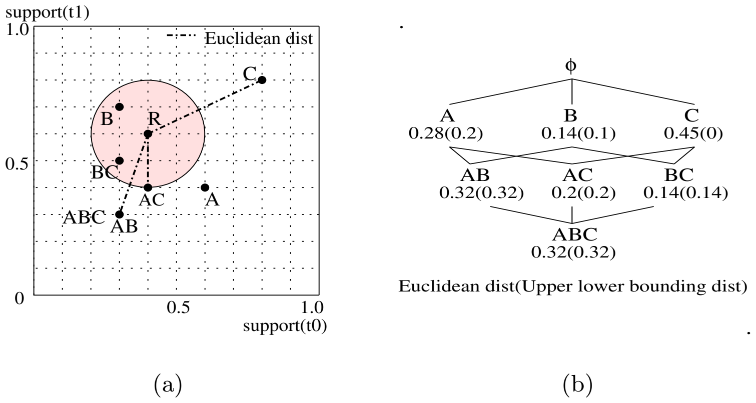


Fig. 5. (a) Non-monotonicity of the Euclidean distance (b) Monotonically non-decreasing property of the upper lower-bounding distance

Lemma 1. *For the upper bound support time sequence* $\boldsymbol{U} =< u_1, \ldots, u_n >$*, lower bound support time sequence* $\boldsymbol{L} =< l_1, \ldots, l_n >$ *and support time sequence* $\boldsymbol{S} =< s_1, \ldots, s_n >$ *of an itemset I, and a reference sequence* $\boldsymbol{R} =< r_1, \ldots, r_n >$*, the lower bounding distance* $D_{lb}(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{L})$ *and the true distance* $D(\boldsymbol{R}, \boldsymbol{S})$ *hold the following inequality:* $D_{lb}(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{L}) \leq D(\boldsymbol{R}, \boldsymbol{S})$.

For the proof, refer to [42]. Thus, if $D_{lb}(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{L})$ of an itemset is greater than a given threshold, the true distance $D(\boldsymbol{R}, \boldsymbol{S})$ should not satisfy the threshold. Therefore, we know that the itemset will not be in the mining result.

3.3 Monotonicity Property of Upper Lower-Bounding Distance

Next, we explore a scheme to further reduce the itemset search space. The most popular technique to reduce itemset search space in association pattern mining is to use the monotonicity property of support measure [5]. The support values of all supersets of a given itemset are not greater than the support value of that itemset. Thus, if an itemset does not satisfy the support threshold, all supersets of the itemset can be pruned. Unfortunately, our $\mathcal{L}_p$ norms-based interest measure does not show any monotonicity with the size of the itemset. For example, Fig. 5 (a) shows the Euclidean distances between the support time sequences of {C}, {A,C} and {A,B,C}, $\boldsymbol{S}_C$, $\boldsymbol{S}_{AC}$ and $\boldsymbol{S}_{ABC}$, and a reference sequence $\boldsymbol{R}$. As can be seen, $D(\boldsymbol{S}_C, \boldsymbol{R})$=0.45, $D(\boldsymbol{S}_{AC}, \boldsymbol{R})$=0.2 and $D(\boldsymbol{S}_{ABC}, \boldsymbol{R})$ =0.32. Thus, $D(\boldsymbol{S}_{ABC}, \boldsymbol{R}) > D(\boldsymbol{S}_{AC}, \boldsymbol{R})$ but $D(\boldsymbol{S}_{AC}, \boldsymbol{R}) < D(\boldsymbol{S}_C, \boldsymbol{R})$. However, we found an interesting property related to our upper lower-bounding distance.

Lemma 2. *The upper lower-bounding distance between the (upper bound) support time sequence of an itemset and a reference time sequence is* **monotonically non-decreasing** *with the size of itemset.*

