Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 05-010

Mining Time-Profiled Associations: A Preliminary Study Report

Jin Soung Yoo, Pusheng Zhang, and Shashi Shekhar

April 04, 2005

Mining Time-Profiled Associations: A Preliminary Study

Jin Soung Yoo, Pusheng Zhang, Shashi Shekhar
Computer Science Department, University of Minnesota
200 Union Street SE, Minneapolis, MN-55455
[$jyoo, pusheng, shekhar$]@cs.umn.edu

April 3, 2005

Abstract

A time-profiled association is an association pattern consistent with a query sequence along time, e.g., identifying interacting relationship of droughts and wild fires in Australia with the El Nino phenomenon in the past 50 years. Association patterns by traditional association rule mining approaches reveal the generic dependency among variables, however, the evolution of these patterns along time is not captured. Hence the time-profiled association mining is used to incorporate the temporal evolution of association patterns and identify the co-occurred patterns consistent along time. Mining time-profiled associations is computational challenging due to large size of itemset space and long time points in practice. A naive approach of mining time-profiled associations can be characterized using a two-phase paradigm. The first phase generates the statistical parameter (e.g., support) sequences along time, and the second phase retrieves similar sequences with the query sequence. However, exponentially increasing computational costs of generating all combinatorial candidate itemsets become prohibitively expensive for the previous work. In this paper, we propose a novel one-step algorithm to unify the generation of the sequence of statistical parameters and sequence retrieval. The proposed algorithm substantially reduces the itemset search space by pruning candidate itemsets based on the monotone property of lower bounding measure of sequence of statistical parameters. Experimental results show that our algorithm outperforms the naive approach.

1 Introduction

A *time-profiled association* is an association pattern [2], which consists with a query sequence over time. One example is the frequent co-occurrences of climate features with the El Nino phenomenon over the last 50 years [13]. El Nino, an abnormal warming in the eastern tropical Pacific Ocean[1], has been linked to climate phenomena such as droughts and wild fires in Australia and heavy rainfall along the eastern coast of South America in the past 50 years [12]. Frequent association patterns discovered by traditional association rule mining[2] reveal the generic dependency among item types of transactions by collapsing the time consideration, e.g., the sales of diaper and beer. However, transaction data are implicitly associated with time, i.e., any transaction is associated with a certain time slot. Thus the association patterns might change over time. For example, a sales association between diaper and beer is high only in the evening but not in other time slots. Association patterns found might have different popularity levels over time as shown in Figure 1. These variations in time are not captured under traditional association rule mining. Hence time-profiled association mining can be used to discover interacting relationships consistent with a query prevalence sequence over time. Mining time-profiled associations is crucial to many applications which analyze temporal trends of interactions among variables, including Earth science, climatology, public health, and commerce.

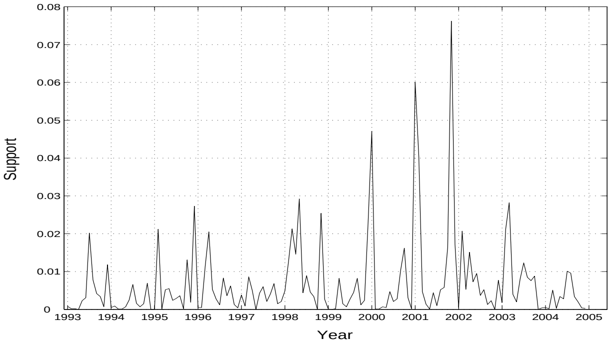


Figure 1: An example of temporal variation of association patterns

Mining time-profiled associations is computationally challenging since the sizes of itemset space and temporal space are extremely large in practice. In the example of the El Nino investigation, there are millions of spatial units with climate features (e.g., temperature and precipitation), each having 50 years worth of daily observations, i.e.,50*12*365=21,900 time points. An observation at one time point in a specific location can be treated as one transaction, so there are more than millions of transactions globally at one time snapshot. Therefore, exploring itemsets of climate features will involve large itemset space and long time series, and exploring all relationships among features would be even more exorbitant.

