Implementation of Sequential Pattern Mining with Item Intervals

Kay Zar Kyaw, Ni Ni Hla
University of Computer Studies, Yangon
kayzarkyaw@gmail.com

Abstract

Sequential Pattern mining is an important data mining field with wide range of applications that can extract frequent sequences while maintaining their order. It is important to identify item intervals of sequential patterns extracted by sequential pattern mining. There are two approaches for integration of item intervals with sequential pattern mining; constraint-based mining and extended sequence-based mining. This paper presents the combination of those two item interval approaches. PrefixSpan algorithm is used to find the frequent sequence patterns from the sequence database. PrefixSpan algorithm overcomes the problems of Apriori-based algorithms since it avoids the candidate generation and multiple database scanning time. Moreover, prefix-projectiong substantially reducest the size of projected databases and leads to efficient processing.

Keywords: Data Mining; Web Mining; Frequent Patterns, Sequential Pattern, Apriori, AprioriAll, PrefixSpan, Pattern Growth Method

1. Introduction

Sequential pattern mining, which discovers frequent subsequences as patterns in a sequence database, is an important data mining problem with broad applications, including the analyses of customer purchase behavior, Web access patterns, scientific experiments, disease treatments, natural disasters, DNA sequences, and so on.

There are many sequential pattern mining algorithms have been proposed. Previous approaches consider only the item occurrence order, but do not consider the item intervals between successive items. Thus, it is impossible to identify the item intervals between successive items extracted as frequent sequential patterns. There are two types of intervals to handle, Item Gap and Time Interval. It is useful to be able to distinguish these customers' actions to understand not only what events will follow, but also when these events will occur.

To handle two types of intervals, there are two approaches for integration of item intervals with sequential pattern mining. They are constraint-based mining and extended sequence-based mining. Constraint-based mining algorithms extract frequent sequences that satisfy both user-specified minimum support constraints and user-specified constraints. For example, when users set the maximal time interval to 1 day, they extract frequent sequences whose items occur within 1 day.

Even though constraint-based mining algorithms escape extraction of sequences for too long intervals, it is difficult for users to specify an optimal constraint related to item interval. Extended sequence-based mining algorithms extract frequent sequences consisting of the same items but satisfying different item interval constraints by one time execution. These algorithms extract frequent sequential patterns with item intervals by converting item intervals to pseudo items. However, they may extract meaningless patterns, such as sequences with too long item intervals. To overcome above problems, this paper presents the combination of constraint-based mining and extended sequence-based mining.

The main problem in frequent sequence pattern mining area is the long processing time. Apriori-like method adopts a multiple-pass, candidate-generation-and-test approach in sequential pattern mining. This paper presents efficient sequential pattern mining method, called PrefixSpan (Prefix-projected Sequential patterning mining). It examines only the prefix subsequences and projects only their corresponding postfix subsequences into projected databases. In each projected database, sequential patterns are grown by exploring only local frequent patterns.

The organization of this paper is as follows: Section 2 presents the related work of the sequential pattern mining approaches. Section 3 illustrates the Sequential pattern mining and PrefixSpan algorithm. Section 4 describes the proposed system design and Section 5 is the implementation of the system and its results. In Section 6, there is conclusion of the paper.

2. Related Work

Sequential pattern mining, which is one of the most important of data mining technologies, extracts patterns that appear more frequently than a user-specified minimum support while maintaining their item occurrence order [4].

There are several approaches contributed to the efficient mining of sequential patterns or other frequent patterns in time related data. Almost all of the previously proposed methods for mining sequential patterns and other time-related frequent patterns are Apriori-like, i.e., based on the Apriori property proposed in association mining [4], which states the fact that any super-pattern of a nonfrequent pattern cannot be frequent.

A typical Apriori-like method such as GSP [5] adopts a multiple-pass, candidate-generation-and-test approach in sequential pattern mining. The first scan finds all of the frequent items which form the set of single item frequent sequences. Each subsequent pass starts with a seed set of sequential patterns, which is the set of sequential patterns found in the previous pass. This seed set is used to generate new potential patterns, called candidate sequences. Each candidate sequence contains one more item than a seed sequential pattern, where each element in the pattern may contain one or multiple items. The number of items in a sequence is called the length of the sequence. So, all the candidate sequences in a pass will have the same length. The scan of the database in one pass finds the support for each candidate sequence. All of the candidates whose support in the database is no less than min support form the set of the newly found sequential patterns. This set then becomes the seed set for the next pass. The algorithm terminates when no new sequential pattern is found in a pass, or no candidate sequence can be generated.