For the proof, refer to [42]. For example, in Fig. 5 (b), $D_{Ulb}(\boldsymbol{S}_A, \boldsymbol{R})$=0.2, $D_{Ulb}(\boldsymbol{S}_B, \boldsymbol{R})$=0.1 and $D_{Ulb}(\boldsymbol{S}_{AB}, \boldsymbol{R})$=0.32. Thus $D_{Ulb}(\boldsymbol{S}_A, \boldsymbol{R}) \leq D_{Ulb}(\boldsymbol{S}_{AB}, \boldsymbol{R})$ and $D_{Ulb}(\boldsymbol{S}_B, \boldsymbol{R}) \leq D_{Ulb}(\boldsymbol{S}_{AB}, \boldsymbol{R})$. We can also see that $D_{Ulb}(\boldsymbol{S}_{AB}, \boldsymbol{R}) \leq D_{Ulb}(\boldsymbol{S}_{ABC}, \boldsymbol{R})$, $D_{Ulb}(\boldsymbol{S}_{AC}, \boldsymbol{R}) \leq D_{Ulb}(\boldsymbol{S}_{ABC}, \boldsymbol{R})$, and $D_{Ulb}(\boldsymbol{S}_{BC}, \boldsymbol{R}) \leq D_{Ulb}(\boldsymbol{S}_{ABC}, \boldsymbol{R})$.

3.4 SPAMINE Algorithm

The Similarity-Profiled temporal Association MINing mEthod(SPAMINE) is developed based on the previous algorithm design concept. Algorithm 1 shows the pseudocode of the SPAMINE. Fig. 6 provides an illustration of trace of a SPAMINE execution with the example data in Fig. 2 (a).

Generate the support time sequences of single items and find similar items (Steps 1 - 3): All singletons ($k = 1$) become candidate items(C_1). With reading the entire transaction data, the supports of singletons are computed per each time slot and their support time sequences($\boldsymbol{S}_1$) are generated. If the interest measure value of an item (i.e., distance between its support time sequence and a reference sequence) does not exceed a given threshold, the item is added to a result set(R_1). On the fly, if the upper lower-bounding distance of an item satisfies

```
Inputs:
$E$ : A set of single items.
$TD$: A time-stamped transaction database
$\boldsymbol{R}$ : A reference sequence
$D$ : A similarity function
$\theta$ : A dissimilarity threshold
Output: All itemsets whose support sequences are similar to $\boldsymbol{R}$ under $D$
and $\theta$
Variables :
$k$ : Itemset size
$C_k$ : A set of size $k$ candidate itemsets
$\boldsymbol{U}_k$ : A set of upper bound support sequences of size $k$ itemsets
$\boldsymbol{L}_k$ : A set of lower bound support sequences of size $k$ itemsets
$\boldsymbol{S}_k$ : A set of support sequences of size $k$ itemsets
$\boldsymbol{S}$ : A set of support sequences of all subsets of itemsets
$B_k$ : A set of size $k$ itemsets whose upper lower-bounding distance $\leq \theta$
$A_k$ : A result set of size $k$ itemsets whose true distance $\leq \theta$
Main:
1) $C_1 = E$;
2) $\boldsymbol{S}_1$= generate_support_sequences($C_1$, $TD$);
3) ($A_1$, $B_1$)= find_similar_itemsets($C_1$, $\boldsymbol{S}_1$, $\boldsymbol{R}$, $D$, $\theta$);
4) $k = 2$;
5) while (not empty $B_{k-1}$) do
6)   ($C_k$,$\boldsymbol{U}_k$,$\boldsymbol{L}_k$) =generate_candidate_itemsets($B_{k-1}$, $\boldsymbol{S}$);
7)   $C_k$ =prune_candidate_itemsets_by_lbd($C_k$, $\boldsymbol{U}_k$, $\boldsymbol{L}_k$, $\boldsymbol{R}$, $D$, $\theta$);
8)   $\boldsymbol{S}_k$=generate_support_sequences($C_k$, $TD$);
9)   ($A_k$, $B_k$) =find_similar_itemsets($C_k$, $\boldsymbol{S}_k$, $\boldsymbol{R}$, $D$, $\theta$);
10)   $\boldsymbol{S} = \boldsymbol{S} \cup \boldsymbol{S}_k$; $k = k + 1$;
11) end
12) return $\bigcup(A_1, \ldots, A_k)$;
```