To our best knowledge, there is no prior work directly tackling the problem of mining time-profiled associations. Some relevant work has attempted to capture temporal dynamics of association patterns, including active data mining [4], cyclic association rule mining [10], and calendar-based association rule mining [9]. Agrawal el al.[4] proposed a specific query shape based rule query as the active data mining paradigm. The query shape was only limited to a discrete sequence of the fluctuations, such as up and down. Özden et al.[10] identified cyclic association rules, which discover frequent periodically repetitive patterns. Li et al.[9] discovered frequent calendar schemas, e.g., year, month, and day, based on repeated patterns satisfying given support and confidence thresholds. Both cyclic association rule mining and calendar-based association rule mining might not be directly applied to identify consistent associations over time with a query sequence.

A naive approach to mining time-profiled associations can be characterized using a two-phase paradigm. The first phase updates the history of the statistical parameters (e.g., support) for rules at different time points using traditional *Apriori* [2] approach, and generates a sequence of statistical parameters. The second phase matches the sequences of statistical parameters to find time-profiled associations with the query sequence. However, exponentially increasing computational costs of generating all combinatorial candidate itemsets become prohibitively expensive. We propose a novel one-step algorithm to unify the generation of statistical parameter sequences and sequence searching. The proposed algorithm prunes the candidate itemsets by using the monotone property of the lower bounding measure of sequence of statistical parameters. It substantially reduces the search space of itemsets, and is efficient in terms of the number of candidate itemset generations. Experimental results show that our algorithm outperforms the naive approach.

Scope and Outline: In this paper, we focus on mining time-profiled associations using *Apriori*-based[2] association mining approaches. Issues beyond the scope of this paper include mining time-profiled associations using other association rule mining approaches[7, 8] and similarity searching in time series databases[6].

The rest of the paper is organized as follows. Section 2 formally defines the problem of mining time-profiled association patterns, and Section 3 introduces basic concepts and properties associated with time-profiled association patterns. An efficient one-phase algorithm of mining time-profile associations is proposed in Section 4, and the experimental results are presented in Section 5. We summarize our work in Section 6.

2 Problem Statement

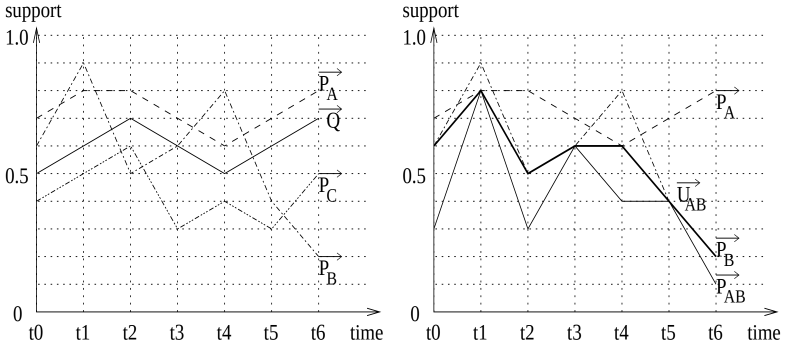
A time-profile association is an association pattern consistent with a specific time sequence over time. The problem of mining the time-profiled association patterns is to find all itemsets whose time sequences of prevalence measure values are similar to user specified query sequence under the given similarity threshold. The detailed problem definition is described as follows.

Given:
1) A set of items $E = \{e_1, \ldots, e_m\}$.
2) A time-stamped transaction database D.
Each transaction $d \in D$ is a tuple $< time - stamp, itemset >$ where $time - stamp$ is a time that the transaction d was executed and $itemset$ is a set of items where are a subsets of E.
3) A time unit t. The ith time slot t_i, $0 \leq i < n$, corresponds to the time interval $[i \cdot t\ ,\ (i+1) \cdot t)$. The set of transactions executed in t_i is denoted by D_i.
4) A query time sequence $\vec{Q} =< q_0, \ldots, q_{n-1} >$ over time slots $t_0, \ldots, t_{n-1}$
5) A threshold of similarity value θ
Find:
Find a complete and correct set of itemsets $I \subseteq E$ where $f_{similar}(\vec{P_I},\ \vec{Q}) \leq \theta$, where $\vec{P_I} =< p_0^I, \ldots, p_{n-1}^I >$ is the time sequence of prevalence values of an itemset I over time slots $t_0, \ldots, t_{n-1}$ and $f_{similar}(\vec{P_I},\ \vec{Q})$ is the similarity function between two sequences, $\vec{P_I}$ and $\vec{Q}$.
Objective: Minimize computation cost.