The main bottle-neck of an Apriori-based sequential pattern mining method come from candidate generation and test [2]. This paper presents an efficient sequential pattern mining method, called PrefixSpan. It examines only the prefix subsequences and created smaller projected databases for each prefix. PrefixSpan minds the complete set of patterns and is efficient and runs faster than both Apriori-based GSP and FreeSpan.

3. Sequential Pattern Mining

Sequential pattern mining is to discover frequent transaction patterns such that the presence of a set of items is followed by another item in the time-stamp ordered transaction set. Let $\{i_1, i_2, i_3, \ldots, i_m\}$ be an itemset i where an itemset is a non-empty set of items and ik is an item. A sequence s is denoted as $<s_1, s_2, s_3, \ldots, s_n>$ where s is an order list of itemsets and s_j is an itemset whose items are purchased by a customer at the same transaction-time. For an item i_k, it can appear only once in s_j, but can appear multiple times in s_i and s_j with different transaction-time.

All the transactions of a customer, ordered by increasing transaction-time, is a customer-sequence. The support count for a customer-sequence is defined as the fraction of total customers who support this sequence. Each sequence satisfying a certain minimum support threshold (user-specified) is called a large sequence. Given a transaction database D and a minimum support threshold, the problem of mining sequential patterns can be defined as finding the maximal large sequences among all the sequences with support count greater than or equal to. Each found maximal large sequence represents a sequential pattern. In addition, time constraints will be considered when finding sequential patterns, and this makes the found sequence patterns more useful.

4. PrefixSpan Algorithm

PrefixSpan is a fast sequential pattern mining algorithm which extracts frequent sequences with depth-first search by executing sequence database projection operations recursively. PrefixSpan consists of sequence database projection operation, which is the most important process in the algorithm.

It is an approach of pattern growth method. Pattern growth is a method of frequent-pattern mining that does not require candidate generation. It is based on the FP-Growth algorithm for frequent pattern mining. PrefixSpan uses prefix projection to mine the complete set of frequent sequential patterns. The general idea of this approach is as follows:

- It follows the frequent single items, then compresses this information into a frequent-pattern tree or FP-tree.
- The FP-tree is used to generate a set of projected databases, each associated with one frequent item.
- Each of these databases is mined separately.

The algorithm builds prefix patterns, which it concentrates with suffix patterns to find frequent patterns, avoiding candidate generation. Prefix spanning algorithm extends the pattern-growth approach to instead mine sequential patterns. Figure

1 presents how prefixSpan is built for a sequence database.

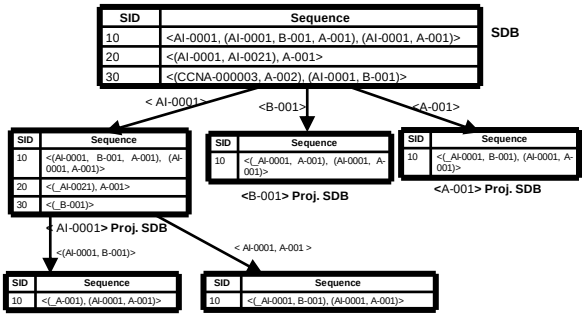

Figure 1: Building PrefixSpan

5. Proposed System

The proposed system extracts frequent interval extended sequences based on Prefix Span Algorithm. It will present the generalization of two types of constraints and extended sequence approach. It is able to substitute all types of conventional sequential pattern mining algorithms with item intervals. It will extend the sequential database projection operation to handle both interval extended sequence and item interval constraints.

This paper presents sequential pattern mining based on PrefixSpan algorithm with item intervals". It includes following points;

- a capability to handle two kinds of item interval measurement, item gap and time interval,
- a capability to handle extended sequences which are defined by inserting items based on the interval itemization function, and

The proposed approach is able to substitute all types of conventional sequential pattern mining algorithms with item intervals. Figure 2 presents the overview process flow of the proposed system.

