An Efficient Algorithm for Mining Frequent Sequential Patterns and Emerging Patterns with Various Constraints

C K Bhensdadia, Y P Kosta

***Abstract*— In many cases, sequential pattern mining still faces tough challenges in both effectiveness and efficiency. On the one hand, there could be a large number of sequential patterns in a large database. A user is often interested in only a small subset of such patterns. Presenting the complete set of sequential patterns may make the mining result hard to understand and hard to use. On the other hand, although efficient algorithms have been proposed, mining a large amount of sequential patterns from large data sequence databases is very expensive task. If we can focus on only those sequential patterns interesting to users, we may be able to save a lot of computation cost by those uninteresting patterns.**

Many types of constraints can be pushed in sequential pattern mining like item constraint, aggregate constraint, length constraint, gap constraint, duration to enhance the performance.

***Index Terms*— Sequential Pattern, Constraints.**

I. INTRODUCTION

In this section of paper experimental results on the performance of C-PrefixSpan (constraint PrefixSpan) algorithm is being shown. Our performance study shows that C-PrefixSpan is an efficient and scalable method. It outperforms the original Prefixspan in many cases. Here, different experiments conducted to measure the relative performance of the proposed C-PrefixSpan for mining constraint based sequential patterns compared to the previous approach PrefixSpan.

We use synthetic dataset to test the methods. In the synthetic dataset, sequences are generated using a standard procedure described in [2]. The same data generator has been used in most studies on sequential pattern mining, such as [11, 6]. Limited by space, only a set of experiments on few datasets are reported and discussed here. In dataset 1, C10TS4T2.5N10, and dataset 2, C10TS6T2.5N10, the numbers of customers are set to 10,000, the average number of transactions are set to 4,000 and 6,000 respectively and other parameters remains same.

Manuscript received October 09, 2011.

C K Bhensdadia, Department of Computer Engineering, Dharmsinh Desai Institute of Technology (Faculty of Technology),Dharmsinh DesaiUniversity, Nadiad, -388 421, INDIA (e-mail: ckbhensdadia@yahoo.co.in).

Y P Kosta, Marwadi Education Foundation's Group of Institutions, Rajkot-360 003, INDIA (e-mail: ypkosta@yahoo.com).

II. OBJECTIVE AND SCOPE OF WORK

Objective

Objective of the project work is to develop an efficient algorithm which discovers the sequential patterns of user's interest with given constraints. The proposed algorithm modifies traditional sequential pattern mining algorithm PrefixSpan (Pattern-Growth based), so that, it also considers additional constraints, the aggregate constraint (with five number summaries), the length constraint and the item constraint to discover the interesting patterns.

Need for the Research

In a real-life situation the environment may change constantly and user's behaviors may also change over time. Therefore, the user's behavior is not necessarily the same as the past ones and a pattern that occurs frequently in the past may never happen again in the future. If only frequency is considered, we may discover patterns which are important for the past but not for now and the future. So this motivates to consider more constraints and design new sequential pattern mining methods in pattern discovery.

Contribution towards the knowledge

"Discovery consists of seeing what everybody has seen and thinking what nobody has thought." Data mining is towards an effective and efficient tool for discovery. By mining, we can see the patterns hidden behind the data more accurately, more systematically and more efficiently.

This thesis is devoted to the sequential pattern mining, which belongs to one of the most frequently used data mining technique. A real universal challenge is to find actionable knowledge from a large amount of data. In this dissertation, we have focus on the problem of mining Sequential patterns efficiently and effectively, and enhanced a new class of pattern growth algorithm i.e. PrefixSpan. This algorithm can reduce the number of uninteresting pattern and increase the discovery capacity with a small value of minimum support and minimum confidence.

The C-Prefixspan's performance is better when average number of transactions per sequence increases. The C-Prefixspan algorithm generates more patterns when average number of transactions per sequence increases but these patterns are comparatively very less and more valuable than Prefixspan algorithm. Most of the existing methods, however, focus on the concept of frequency in the sequence database and did not consider other factors like concepts of length, gap and aggregates. To apply these factors in sequential pattern mining process the Prefixspan algorithm which uses prefix projected pattern growth approach is modified to form Constrained Prefixspan algorithm. From the results presented in below section, it is evident that as database size increases Constrained Prefixspan algorithm generates very less but more valuable patterns than Prefixspan algorithm.

Focus of the Project

Given a sequence database and a *min support* threshold, the problem of sequential pattern mining is to find the complete set of sequential patterns in the database. To improve the performance of sequential pattern mining, a PrefixSpan algorithm was developed.

The *PrefixSpan* (Prefix-projected Sequential pattern mining) algorithm presented by Jian Pei, Jiawei Han and Helen Pinto[1] representing the pattern-growth methodology, which finds the frequent items after scanning the sequence database once. The PrefixSpan algorithm successfully discovered patterns employing the divide-and-conquer strategy. It uses depth first method for searching. It requires single database scan, whereas other candidate generation methods require multiple database scanning. It resolves difficulties at mining long sequential patterns by using prefix projected database.