Algorithm 1. SPAMINE algorithm

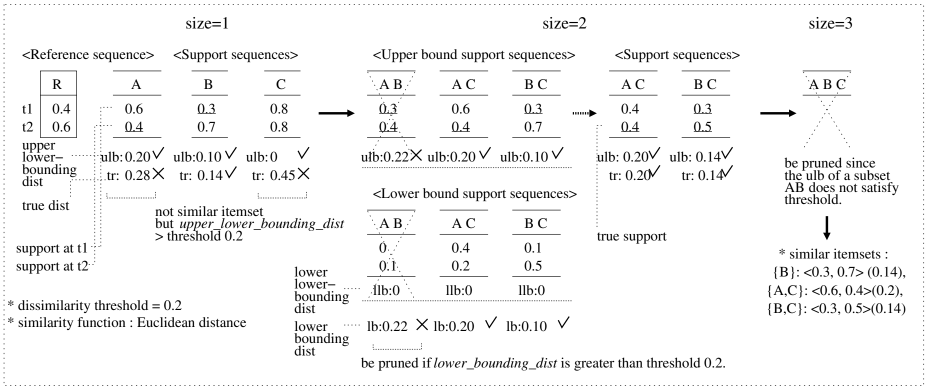

size=1
size=2
size=3
<Reference sequence>
<Support sequences>
<Upper bound support sequences>
<Support sequences>
R
A
B
C
AB
AC
BC
ABC
t1
t2
upper lower–bounding dist
true dist
support at t1
support at t2
ulb:0.20 ✓ ulb:0.10 ✓ ulb:0 ✓
tr: 0.28 ✗ tr: 0.14 ✓ tr: 0.45 ✗
not similar itemset but upper_lower_bounding_dist > threshold 0.2
ulb:0.22 ✗ ulb:0.20 ✓ ulb:0.10 ✓
<Lower bound support sequences>
lower lower–bounding dist
llb:0 llb:0 llb:0
lower bounding dist
lb:0.22 ✗ lb:0.20 ✓ lb:0.10 ✓
be pruned if lower_bounding_dist is greater than threshold 0.2.
ulb: 0.20 ✓ ulb: 0.14 ✓
tr: 0.20 ✓ tr: 0.14 ✓
true support
be pruned since the ulb of a subset AB does not satisfy threshold.
* similar itemsets :
{B}: <0.3, 0.7> (0.14),
{A,C}: <0.6, 0.4>(0.2),
{B,C}: <0.3, 0.5>(0.14)
* dissimilarity threshold = 0.2
* similarity function : Euclidean distance

Fig. 6. An illustration of SPAMINE algorithm trace

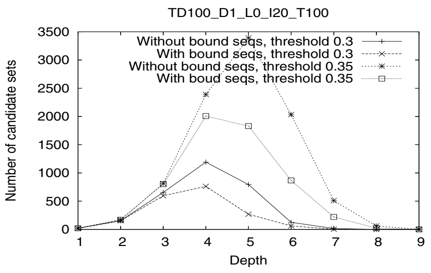
the dissimilarity threshold, the item is kept to B_1 for generating the next size candidate itemsets. In Fig. 6, only item B is a similar item but items A and C are also kept for generating the next size candidate itemsets.

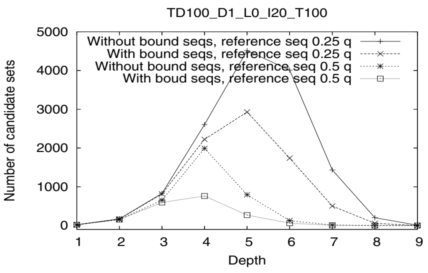
Generate candidate itemsets and their upper and lower bound support sequences (Step 6): All size k ($k > 1$) candidate itemsets(C_k) are generated using size $k-1$ itemsets(B_{k-1}) whose upper lower-bounding distances satisfy the dissimilarity threshold. If any subset of size $k-1$ of the generated itemset is not in the B_{k-1}, the candidate itemset is eliminated according to Lemma 2. The upper and lower bound support sequences of candidate itemsets are generated using Definition 3.

Prune candidate itemsets using their lower bounding distances (Step 7): If the lower bounding distance of the upper and lower bound support sequences of a candidate itemset exceeds the dissimilarity threshold, the candidate itemset is eliminated according to Lemma 1. For example, in Fig. 6, the lower bounding distance of itemset {A, B} is 0.22. Because the value is greater than the threshold 0.2, the candidate itemset is pruned.

Scan the transaction data and generate the support time sequences (Step 8): The supports of survived candidates are computed during the scan of the transaction data from time slot t_1 to t_n, and their support time sequences($\boldsymbol{S}_k$) are generated.

