An Efficient Upper-bound Model for

Mining Weighted Sequential Patterns

Tzung-Pei Hong*
Department of Computer Science and Information Engineering
National University of Kaohsiung, 811, Taiwan
Department of Computer Science and Engineering
National Sun Yat-Sen University, 804, Taiwan
*tphong@nuk.edu.tw

Hong-Yu Lee
Department of Computer Science and Information Engineering
National University of Kaohsiung, 811, Taiwan
m0995511@mail.nuk.edu.tw

Guo-Cheng Lan
Department of Computer Science and Information Engineering
National Cheng Kung University, 701, Taiwan
rrfoheiay@gmail.com

Abstract

Weighted sequential pattern mining has widely been discussed in the recent years. In this paper, we propose a novel approach for finding weighted sequential patterns from sequence databases. In particular, an improved upper-bound model and a pruning strategy are proposed to obtain more accurate weighted upper-bounds for patterns and avoid evaluation of unpromising candidates. The experimental results also show the proposed approach has a good performance in terms of pruning effect and execution efficiency.

Keywords: Data mining, sequential pattern, weighted sequential pattern, upper bound.

1. Introduction

The main purpose of data mining on knowledge discovery is to extract useful rules or patterns from a set of data. In the field of data mining, sequential pattern mining [2] has been widely applied to trend analysis from a set of long-term event sequence data. The traditional sequential pattern mining, however, only considers the occurrence of items, and then it does not reflect any other factors, such as price or profit. Besides, the same significance was assumed for all items in a set of sequences. Thus, the actual significance of a pattern cannot be easily recognized. For example, some events with low-frequency are quite important ones, such as products with high-profit in a transaction database or attacked events in a long-term network monitor data. Such events may not be easily found by using traditional sequential pattern mining techniques. To handle this, Yun *et al* proposed a new research issue, namely weighted sequential pattern mining [8], to discover weighted sequential patterns from a set of sequence data. In Yun *et al.*'s study, different weights were assigned to items by the importance of each item. In addition, a weighted average function was designed to identify the weight value of a pattern in a sequence. The major challenge for the weighted sequential pattern mining was that the downward-closure property in traditional sequential pattern mining was not kept in weighted sequential pattern mining. To deal with the problem, Yun *et al.* [8] was designed an upper-bound method, which maximum weight among items in a sequence database was used as maximum weight of each sequence, to construct a new downward-closure property for weighted sequential pattern mining.

However, the upper-bound model used in Yun *et al.*'s study [8] could be applied to keep the downward-closure property of weighted sequential pattern mining, but we can observe that maximum weight among items in a sequence is more suitable for of maximum weight of the sequence. Different from the traditional model, the upper-bounds of maximum weight values of candidates can be further tightened for mining. As mentioned above, the observation can be applied to our proposed algorithm to reduce unpromising candidates and speed up execution efficiency.

In this paper, we propose an improved upper-bound approach (abbreviated as *IUA*), which is based on the traditional upper-bound model [8] and the projection technique, for mining weighted sequential patterns from sequence datasets. In particular, an improved model is designed to tighten upper-bounds of maximum weights for patterns. In addition, an efficient pruning strategy is adopted to reduce the number of unpromising candidates in the mining process, thus avoiding unnecessary evaluation. The efficiency in finding weighted sequential patterns can thus be raised. Finally, the experimental results show the proposed algorithm executes faster than the *WSpan* algorithm.

2. Review of Related Works

Data mining involves applying specific algorithms to extract valuable patterns or rules from data sets in a particular representation, such as association rules [1][3]. In real-world applications, the transaction time of each transaction is usually recorded in databases. These transactions or records can then be listed as a time-series data (called sequence data) in their occurring time order. To handle such data, sequential pattern mining was first proposed to discover useful patterns from a set of sequence data [2]. In the field of sequential pattern mining, Agrawal *et al.* first proposed sequential pattern mining algorithms, such as *AprioriAll*, *AprioriSome*, and *DynamicSome* [2]. However, since the three algorithms adopted the level-wise technique, the algorithms had to execute multiple data scans to complete the sequential pattern mining task. Thus, many other algorithms were subsequently proposed to improve execution efficiency, such as *GSP* [7] and *PrefixSpan* [6].