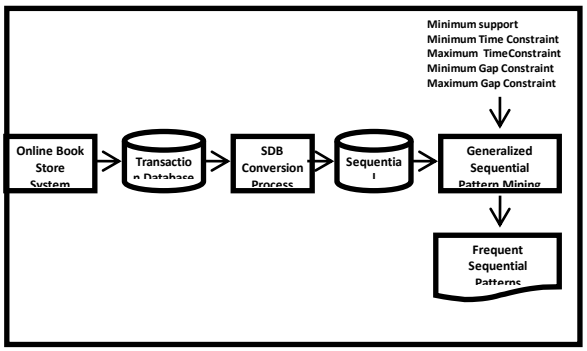

Figure 2: Overview process flow of proposed system

5.1. Types of Item-Intervals

Item intervals are represented in two ways; item gap and time interval.

- Item gap is defined as the number of items between successive items, and
- Time interval is defined as the length of time between the occurrence times of successive items.

Generally, item gap measurement has the advantage of being applicable to datasets the items of which have no occurrence time information. On the other hand, time interval measurement has the advantage of allowing the extraction of frequent sequential patterns with time interval information. When applied to datasets for the items of which occurrence time information is available, sequential pattern mining with time intervals can extract more precise patterns than sequential pattern mining with item gaps.

5.2. Approaches for Counting Support Values

There exist two approaches to count support value; the item constraint approach and extended sequence approach.

- The item constraint approach involves extraction of sequences satisfying not only a user-specified minimum support constraint, but also user-specified constraints, such as maximum / minimum item intervals.
- The extended sequence approach extends sequences by inserting pseudo items which represent item intervals. After extending the original sequences, it extracts frequent sequential patterns from them.

The proposed system use combination of those two approaches. Constraint-based mining approach avoids the extraction of sequences with non-interest time intervals such as too long intervals it has setbacks in that it is difficult to specify optimal constraints related to item interval, and users must re-execute constraint-based algorithms with changing constraint values. On the other hand, extended sequence-based mining approach does not need to specify constraints and re-execute. Since extended sequence-based mining approach cannot adopt any constraints based on time intervals, it may extract meaningless patterns, such as sequences with too long item intervals. This means these two approaches have not only advantages but also disadvantages. To solve this problem, in this paper, this system presents combination of those two approaches.

6. System Implementation

This system is implemented as web based online book store system. It is developed using Microsoft

Visual Studio .Net 2008. ASP .Net C# is used to implement the system and Microsoft Access 2003 is used to store the transaction data, book information and user profiles. Sequence database is created based on user buying pattern.

Transaction sales data are stored in the relational database. Transactions are generated by selling items online. Then transaction data is converted into sequence database. PrefixSpan with Item Interval is applied to sequence database to generate the frequent sequence patterns.

Transaction database contains sales information and customer profile information to build the sequence database. The database model used in this system has been shown in Figure 3.

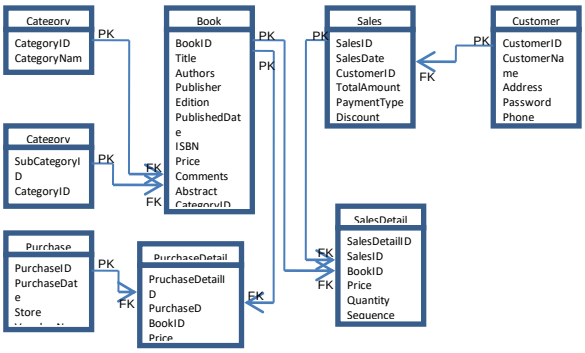

Figure 3: Database Model of the System

6.1. Process Flow

The process flow of the system is as follows:

- It first reads the transaction database from online book store system.
- Then convert it into sequential database by grouping transactions of the same user based on transaction time.
- Then prefix items are computed and prune items whose support is less than minimum support.
- Prepare post-fix items according to item gap and time gap constraints.
- Count sequences for generating frequent sequence patterns.

Figure 4 presents the process flow of the system.