Find similar itemsets (Step 9): The true distance between the support time sequence of an itemset and the reference sequence is computed. If the value satisfies the threshold, the itemset is included in the result set(R_k). On the fly, if the upper lower-bounding distances of candidate itemsets satisfy the threshold, the itemsets are added to B_k for generating the next size candidate itemsets. The size of examined itemsets is increased to $k = k + 1$. The above procedures(*Steps 6-10*) are repeated until no itemset in B_k remains.





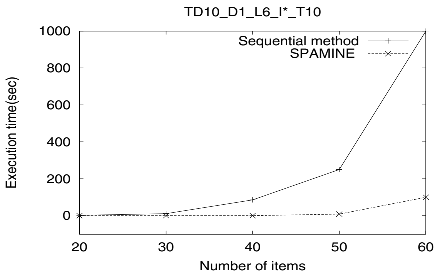
(a) (b)

Fig. 7. Effect of lower bounding distance pruning of bounds of support sequence

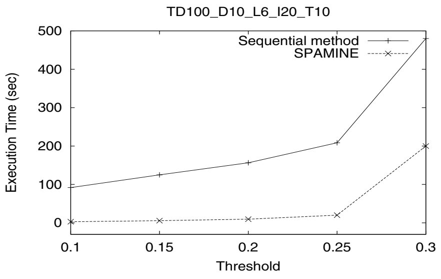
4 Experimental Evaluation

The proposed algorithm (SPAMINE) was evaluated using synthetic and real datasets. Synthetic datasets were generated with modifying a transaction data generator [5]. In the rest of paper, we use the following parameters to characterize the synthetic datasets we used. TD is the total number of transactions($\times$ 1,000), D is the number of transactions per time slot($\times$ 1,000), I is the number of distinct items, L is the average size of transactions, and T is the number of time slots. A reference time sequence was generated by choosing randomly a support sequence of an itemset or by selecting a support value near a quartile e.g., 0.25, 0.5, 0.75, of the sorted supports of single items at each time slot. The default reference time sequence was chosen near the 0.5 quartile. For the experiment with real data, we used a Earth climate dataset which includes monthly measurements of various climate variables(e.g., temperature and precipitation) and other related variables(e.g., Net Primary Production). This dataset is non public and was obtained from an Earth science project [1]. Throughout the experiment, we used the normalized Euclidean distance as a similarity function. For the experimental comparison, an alternative algorithm, a sequential method [42] is used. All experiments were performed on a workstation with Intel Xeon 2.8 GHz with 2 Gbytes of memory running the Linux operating system. The following presents the experimental results.

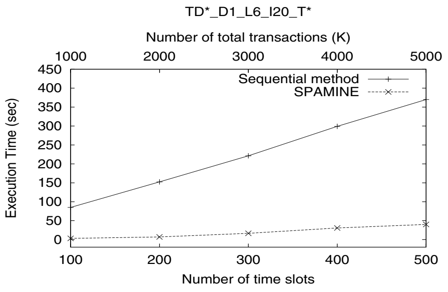
1) Effect of pruning by bounds of support sequences: In this experiment, we examined the pruning effect by low bounding distance of the upper and lower bound support sequences. Two versions of the SPAMINE algorithm are used. one uses the bounds of support sequences and the other does not. A synthetic dataset, TD100-D1-L10-I20-T100, is used and the dissimilarity threshold is set to 0.2. Fig. 7 (a) shows the number of candidates which need the database scan to compute their support values. As can be seen, the algorithm using the pruning



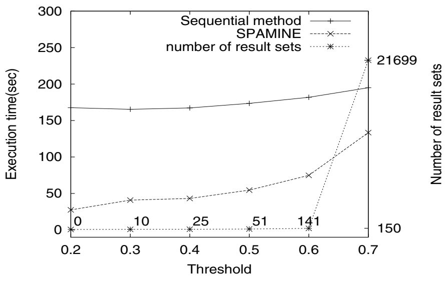
(a) Effect of number of items



(b) Effect of threshold



(c) Effect of number of time slots



(d) With a real dataset

Fig. 8. Experimental Result

scheme based on the bounds of support sequences and the lower bound distance produces fewer candidate itemsets. Fig. 7 (b) shows the results with different reference sequence types.