In real-world applications, however, the importance of items is usually different. For example, manager may interest in buying product patterns with high-profit in a sequence database. However, the patterns may not be found by using traditional sequential pattern mining techniques. To handle this problem, Yun *et al.* proposed a new research issue, weighted sequential pattern mining [8], to find weighted sequential patterns from sequence databases. Weighted sequential pattern mining could assign different weight values to items by referring to factors, such as profits of items or users' preferences. Thus, the actual significance of a pattern could be easily recognized when compared with the traditional sequential pattern mining. In Yun *et al.*'s study, in addition, a weighted average function was designed to identify the weight value of a pattern in a sequence. However, the downward-closure property in traditional sequential pattern mining [2] was not kept in weighted sequential pattern mining [8]. To deal with the problem, Yun *et al.* was also designed an upper-bound method [8], which maximum weight among items in a sequence database was used as maximum weight of each sequence, to construct a new downward-closure property for weighted sequential pattern mining.

3. The Proposed Mining Algorithm

In this study, we propose a new projection-based mining algorithm to effectively handle the problem. The improved upper-bound strategy is first described below.

3.1 An Improved Upper-bound Strategy

In this study, an improved model is proposed to improve the traditional upper-bound model [8], thus tightening upper-bounds of maximum weight values for patterns in the mining process. In traditional upper-bound model, the maximum weight among items in a sequence dataset is regarded as the maximum weight of each sequence in the dataset. However, it can be observed that maximum weight among items in a sequence can also be regarded as the maximum weight of the sequence, and then the downward-closure property in weighted sequential pattern can be equally kept. Through the observation, it can be known that the traditional upper-bound model can be further improved by using the improved model.

3.2 The Pruning Strategy

In this section, a simple pruning approach based on the improved upper-bound model is designed to reduce the number of unpromising candidate patterns for mining. The major concept is that all weighted sequential patterns in a sequence dataset have to be included in weighted frequent upper-bound patterns [8]. Based on the improved model, all sub-patterns of a weighted frequent upper-bound pattern have to be also weighted frequent upper-bound patterns. According to the major principle, all the items appearing in the currently processed set of weighted frequent upper-bound patterns of r items ($WFUB_r$) are gathered as the pruning information of the strategy. Each item in a sequence is checked whether the item appears in the set of gathered items of the current prefix itemset. If it does, the item is kept in the sequence; otherwise, it is removed from the sequence.

3.3 The Proposed Projection-based Mining Algorithm with the Improved Model

The proposed weighted sequential pattern mining algorithm with an improved upper-bound model is then stated below.

INPUT: A set of items, each with a weight value; a sequence database *SDB*, in which each sequence includes a subset of items; a minimum weighted support threshold λ.

OUTPUT: A final set of weighted sequential patterns, *WS*.

STEP 1: For each y-th sequence Seq_y in *SDB*, find the maximal weight mw_y in Seq_y as:

$$mw_y = max\{w(i_{y1}), w(i_{y2}), \ldots, w(i_{yj})\},$$

where $w(i_{yj})$ is the weight value of each j-th item i_{yj} in Seq_y.

STEP 2: For each item i in *SDB*, do the following substeps.

(a) Calculate the maximal weight count upper-bound $mwub_i$ of the item i as:

$$mwub_i = \sum_{i \in Seq_y} mw_y \ ,$$

where mw_y is the maximal weight of each Seq_y in *SDB*.

(b) Calculate the actual weighted support count wsc_i of the item i as:

$$wsc_i = w(\,i\,) \times \sum_{i \in Seq_y} \delta(\,Seq_y\,)\,,$$

where $\sum_{i \in Seq_y} \delta(\,Seq_y\,)$ is the total number of the sequences containing i in *SDB*.

STEP 3: Calculate the minimum weighted support count value $mwsc$ of the threshold as:

$$mwsc = \lambda \times |\,SDB\,|\,,$$

where $|SDB|$ and λ are the total number of sequences in *SDB* and the minimum weighted support threshold, respectively.

STEP 4: Do the following substeps for each item i in the set of candidate 1-patterns.

(a) If the maximal weight upper-bound value $mwub_i$ of the 1-itemset i is larger than or equal to the minimum weighted support count value $mwsc$, put it in the set of weighted frequent upper-bound 1-patterns, $WFUB_1$.