We assume that a query time sequence $\vec{Q}$ is in the same scale of the prevalence measures or can be transformed to the same scale.

time	tid	items	time	tid	items	time
t0	1	A	t1	11	C	..
t0	2	A	t1	12	A, B, C	..
t0	3	A	t1	13	A, B, C	..
t0	4	A, C	t1	14	A, B	
t0	5	A, B, C	t1	15	A, B	
t0	6	A, B	t1	16	A, B	
t0	7	A, B	t1	17	A, B	
t0	8	B, C	t1	18	A, B	
t0	9	B, C	t1	19	A, B	
t0	10	B	t1	20	A, C	

(a)



(b)

(c)

Figure 2: (a) An example of time-stamped transactions (b) Support time sequences of single items and a query time sequence (b) Upper bound of support time sequences of itemset AB

3 Properties of Time-Profiled Associations

3.1 Basic Concepts

Prevalence measures, e.g., support, have been successfully used to reduce the itemset search space in traditional association rule mining. Similarly, we propose to adopt a time sequence of prevalence measures for time-profiled association mining.

3.1.1 Support Time Sequence

We use support as a prevalence measure in the transaction database since it represents how statistically significant a pattern is, and it has an anti-monotone property [11].

Definition 1 *Given a time-stamped transaction database $D = D_0 \cup \ldots \cup D_{n-1}$, where D_i is a set of transactions executed in ith time slot, $0 \leq i < n$, the support of an itemset I at a time slot t_i is the fraction of transactions d in D_i that contain the itemset I such that*

$$support_{D_i}(I) = \frac{|\{d \in D_i | I \subseteq d\}|}{|D_i|}$$

The support time sequence of an itemset I, $\vec{P_I} =< p_0^I, \ldots, p_{n-1}^I >$ is the time sequence of support values of an itemset I over time slots $t_0, \ldots, t_{n-1}$ such that

$$\vec{P_I} =< support_{D_0}(I), \ldots, support_{D_{n-1}}(I) >$$

Figure 2 (a) shows an example of time-stamped transaction database. Figure 2 (b) shows the support time sequences of single items, i.e., A, B and C.

3.1.2 Choice of Similarity Measure

Several similarity measures have been proposed in the time series literature [6]. We propose using Euclidean distance as the similarity measure between two sequences because it is a typical similarity measure and is useful in many applications [3, 5]. For two time sequences $\vec{X} =< x_0, \dots, x_{n-1} >$ and $\vec{Y} =< y_0, \dots, y_{n-1} >$, the Euclidean similarity measure is defined as

$$f_{similar}(\vec{X}, \vec{Y}) = D(\vec{X}, \vec{Y}) = \sqrt{\sum_{i=0}^{n-1} (x_i - y_i)^2}$$

If this distance is below a user-defined threshold θ, we say that the two sequences are similar. Addressing issues like dedicate similarity measures is outside the scope of this paper.

3.2 Upper Bound Time Sequence and Lower Bounding Measure

Lemma 1 *Let I_{k+1} be a size $k+1$ itemset $\subseteq E$ and $\{I_k^1, \dots, I_k^{k+1}\}$ be a set of all size k sub itemsets of I_{k+1}, where $I_k \subset I_{k+1}$. Let $\vec{P}_{I_{k+1}} =< p_0^{I_{k+1}}, \dots, p_{n-1}^{I_{k+1}} >$ be the support time sequence of I_{k+1} and $\vec{P}_{I_k} =< p_0^{I_k}, \dots, p_{n-1}^{I_k} >$ be the support time sequence of I_k. The upper bound sequence of $\vec{P}_{I_{k+1}}$, $\vec{U}_{I_{k+1}} =< u_0^{I_{k+1}}, \dots, u_{n-1}^{I_{k+1}} >$ is $< min\{p_0^{I_k^1}, \dots, p_0^{I_k^{k+1}}\}, \dots, min\{p_{n-1}^{I_k^1}, \dots, p_{n-1}^{I_k^{k+1}}\} >$.*

Figure 2(c) shows the upper bound of support time sequence of itemset AB.

Definition 2 *Given a query time sequence $\vec{Q}$, the lower bounding measure between $\vec{Q}$ and the support time sequence $\vec{P_I}$ of an itemset I is defined as*

$$D_{lb}(\vec{Q}, \vec{P_I}) = \sqrt{\sum_{i=0}^{n-1} (q_i - u_i)^2},\ q_i \geq u_i,$$

where i is a time slot, $q_i \in \vec{Q} =< q_0, \dots, q_{n-1} >$ and $u_i \in \vec{U_I} =< u_0, \dots, u_{n-1} >$, the upper bound time sequence of $\vec{P_I}$.

