

# Mining Time-Profiled Associations: An Extended Abstract

Jin Soung Yoo, Pusheng Zhang, and Shashi Shekhar\*

Computer Science & Engineering Department, University of Minnesota,  
200 Union Street SE, Minneapolis, MN 55455, U.S.A.  
[jyoo|pusheng|shekhar]@cs.umn.edu

**Abstract.** A time-profiled association is an association pattern consistent with a query sequence over time, e.g., identifying the interacting relationship of droughts and wild fires in Australia with the El Nino phenomenon in the past 50 years. Traditional association rule mining approaches reveal the generic dependency among variables in association patterns but do not capture the evolution of these patterns over time. Incorporating the temporal evolution of association patterns and identifying the co-occurring patterns consistent over time can be done by time-profiled association mining. Mining time-profiled associations is computationally challenging due to the large size of the itemset space and the long time points in practice. In this paper, we propose a novel one-step algorithm to unify generating statistical parameter sequences and sequence retrieval. The proposed algorithm substantially reduces the itemset search space by pruning candidate itemsets based on the monotone property of the lower bounding measure of the sequence of statistical parameters. Experimental results show that our algorithm outperforms a 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 [10]. 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. 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

---

<sup>1</sup> This work was partially supported by NSF grant 0431141 and Army High Performance Computing Research Center contract number DAAD19-01-2-0014. The content of this work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred. AHPCRC and Minnesota Supercomputer Institute provided access to computing facilities. Readers may refer to the technical report [9] for more details.

popularity levels over time. These variations in time are not captured under traditional association rule mining[2]. 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.

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.,  $50 \times 12 \times 365 = 21,900$  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 a pair of climate features will involve about trillion 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 [8], and calendar-based association rule mining [7]. However, these related work 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.

## 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 measures are similar to user specified query sequence under a given similarity threshold. The detailed problem definition is described as follows. We assume that a query time sequence  $Q$  is in the same scale of the prevalence measures or can be transformed to the same scale.

**Given:**

- 1) A set of items  $E = \{e_1, \dots, e_m\}$ .
- 2) A time-stamped transaction database  $D$ . Each transaction  $d \in D$  is a tuple  $\langle \text{time-stamp}, \text{itemset} \rangle$  where  $\text{time-stamp}$  is a time that the transaction  $d$  is executed and  $\text{itemset}$  is a set of items which is a subsets of  $E$ .
- 3) A time unit  $t$ . The  $i$ th 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  $\mathbf{Q} = \langle q_0, \dots, q_{n-1} \rangle$  over time slots  $t_0, \dots, t_{n-1}$ .
- 5) A threshold of similarity value  $\theta$ .

**Find:** A complete and correct set of itemsets  $I \subseteq E$  where  $f_{\text{similar}}(\mathbf{P}_I, \mathbf{Q}) \leq \theta$ , where  $\mathbf{P}_I = \langle p_0^I, \dots, p_{n-1}^I \rangle$  is the time sequence of prevalence values of an itemset  $I$  over time slots  $t_0, \dots, t_{n-1}$  and  $f_{\text{similar}}(\mathbf{P}_I, \mathbf{Q})$  is a similarity function between two sequences  $\mathbf{P}_I$  and  $\mathbf{Q}$ .

**Objective:** Minimize computational cost.

### 3 Properties of Time-Profiled Associations

#### 3.1 Basic Concepts

**Support Time Sequence :** We use support as the prevalence measure of an itemset since it represents how statistically significant a pattern is, and it has an anti-monotone property [2].

**Definition 1.** Given a time-stamped transaction database  $D = D_0 \cup \dots \cup D_{n-1}$ , the support time sequence  $\mathbf{P}_I$  of an itemset  $I$  is the sequence of support values of an itemset  $I$  over  $D_0, \dots, D_{n-1}$  such that

$$\mathbf{P}_I = \langle \text{support}_{D_0}(I), \dots, \text{support}_{D_{n-1}}(I) \rangle$$

where  $\text{support}_{D_i}(I) = |\{d \in D_i | I \subseteq d\}| / |D_i|$ .

**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 since it is a typical similarity measure and is useful in many applications [3,5]. For two time sequences  $\mathbf{X} = \langle x_0, \dots, x_{n-1} \rangle$  and  $\mathbf{Y} = \langle y_0, \dots, y_{n-1} \rangle$ , the Euclidean similarity measure is defined as  $f_{\text{similar}}(\mathbf{X}, \mathbf{Y}) = D(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=0}^{n-1} (x_i - y_i)^2}$ . If this distance is below a user-defined threshold  $\theta$ , we say that the two sequences are similar.