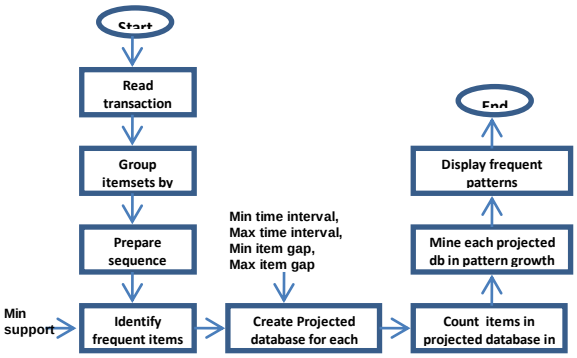

Figure 4: Process Flow of the System

6.2 Implementation of PrefixSpan Algorithm

Input: A sequence database S, and the minimum support threshold min_sup
Output: The complete set of sequential patterns
Method: Call PrefixSpan(<>, 0, S).
Subroutine PrefixSpan(a, l, S|a)
Parameters: a: a sequential pattern; l: the length of a; S|a: the a-projected database, if a != <>; otherwise, the sequence database S.
Method:
1. Scan S|a once, find the set of frequent items b such that
(a) b can be assembled to the last element of a to form a sequential pattern; or
(b) <b> can be appended to a to form a sequental pattern.
2. For each frequent item b, append it to a to form a sequential pattern a', and output a';
3. For each a', construct a'-projected database S|a', and call PrefixSpan (a', l+1, S|a')

6.3 Implementation of Projection Algorithm with four user Constraints

INPUT ISDB, min_sup, C_1, C_2, C_3, C_4
OUTPUT R(=frequent sequences satisfying constraints)
METHOD
1) Set is as ϕ.
2) Set R as ϕ.
3) Scan ISDB, and find frequent items with higher than min sup. For all frequent items i,
a) Define is =< (0; i) >, then R = {R, is}.
b) Execute R = projection(ISDB|is, R, min_sup, C_1, C_2, C_3, C_4).
4) Output R.

The projection routine is shown below. It computes level 2 or later projection.
INPUT ISDB|is,R, min_sup, C_1, C_2, C_3, C_4
OUTPUT R
METHOD

1) Scan ISDB|is to find all pairs of item and its itemized interval, denoted as (Δt, i), that satisfy min sup, C_1, and C_2.
2) Define is =< is; (Δt, i) >.
3) Check is whether satisfies C_4 or not.
4) Only when is satisfies C_4,
a) Execute R = projection(ISDB|is, R, min_sup, C_1, C_2, C_3, C_4).
b) When is satisfies C_3, R = {R, is}.
5) Return R.

7. Experimental Result

This system is tested on the computer with Processor Intel(R) Pentium(R) Dual CPU 2.20GHz, Memory 2 GB of RAM. 200 user profiles are created and there are 650 transactions and 1750 transaction items for all 200 users. Therefore we have got 200 sequences in the sequence database. The experimental results show that PrefixSpan algorithms outperforms significantly over the Apriori-based algorithms. Moreover, frequent sequence patterns with item intervals, produce higher accuracy than traditional sequential pattern mining algorithms. Table 1 presents the processing time comparison with Apriori-based GSP algorithm.

Table 1: Experimental Results for Processing Time in milliseconds

No.	Min-sup	Processing Time for PrefixSpan	Processing Time for GSP
1	2	125	2300
2	3	114	2150
3	4	103	1980

Table 2 presents the accuracy comparison with PrefixSpan without item intervals.

Table 2: Accuracy for PrefixSpan with Item interval and PrefixSpan without Item interval

No.	Min-sup	PrefixSpan with Item Interval	PrefixSpan Alone
1	2	92.20%	89.24%
2	3	93.24%	89.76%
3	4	93.13%	90.09%

8. Conclusion

This paper presents the discovering of frequent sequential pattern from the transaction database. PrefixSpan algorithm is used to extract the frequent sequence patterns. The experimental results show it outperforms over the Apriori-based GSP algorithm. Moreover this paper describes the item intervals to get the more relevant frequent sequences, producing higher accuracy.

9. References

[1] D.Y. Chiu, Y. H. Wu and A.L.P Chen, "An Efficient Algorithm for Mining Frequent Sequences by a New Strategy without Support Counting", MOE Program for Promoting Academic Excellence of Universities, 2003.

[2] J. Han, J. Pei, and Y. Yin "Mining frequent patterns without candidate generation". In Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), pages 1–12, Dallas, TX, May 2000.

[3] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns by Pattern-Growth: Methodology and Implications", ACM SIGKDD, December 2000.

[4] J. Pei, J. Han, B. Mortazavi-Asl and H. Pinto, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Natural Sciences and Engineering Research Council of Canada (grant NSERC-A3723), 2001.

[5] J. Pei and J. Han, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach", In the proceeding of IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 11, November 2004.

[6] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94), pages 487–499, Santiago, Chile, Sept. 1994.

[7] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In Proc. 5th Int. Conf. Extending Database Technology (EDBT'96), pages 3–17, Avignon, France, Mar. 1996.