(b) If the actual weighted support count value wsc_i of the 1-pattern i is larger than or equal to the minimum weighted support count value $mwsc$, put it in the set of weighted sequential 1-patterns, WS_1.

STEP 5: Acquire the items appearing in the set of $WFUB_1$, and put them in the set of possible items, PI_r.

STEP 6: For each y-th sequence Seq_y in *SDB*, do the following substeps.

(a) Get each item i located after x in Seq_y.

(b) Check whether i appears in PI_r or not. If it does, then keep the item i in Seq_y;

otherwise, remove the item *i*.

(c) If the number of items kept in the modified sequence Seq_y is less than the value (= *r* + 1), remove the modified sequence Seq_y from *SDB*; otherwise, kept it in *SDB*.

STEP 7: Process each item *i* in the set of $WFUB_1$ in alphabetical order by the following substeps.

(a) Find the relevant sequences including *i* in *SDB* and put the sequences in the set of projected sequences sd_i of *i*.

(b) Set $r = 1$, where *r* represents the number of items in the processed patterns.

(c) Find all the weighted sequential patterns with the item *i* as their prefix item by the *Finding-WS(i,* sd_i*, r)* procedure. Let the set of returned weighted sequential patterns be WS_i.

STEP 8: Output the set of weighted sequential patterns in all the WS_i.

After STEP 8, all the weighted frequent sequential patterns are found. The *Finding-WS(x,* sd_x*, r)* procedure finds all the weighted sequential patterns with the *r*-pattern *x* as their prefix patterns and is stated as follows.

The Finding-WS*(x,* sd_x*, r)* ***procedure:***

Input: A prefix *r*-itemset *x* and its corresponding projected sequences sd_x.

Output: The weighted frequent sequential patterns with *x* as its prefix sub-pattern.

PSTEP 1: Initialize the temporary pattern table as an empty table, in which each tuple consists of three fields: pattern, weighted upper-bound (*wub*) of the pattern, and actual weighted support count (*wsc*) of the pattern.

PSTEP 2: For each *y*-th sequence Seq_y in sd_x, do the following substeps.

(a) Get each item *i* located after *x* in Seq_y.

(b) Generate the (*r*+1)-pattern composed of the prefix *r*-pattern *x* and *i*, put it in the temporary set of patterns in Seq_y. If the pattern has not been appeared in the temporary set of patterns, then put it in the set; otherwise, omit the pattern.

(c) For each pattern in the set of temporary patterns, add the maximal weight value of the sequence Seq_y and actual weighted support value of the pattern in the corresponding fields in the temporary pattern table.

PSTEP 3: Do the following substeps for each pattern *P* in the set of candidate (*r*+1)-patterns.

(a) If the weighted upper-bound value wub_P of the (*r*+1)-pattern *P* is larger than or equal to the minimum weighted support count value *mwsc*, put it in the set of weighted frequent upper-bound (*r*+1)-patterns with the *x* as their prefix sub-patterns, $WFUB_{(r+1),\,x}$.

(b) If the actual weighted support count value wsc_P of the (*r*+1)-pattern *P* is larger than or equal to the minimum weighted support count value *mwsc*, put it in the set of weighted sequential (*r*+1)-patterns, $WS_{(r+1),\,x}$.

PSTEP 4: Acquire the items appearing in the set of $WFUB_{(r+1),\,x}$ of *x*, and put them in the set of possible items, PI_x.

PSTEP 5: Set $r = r + 1$, where *r* represents the number of items in the processed patterns.

PSTEP 6: For each *y*-th sequence Seq_y in sd_x, do the following substeps.

(a) Get each item *i* located after *x* in Seq_y.

(b) Check whether *i* appears in PI_x or not. If it does, then keep the item *i* in Seq_y; otherwise, remove the item *i* from Seq_y.

(c) If the number of items kept in the modified sequence Seq_y is less than the value (= *r* + 1), remove the modified sequence Seq_y from sd_x; otherwise, kept it in sd_x.

PSTEP 7: Process each pattern P in the set of $WFUB_r$ in alphabetical order by the following substeps.

(a) Find the relevant sequences including P from sd_x, and then put the sequences including P in the set of projected sequences sd_P of P.

(b) Find all the weighted sequential patterns with P as their prefix sub-patterns by the *Finding-WS(P, sd_P, r* + 1*)* procedure. Let the set of returned weighted sequential patterns be WS_P.