2) Effect of number of items: In this experiment, synthetic datasets of different number of items, TD10_D1_L6_I*_T10, are used. As seen in Fig. 8 (a), SPAMINE showed a similar execution time and received a less effect with the increase of small numbers of items. In contrast, the execution time of the sequential method dramatically increased.

3) Effect of similarity threshold: The performance of the two algorithms are examined with different threshold values. The TD10_D1_L10_I20_S10 dataset was used for that. As can be seen in Fig. 8 (b), SPAMINE had overall less execution time than the sequential method.

4) Effect of number of time slots: This experiment examined the effect of number of time slots using synthetic datasets, TD*_D1_L6_I20_T*. The number of transactions per time slot was fixed but the total dataset size increased with the number of time slots. Reference sequences were chosen near the 0.5 quartiles in each dataset and the threshold was 0.1. As seen in Fig. 8 (c), SPAMINE's execution time increases slowly with increase of number of time slots. In contrast, the sequential method's execution time rapidly increases.

5) Evaluation with real data: For this experiment, an Earth climate dataset was used. The dataset consists of global snapshots of measurement values for a number of variables (e.g., temperature, precipitation, NPP, CO2 and Solar). The data was measured at points (grid cells) on latitude-longitude spherical grids of 0.5 degree $\times$ 0.5 degree. For our analysis, we used measurement values in the Australia region since the climate phenomena in Australia has been known to be linked to El Niño, the anomalous warning of the eastern tropical region of the Pacific [39]. The total number of grids in the Australia data, i.e., the number of transactions per time slot, was 2827. First, we removed seasonal variation form the time series measurement data using a monthly Z score, i.e., by subtracting off the mean and dividing by the standard deviation. We defined event items based on 4 percentiles from the time series data of each variable(e.g., PRECI-LL, PRECI-L, PRECI-H, PRECI-HH). The total number of items for our analysis was 50. The dataset is available at monthly intervals from 1982 to 1999. We used 214 months of data, i.e., the number of time slots was 214. The total number of transactions was 604,978. For the reference sequence, the sequence of Southern Oscillation Index(SOI) was used, which is one of the indexes related to the El Niño phenomenon. Since the SOI index range is different from the support range, the index values were normalized to the range of 0 to 1 using a min-max normalization method. The transformation of a raw value x was calculated as $(x - x_{min})(1 - 0)/(x_{max} - x_{min}) + 0$, where x_{min} is the minimum value of raw value x, and x_{max} is the maximum value of x. Fig. 8 (d) shows the execution time and number of result sets by different thresholds. The pattern result shows that the prevalence variations of PRECI-L(0.20), CO2-L(0.21), Solar-H(0.25), NPP-L(0.26), PRECI-L & CO2-L(0.21), NPP-L & CO2-L(0.23), PRECI-L & NPP-L(0.27), PRECI-L & NPP-L & CO2-L(0.29), etc. were very related with the El Niño index sequence, with dissimilarity values of around 0.2.

5 Related Work

Although much work has been done on finding association patterns and similar time series, little attention has been paid to temporal association patterns that can discover similar variance groups over time. The closest related efforts have attempted to capture special temporal regulations of frequent association patterns such as cyclic association rule mining and calendar-based association rule mining [35,37,28,29] in temporal association mining. Özden et al.[35] examined cyclic association rule mining, which detects periodically repetitive patterns of frequent itemsets over time. Cyclic associations can be considered as itemsets that occur in every cycle with no exception. The work of [35] was extended in [37] for relaxed match. The cyclic association rules may not hold on all but most of the time points defined by the temporal patterns. Li et al.[28] explored the problem of finding frequent itemsets along with calendar-based patterns. The calendar-based patterns are defined with a calendar schema, e.g., (year, month, day). For example, (*,10,31) represents the set of time points each corresponding to the 31st day of October. However, real-life patterns are usually imperfect and may not demonstrate any regular periodicity. In the work of [29], a temporal pattern defines the set of time points where the user expects a discovered itemset to be frequent. However, our temporal patterns are searched with a user defined numeric reference sequence, and consider the prevalence similarity of all possible itemsets not only frequent itemsets.