Lemma 2 *For the true similarity measure $D(\vec{Q}, \vec{P_I})$ and the lower bounding measure $D_{lb}(\vec{Q}, \vec{P_I})$ of a query time sequence $\vec{Q}$ and the support time sequence $\vec{P_I}$ of an itemset I, the following inequality holds:*

$$D_{lb}(\vec{Q}, \vec{P_I}) \leq D(\vec{Q}, \vec{P_I})$$

3.3 Monotone Property of the Lower Bounding Measure

Lemma 3 *Let $\vec{P_{I_k}}$ be a time sequence of supports of a size k itemset I_k and $\vec{P}_{I_{k+1}}$ be a time sequence of supports of a size k+1 itemset I_{k+1} , where $I_{k+1} = I_k \cup I_1$, $I_1 \notin I_k$. The following inequality holds:*

$$D_{lb}(\vec{Q}, \vec{P}_{I_k}) \leq D_{lb}(\vec{Q}, \vec{P}_{I_{k+1}})$$

It is clear by Lemma 1 and Definition 2. The upper bound of support time sequence of an itemset does not increase with increasing size of the itemset. As a result the lower bounding does not decrease with increasing size of itemset. For a similarity threshold θ, if $D_{lb}(\vec{Q}, \vec{P}_{I_k}) > \theta$ then $D_{lb}(\vec{Q}, \vec{P}_{I_{k+1}}) > \theta$. Lemma 3 ensures that the lower bounding measure can be used to to effectively prune the search space and efficiently find interesting itemsets.

4 Time-Profiled Association Mining Algorithm

The naive method for finding time-profiled association patterns follows a two-step procedure. First, it counts the supports of all possible itemsets over the time-stamped database and generates their support time sequences. Second, it searches time sequences similar with a query time sequence. However, as the number of items and transactions increases, the computation cost of all combination itemsets becomes prohibitively expensive. We propose a one-step algorithm to combine the generation of support time sequence and the time sequence search, and use a tightening upper bound. It is efficient because it considers fewer candidate itemsets for counting of supports and searching of time sequences in a single pass. Our algorithm prunes the candidate itemsets by using the monotone property of the lower bounding measure of support time sequences without considering the time-stamped transactions in the database and even without computing their true similarity measure. The following is the simple description of the algorithm.

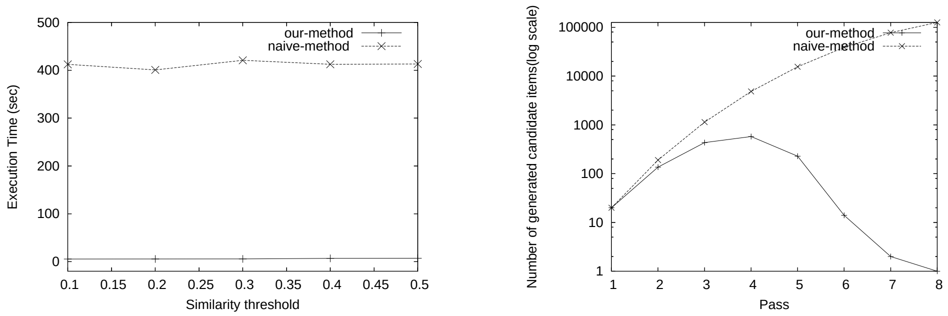
Generation of support time sequences of single items : In the first scan of a time-stamped database, the supports of all single items ($k = 1$) are counted per each time slot and their support time sequences are generated. If the lower bounding measure between a query sequence and the support time sequence is greater than a given similarity threshold value, the single item is pruned from the candidate set. If the true similarity value between them satisfies the threshold, the item is added to a result set.

Generation of candidate itemsets : All size $k + 1$ candidate itemsets are generated using size k candidate itemsets.

Generation of upper bound sequences : The upper bound time sequences of size $k + 1$ candidate itemsets are generated using the support sequences of their size k subsets.

Pruning of candidate itemsets using the lower bounding measure : Calculate the lower bounding measure between the upper bound sequence of the candidate itemset and the query time sequence. If the lower bounding measure is greater than the similarity threshold, the candidate itemset is eliminated from the set of candidate itemsets.