Literature review

In this paper, we consider the problem of clustering association rules of the form A^B => C. where the LHS attributes (A and B) are quantitative and the RHS attribute (C) is categorical. We define segmentation as the collection of all the clustered association rules for a specific value C of the criterion attribute.

The algorithm introduce here is to cluster association rules using a 2D grid solves only a part of the overall problem. The association rule engine is a special-purpose algorithm that operates on the binned data. It applies our BitOp algorithm to this grid.

Mining Quantitative Association Rules in Large Relational Tables

This paper introduces the problem of mining association rules in large relational tables containing both quantitative and categorical attributes. It gives an algorithm for mining such quantitative association rules.

Mining Optimized Association Rules with Categorical & Numeric Attributes

This paper focuses on the Optimized Association Rules and they are permitted to contain uninstantiated attributes & the problem is to determine instantiation such that either the support or confidence of the rule is maximized. In this paper they generalized the optimized association rules problem in three ways: 1) association rules are allowed to contain disjunction over uninstantiated attributes, 2) association rules are permitted to contain an arbitrary number of in instantiated attributes, & 3) uninstantiated attributes can be either categorical or numeric.

Quantitative Association Rules Mining Methods with Privacy-preserving

This paper considers the different size of quantitative attribute values and categorical attribute values in databases, here it presents two quantitative association rules mining methods with privacy-preserving respectively, one bases on Boolean association rules, which is suitable for the smaller size of quantitative attribute values and categorical attribute values in databases; the other one bases on partially transforming measures, which is suitable for the larger ones. To each approach, the privacy and accuracy are analyzed, and the correctness and feasibility are proven by experiments.

DISTANCE BASED CLUSTERING OF ASSOCIATION RULES

This paper focuses on the Association rule mining is one of the most important procedures in data mining. In this paper, they present a new normalized distance metric to group association rules. Based on these distances, an agglomerative clustering algorithm is used to cluster the rules. Also the rules are embedded in a vector space by multi-dimensional scaling and clustered using a self organizing feature map. The results are combined for visualization.

Mining Quantitative Association Rules under Inequality Constraints

This paper presents how to integrate the inequality constraints into the mining process and reduce the number of database scanning. The algorithm they present generates the large item sets by building the expression tree and prunes away the undesired one by checking the acceptance range.

DEFINING INTERESTINGNESS FOR ASSOCIATION RULES

This paper provides an overview of some existing measures of interestingness and we will comment on their properties. In general, interestingness measures can be divided into objective and subjective measures. Objective measures tend to express interestingness by means of statistical or mathematical criteria, whereas subjective measures of interestingness aim at capturing more practical criteria that should be taken into account, such as unexpectedness or action ability of rules. This paper only focuses on objective measures of interestingness.

III. PROBLEM STATEMENT

In many cases, sequential pattern mining still faces tough challenges in both effectiveness and efficiency. On the one hand, there could be a large number of sequential patterns in a large database. A user is often interested in only a small subset of such patterns. Presenting the complete set of sequential patterns may make the mining result hard to understand and hard to use. On the other hand, although efficient algorithms have been proposed, mining a large amount of sequential patterns from large data sequence databases is very expensive task. If we can focus on only those sequential patterns interesting to users, we may be able to save a lot of computation cost by those uninteresting patterns.

Constraint based sequential pattern mining extracts the sequential patterns which are of user's interest. For effectiveness and efficiency considerations, constraints are essential in many sequential pattern mining applications. Many types of constraints can be pushed in sequential pattern mining like item constraint, aggregate constraint, length constraint, gap constraint, duration constraint, regular expression constraint, time constraint etc.

Proposed approach works on applying constraint in existing algorithm to improve efficiency. Proposed C-PrefixSpan sequential pattern mining approach discovers those sequential patterns from sequence database which are frequent and which also satisfy Aggregate, Length and Gap constraint.

IV. PROPOSED SYSTEM

The proposed algorithm improves the performance of finding frequent sequences by applying modified candidate generation and support counting approach. Traditional sequential pattern mining only distinguishes whether a pattern appears or not, while RFM pattern mining approach not only determines the existence of a pattern but also checks whether it satisfies the recency and the monetary constraints. Some key concepts required to understand the proposed algorithm is as follows:

1. Recency Constraint: Recency constraint is specified by giving a recency minimum support (*r_minsup*), which is the number of days away from the starting date of the sequence database. For example, if our sequence database is from 27/12/2007 to 31/12/2008 and if we set *r_minsup* = 200 then the recency constraint ensures that the last transaction of the discovered pattern must occur after 27/12/2007+200 days. In other words, suppose the discovered pattern is < (*a*), (*bc*)>, which means "after buying item a, the customer returns to buy item b and item c". Then, the transaction in the sequence that buys item b and item c must satisfy recency constraint.