#### 3.2 Upper Bound Time Sequence and Lower Bounding Measure

**Lemma 1.** Let  $\mathbf{P}_{I_k} = \langle p_0^{I_k}, \dots, p_{n-1}^{I_k} \rangle$  be the support time sequence of a size  $k$  itemset  $I_k = \{i_1, \dots, i_k\} \subseteq E$  and  $\mathbf{P}_{J_k} = \langle p_0^{J_k}, \dots, p_{n-1}^{J_k} \rangle$  be the support time sequence of a size  $k$  itemset  $J_k = \{j_1, \dots, j_k\} \subseteq E$  where  $i_1 = j_1, \dots, i_{k-1} = j_{k-1}$  and  $i_k \neq j_k$ ,  $k \geq 1$ . Let  $\mathbf{P}_{I_{k+1}} = \langle p_0^{I_{k+1}}, \dots, p_{n-1}^{I_{k+1}} \rangle$  be the support time sequence of a size  $k+1$  itemset  $I_{k+1} = I_k \cup J_k = \{i_1, \dots, i_k, j_k\}$ ,  $k \geq 1$ . The upper bound time sequence  $\mathbf{U}_{I_{k+1}}$  of  $\mathbf{P}_{I_{k+1}}$  is  $\langle \min\{p_0^{I_k}, p_0^{J_k}\}, \dots, \min\{p_{n-1}^{I_k}, p_{n-1}^{J_k}\} \rangle$ .

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

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

where  $i$  is a time slot,  $q_i \in \mathbf{Q} = \langle q_0, \dots, q_{n-1} \rangle$  and  $u_i \in \mathbf{U}_I = \langle u_0, \dots, u_{n-1} \rangle$ , the upper bound time sequence of  $\mathbf{P}_I$ .

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

$$D_{lb}(\mathbf{Q}, \mathbf{P}_I) \leq D(\mathbf{Q}, \mathbf{P}_I)$$

### 3.3 Monotone Property of Lower Bounding Measure

**Lemma 3.** Let  $\mathbf{P}_{I_k}$  be the support time sequence of a size  $k$  itemset  $I_k$  and  $\mathbf{P}_{I_{k+1}}$  be the support time sequence 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:

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

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

## 4 Time-Profiled Association Mining Algorithm

We propose a one-step algorithm, Time-Profiled association MINing mEthod (TPMINE) to combine the generation of support time sequence and the time sequence search. Our algorithm prunes the candidate itemsets by using the monotone property of the lower bounding measure of support time sequences without scanning the transaction database and even without computing their true similarity measure. The following is the simple description of the TPMINE 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.

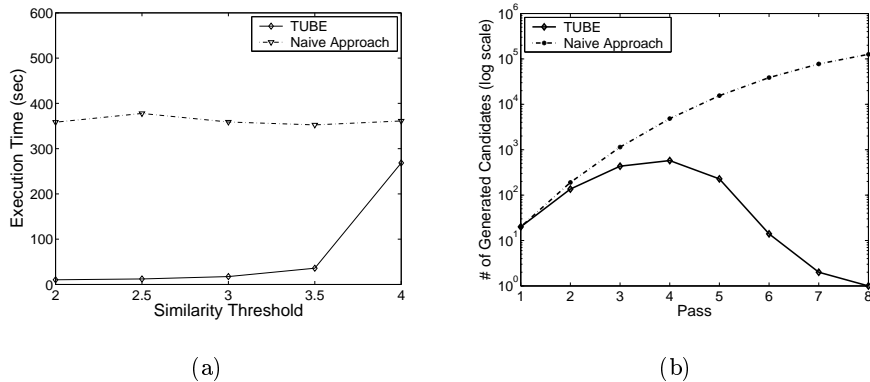


Fig. 1. Experiment Results: (a) Effect of threshold (b) Effect of pruning

**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 itemset are generated using the support sequences of their size  $k$  subsets.

**Pruning of candidate itemsets using lower bounding measure :** Calculate the lower bounding measure between the upper bound sequence of 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.

## 5 Experimental Evaluation

Our experiments were performed on both real and synthetic datasets. The synthetic 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 Fig. 1 (a), the TPMINE algorithm showed dramatically less execution time compared with the naive approach. With the increase in the similarity threshold, the execution time of TPMINE increased. Otherwise, the naive approach showed stable execution time because the approach calculated all time sequences of all combination itemsets independent with the threshold value.

**Effect of lower bounding pruning :** Fig. 1 (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. TPMINE generated much fewer candidate itemsets compared with the naive method.

The experiment results with other parameters, i.e., type of the query sequence, the number of items and the number of time slots, and a real dataset are referred in [9].

## 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 Conference*, 1994.
3. R. Agrawal, C. Faloutsos, and A. Swami. Efficient Similarity Search in Sequence Databases. In *Proc. Int. Conference on Foundations of Data Organization*, 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. Y. Li, P. Ning, X. Wang, and S. Jajodia. Discovering Calendar-Based Temporal Association Rules. In *Proc. Symp. Temporal Representation and Reasoning*, 2001.
8. B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic Association Rules. In *Proc. of IEEE Int. Conference on Data Engineering*, 1998.
9. J.S. Yoo, P. Zhang, and S. Shekhar. Mining Time-Profiled Associations. In *Tech Report, University of Minnesota*, 2005.
10. 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 M. Kantardzic and J. Zurada, editors, *in Next Generation of Data Mining Applications*. IEEE Press, 2005.