PSTEP 8: Return the set of weighted sequential patterns in all the WS_P.

4. Experimental Evaluation

In the experiments, since it was difficult to obtain the real datasets, a public *IBM* data generator [4] was adopted to generate the required experimental datasets. In addition, our purpose was to find weighted sequential patterns [8] from sequence datasets. Thus, we developed a simulation model, which was similar to that used in the issue of utility mining to generate the profits of the items, and then the profit of each item was normalized as a weight value in the range from 0.0 to 1.0.

Figure 1 showed the execution time comparisons of the proposed improved upper-bound approach (abbreviated as *IUA*) and the traditional weighted sequential pattern mining approach (named *WSpan*) [8] for the synthetic S8I4N3KD200K dataset with various minimum weighted support thresholds, varying from 2% to 0.2%.

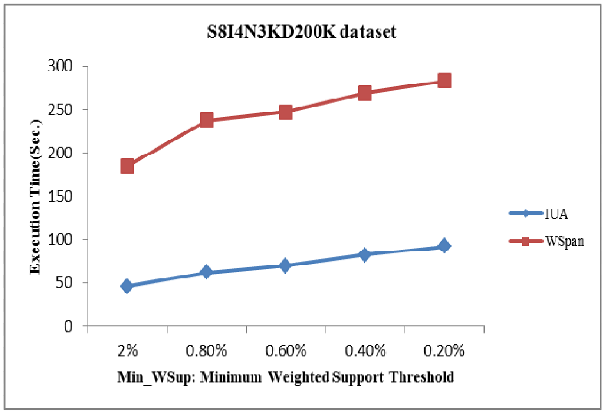

Figure 1. Execution time comparison of the two algorithms along with various thresholds.

It could be seen in the figure that the efficiency of the proposed *IUA* algorithm was better than that of the traditional *WSpan* algorithm when the minimum weighted support threshold decreased. The main reason is that the maximum weight among all items in a sequence was regarded as the sequence weight of the sequence. In addition, a pruning strategy was designed to avoid evaluation of unnecessary candidates. Thus, the proposed *IUA* algorithm executed faster than the traditional *WSpan* algorithm.

5. Conclusions

In this paper, we proposed an efficient projection-based algorithm (named *IUA*) for mining weighted sequential patterns from sequence databases. An improved strategy and a simple pruning strategy are designed to speed up execution efficiency. The experimental results on the synthetic datasets show the proposed *IUA* algorithm also outperforms the traditional *WSpan* algorithm in terms of pruning effect and execution efficiency.

References

1. R. Agrawal and R. Srikant, “Fast Algorithm for Mining Association Rules,” *The International Conference on Very Large Data Bases*, pp. 487-499, 1994.
2. R. Agrawal and R. Srikant, “Mining Sequential Patterns,” *The IEEE International Conference on Data Engineering*, pp. 3-14, 1995.
3. R. Agrawal, T. Imielinksi, and A. Swami, “Mining Association Rules between Sets of Items in Large Database,” *The ACM SIGMOD Conference*, pp. 207-216, 1993.
4. IBM Quest Data Mining Project, “Quest Synthetic Data Generation Code,” Available at (http://www.almaden.ibm.com/cs/quest/syndata.html).
5. Y. Liu, W. Liao, and A. Choudhary, “A Fast High Utility Itemsets Mining Algorithm,” *The Utility-Based Data Mining Workshop*, pp. 90-99, 2005.
6. J. Pei, J. Han, B. M. Asi, J. Wang, and Q. Chen, “Mining Sequential Patterns by Pattern-Growth: the PrefixSpan Approach,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 11, pp.1424-1440, 2004.
7. R. Srikant and R. Agrawal, “Mining Sequential Patterns: Generalizations and Performance Improvements,” *The 5th International Conference Extending Database Technology*, pp. 3-17, 1996.
8. U. Yun and J.J. Leggett, “WSpan: Weighted Sequential Pattern Mining in Large Sequence Databases,” *The 3rd International IEEE Conference on Intelligent Systems*, 2006.
9. U. Yun, “New Framework for Detecting Weighted Sequential Patterns in Large Sequence Databases,” *Knowledge Based Systems*, Vol. 21, No. 2, pp. 110-122, 2008.