Scanning the database and finding itemsets showing similar support time sequences : The supports of candidate itemsets after pruning are counted from the database and their support time sequences are calculated. If the similarity value between the support sequences and the query sequence is less than the threshold value, the itemset is included in the result set. The size of examined itemsets is increased to $k = k + 1$ and the above procedures are repeated until no candidate itemset remains in the previous pass.



(a) (b)

Figure 3: Experiment Results: (a) Effect of threshold (b) Effect of pruning

5 Experimental Evaluation

Our experiments were performed to examine the effect of different threshold values and the effect of itemset pruning by the lower bounding measure. The results were compared with the naive method. The dataset was generated using the transaction generator designed by the IBM Quest project used in [2]. We added a time slot parameter for generating time-stamped transactions. All experiments were performed on a workstation with 4 processors, each an Intel Xeon 2.8 GHz with 3 Gbytes of memory running the Linux operating system.

Effect of similarity threshold : The effect of similarity measure was examined with different similarity thresholds using a synthetic dataset in which the total number of transactions was 100,000, the number of items was 20, the average size of transaction was 10 and the number of time slots was 10. The query sequence was chosen near the median support value of single items at each time slot. In Figure 3 (a), our method showed dramatically less execution time compared with the naive approach. With the increase in the similarity threshold, the execution time increased. Otherwise, the naive approach showed stable execution time because the approach calculated all time sequences of all combination itemsets independent of the threshold value.

Effect of lower bounding pruning : Figure 3 (b) shows the number of generated candidate itemsets per each pass in the experiment using the same dataset. Note that the y value is in log scale. Our method generated much fewer candidate itemsets compared with the naive method.

6 Conclusions

We introduced the problem of mining time-profiled association patterns and proposed a one-phase algorithm to efficiently discover time-profiled associations. The proposed algorithm substantially reduced the search space by pruning candidate itemsets based on the monotone property of the lower bounding measure of the sequence of statistical parameters. Experimental results showed that our algorithm outperformed the naive approach.

References

[1] NOAA El Nino Page. http://www.elnino.noaa.gov/.

[2] R. Agarwal and R. Srikant. Fast algorithms for Mining association rules. In *Proc. of the 20th VLDB*, 1994.

[3] R. Agrawal, C. Faloutsos, and A.Swami. Efficient Similarity Search in Sequence Databases . In *Proc. Int. Conference on Foundations of Data Organization(FODO)*, 1993.

[4] R. Agrawal and G. Psaila. Active Data Mining. In *Proc. The First International Conference on Knowledge Discovery and Data Mining*, 1995.

[5] C. Faloutsos, M. Ranganathan, and Y.Manolopoulos. Fast subsequence matching in time-series database. In *Proc. ACM SIGMOD Conference*, 1993.

[6] D. Gunopulos and G. Das. Time Series Similarity Measures and Time Series Indexing. *SIGMOD Record*, 30(2), 2001.

[7] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In Weidong Chen, Jeffrey Naughton, and Philip A. Bernstein, editors, *2000 ACM SIGMOD Int'l. Conference on Management of Data*, pages 1–12, 2000.

[8] J. Hipp, U. Güntzer, and G. Nakjaeizadeh. Algorithms for Association Rule Mining - A General Survey and Comparison. *ACM SIGKDD Explorations*, 2(1), 2000.

[9] Y. Li, P. Ning, X. S. Wang, and S. Jajodia. Discovering Calendar-Based Temporal Assocation Rules. In *Proc. Int. Symposium Temporal Representation and Reasoning(TIME)*, 2001.

[10] B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic Association Rules. In *Proc. of IEEE Int. Conference on Data Engineering*, 1998.

[11] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *the Proc. of ACM SIGKDD Int'l Conf. on Knowledge Discovery Data Mining*, 2002.

[12] G. H. Taylor. Impacts of el nino on southern oscillation on the pacific northwest. http://www.ocs.orst.edu/reports/enso_pnw.html.

[13] P. Zhang, M. Steinbach, V. Kumar, S. Shekhar, P. Tan, S. Klooster, and C. Potter. Discovery of Patterns of Earth Science Data Using Data Mining. In Mehmed M. Kantardzic and Jozef Zurada, editors, *to appear in Next Generation of Data Mining Applications*. IEEE Press, 2004.