In temporal pattern mining, sequential patterns mining [9] considers the relative order of the transactions of one customer. In [9], a frequent sequence is defined to consist of frequent itemsets taking place in separate consecutive transactions of the same user. The notion of episodes was introduced in [32]. An episode uses the data model of a sequence of elements, where the inter-element causality happens within a window of a given size. The frequency of an episode is the number of windows that contain the episode. In [11], frequent event patterns are found from time sequences which satisfy a user-specified skeleton. The user-specified skeleton is defined with a reference event and temporal constraints with time granularities. For example, the work can find events which frequently happen within two business days after a reference event, e.g., a rise of the IBM stock price. Recent work has applied mining techniques in a data streaming context. The temporal frequency counting problem for a data stream environment was proposed by [40]. The work is based on the model of inter-transaction association rules [31] for searching associations between itemsets that belong to different transactions performed by the same user in a given time span.

Other studies in temporal data mining have discussed the change of found association rules. Dong et al. [17] presented the problem of mining emerging patterns, which are the itemsets whose supports increase significantly from one dataset to another. The concept of emerging patterns can capture useful contrasts between classes. Ganti et al.[19] presented a framework for measuring difference in two sets of association rules from two datasets. Liu et al.[30] studied the change of fundamental association rules between two time periods using support and confidence. When new transactions are added to the original dataset, the maintenance of discovered association rules with an incremental updating technique was proposed in [15]. In contrast, [9] addressed the problem of monitoring the support and confidence of association rules. First, all frequent rules satisfying a minimum threshold from different time periods are mined and collected into a rule base. Then interesting rules can be queried by specifying shape operators(e.g., ups and downs) in support or confidence over time. On the other hand, online association rule mining was proposed by [24] to give the user the freedom to change the support threshold during the first scan of the transaction sequence.

6 Conclusion

We presented the problem of mining similarity-profiled temporal association patterns. Current similarity model using a $\mathcal{L}_p$ norm-based similarity function is a little rigid in finding similar temporal patterns. It may be interesting to consider a relaxed similarity model to catch temporal patterns which show similar trends but phase shifts in time. For example, the sale of items for clean-up such as chain saws and mops would increase after a storm rather than on the way of the storm. The current framework considers whole-sequence matching for the similar temporal patterns. However, a pattern's similarity may not persist for entire length of the sequence and so may manifest only in some segments. These relaxations present many new challenges for the automatic discovery of all partial temporal patterns based on the similarity constraint. The field of temporal data mining is relatively young and one expects to see many new developments in the near future.

References

1. Discovery of Changes from the Global Carbon Cycle and Climate System Using Data Mining, `http://www.cs.umn.edu/old-ahpcrc/nasa-umn/`
2. NOAAEconomics, `http://www.ncdc.noaa.gov/oa/esb/?goal=climate&file=users/business/`
3. Weather.com, `http://www.weather.com/aboutus/adsales/research.html`
4. After Katrina: Crisis Management, The Only Lifeline Was the Wal-Mart. FORTUNE Magazine, October 3 (2005)
5. Agarwal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proc. of the International Conference on Very Large Databases, VLDB (1994)
6. Agrawal, R., Faloutsos, C., Swami, A.: Efficient Similarity Search in Sequence Databases. In: Proc. of the International Conference on Foundations of Data Organization, FODO (1993)
7. Agrawal, R., Lin, K.I., Sawhney, H.S., Shim, K.: Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-series Database. In: Proc. of the International Conference on Very Large Databases (VLDB) Conference (1995)
8. Agrawal, R., Lin, K.I., Sawhney, H.S., Shim, K.: Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-series Database. In: Proc. of the International Conference on Very Large Databases (VLDB) Conference (1995)
9. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Proc. of the IEEE International Conference on Data Engineering, ICDE (1995)
10. Bettini, C., Jajodia, S., Wang, X.S.: Time Granularities in Databases, Data Mining and Temporal Reasoning. Springer, Heidelberg (2000)
11. Bettini, C., Wang, X.S., Jajodia, S., Lin, J.: Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences. IEEE Transactions on Knowledge and Data Engineering 10(2) (1998)
12. Box, G., Jenkins, G., Reinsel, G.: Time Series Analysis: Forecasting and Control. Prentice-Hall, Englewood Cliffs (1994)
13. Calders, T.: Deducing Bounds on the Frequency of Itemsets. In: Proc. of EDBT Workshop DTDM Database Techniques in Data Mining (2002)
14. Chan, F.K., Fu, A.W.: Efficient Time Series Matching by Wavelets. In: Proc. of International Conference on Data Mining (1999)

15. Cheung, W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In: Proc. of the IEEE International Conference on Data Engineering, ICDE (1996)
16. Das, G., Gunopulos, D., Mannila, H.: Finding Similar Time Series. In: Proc. of Principles of Data Mining and Knowledge Discovery, European Symposium (1997)
17. Dong, G., Li, J.: Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (1999)
18. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast Subsequence Matching in Time-series Database. In: Proc. of the ACM SIGMOD International Conference on Management of Data (1994)
19. Ganti, V., Gehrke, J., Ramakrishnan, R.: A Framework for Measuring Changes in Data Characteristics. In: Proc. of the ACM PODS Conference (1999)
20. Gunopulos, D., Das, G.: Time Series Similarity Measures. Tutorial Notes of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2000)
21. Gunopulos, D., Das, G.: Time Series Similarity Measures and Time Series Indexing. SIGMOD Record 30(2) (2001)
22. Han, J., Fu, Y.: Discovery of Multi-level Association Rules From Large Databases. In: Proc. of the International Conference on Very Large Databases, VLDB (1995)
23. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of the ACM SIGMOD International Conference on Management of Data (2000)
24. Hidber, C.: Online Association Rule Mining. In: Proc. of the ACM SIGMOD International Conference on Management of Data (1998)
25. Hsu, W., Lee, M., Wang, J.: Temporal and Spatio-temporal Data Mini. IGI Publishing (1997)
26. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Journal of Knowledge and Information Systems 3(3) (2001)
27. Keogh, E., Ratanamahatana, C.A.: Exact Indexing of Dynamic Time Warping. Knowledge and Information Systems 17(3), 358–386 (2005)
28. Li, Y., Ning, P., Wang, X.S., Jajodia, S.: Discovering Calendar-Based Temporal Assocation Rules. Journal of Data and Knowledge Engineering 15(2) (2003)
29. Li, Y., Zhu, S., Wang, X.S., Jajodia, S.: Looking into the Seeds of Time: Discovering Temporal Patterns in Large Transaction Sets. Journal of Information Sciences 176(8) (2006)
30. Liu, B., Hsu, W., Ma, Y.: Discovering the Set of Fundamental Rule Change. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2001)
31. Lu, H., Han, J., Feng, L.: Stock Movement Prediction and N-Dimensional Inter-Transaction Association Rules. In: Proc. of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (1998)
32. Mannila, H., Toivonen, H., Verkamo, A.: Discovering Frequent Episodes in Sequences. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (1995)
33. Mitsa, T.: Temporal Data Mining. Chapman and Hall/CRC (2010)
34. NOAA. El Nino Page, `http://www.elnino.noaa.gov/`
35. Ozden, B., Ramaswamy, S., Silberschatz, A.: Cyclic Association Rules. In: Proc. of the IEEE International Conference on Data Engineering, ICDE (1998)

36. Park, J.S., Chen, M., Yu, P.: An Effective Hashing-based Algorithm for Mining Association Rules. In: Proc. of the ACM SIGMOD International Conference on Management of Data (1995)
37. Ramaswamy, S., Mahajan, S., Silberschatz, A.: On the Discovery of Interesting Patterns in Association Rules. In: Proc. of the International Conference on Very Large Database (VLDB) (1998)
38. Srikant, R., Agrawal, R.: Mining Generalized Association Rules. In: Proc. of the International Conference on Very Large Databases, VLDB (1995)
39. Taylor, G.H.: Impacts of El Nino on Southern Oscillation on the Pacific Northwest, `http://www.ocs.orst.edu/reports/enso_pnw.html`
40. Teng, W., Chen, M., Yu, P.: A Regression-Based Temporal Pattern Mining Scheme for Data Streams. In: Proc. of the International Conference on Very Large Databases, VLDB (2003)
41. Yi, B.K., Faloutsos, C.: Fast Time Sequence Indexing for Arbitrary $\mathcal{L}_p$ norms. In: Proc. of the International Conference on Very Large Data Bases, VLDB (2000)
42. Yoo, J.S., Shekhar, S.: Similarity-profiled Temporal Association Mining. IEEE Transactions on Knowledge and Data Engineering 21(5), 1147–1161 (2009)