2. Monetary Constraint: Monetary constraint is specified by giving monetary minimum support (*m_minsup*). It ensures that the total value of the discovered pattern must be greater than *m_minsup*. Suppose the pattern is <(*a*), (*bc*)>. Then we can say that a sequence satisfies this pattern with respect to the monetary constraint, if we can find an occurrence of pattern <(*a*), (*bc*)> in this data sequence whose total value must be greater than *m_minsup*.

3. Frequency Constraint: Frequency constraint is specified by giving frequency minimum support (*f_minsup*). The frequency of a pattern is the percentage of sequences in database that satisfy the recency constraint and monetary constraint. And a pattern could be output as an *RFM*-pattern if its frequency is greater than *f_minsup*. By setting these three constraints properly, we can discover an *RFM*-pattern like this, "30% customers who recently bought a computer would return to buy the scanner and microphone, and the total value will exceed Rs. 50,000".

4. f-pattern(LI_k^f), rf-pattern(LI_k^{rf}), rfm-pattern(LI_k^{rfm}) of length k: Let $B = <I_1 I_2 ... I_S>$ be a sequence of itemsets. If the percentage of data sequences in database containing *B* as a subsequence, called *f*-support, is no less than *f_minsup*, *B* is called an *f*-pattern. *B* is called an *rf*-pattern if the percentage of data sequences in database containing *B* as a recent subsequence (which satisfies recency constraint), called *rf*-support, is no less than *f_minsup*. Finally, *B* is called an *rfm*-pattern if the percentage of data sequences in database containing *B* as a recent monetary subsequence (which satisfies recency and monetary constraints), called *rfm*-support, is no less than *f_minsup*.

5. Candidate Pattern: Individual sequence of a candidate sequence is known as candidate pattern.

6. Representation of Data Sequence: Data-sequence *A* is represented as $<(a_1, t_1, m_1), (a_2, t_2, m_2), \ldots, (a_n, t_n, m_n)>$, where (a_j, t_j, m_j) means that item a_j is purchased at time t_j with total value m_j, $1 \le j \le n$, and $t_{j-1} \le t_j$ for $2 \le j \le n$. In the data-sequence, if items occur at the same time, they are ordered alphabetically.

V. PROPOSED ALGORITHM

The Proposed Algorithm

Input: Data-sequence database
The threshold of support given by the user (f_minsup)
The threshold of recency given by the user (r_minsup)
The threshold of monetary given by the user (m_minsup)

Output: The set of all large RFM-patterns

Method:

1. Scan the sequence database to count f-support,rf-support and rfm-support for each itemset.
2. Generate 1-frequent patterns (LI_1^f, LI_1^{rf}, LI_1^{rfm}) which satisfies the user specified support threshold (f_minsup) using apriori process.
3. if k=2 then merge frequent patterns (LI_1^f, LI_1^{rf}) to generate candidate sequence (CI_2).
4. if k>2 then merge frequent patterns (LI_{k-1}^{rf},LI_{k-1}^{rf}) which have same k-2 postfix to generate candidate sequence (CI_k).
5. Build inverse candidate tree by inserting the itemsets in all candidate patterns of candidate sequence (CI_k) into an empty tree in reverse order.

6. Take one by one data sequences from sequence database and traverse the tree to match different subsequences of data sequence with all candidate patterns of current candidate sequence.
7. If matched subsequence satisfies both recency(r) and monetary(m) constraint then increase rf-support and rfm-support for current candidate pattern by one.
8. Eliminate candidate patterns that have small support count (rf-support and rfm-support) compared to user specified support threshold (f_minsup) and find frequent patterns (LI_k^{rf} , LI_k^{rfm}).
9. Repeat through step 4 until no more frequent pattern (LI_k^{rf}) is found.

Output the set of all large RFM-patterns.

VI. SIMULATION, EXPERIMENTAL SETUP AND IMPLEMENTATION

Implementation of C-Prefixspan Approach

The C-Prefixspan approach produces sequential patterns which satisfies Length, Aggregate, Frequency and Gap constraints also minimum confidence. The original Prefixspan algorithm only applies frequency constraints to discover sequential patterns from sequence database. To apply above constraints in sequential pattern mining process changes are done in original Prefixspan algorithm.

Following figures shows the snapshots of output screen:

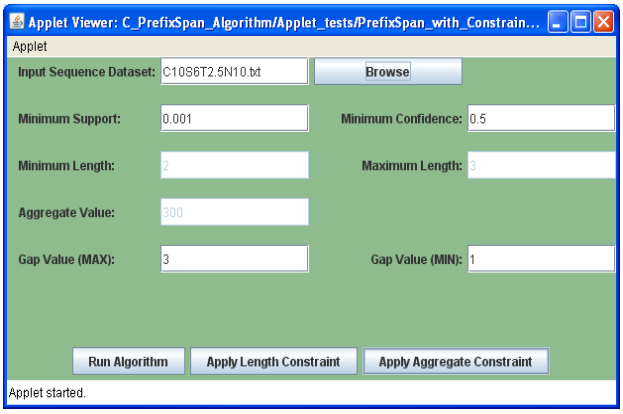

Figure 3.1: Main Screen of Project Output

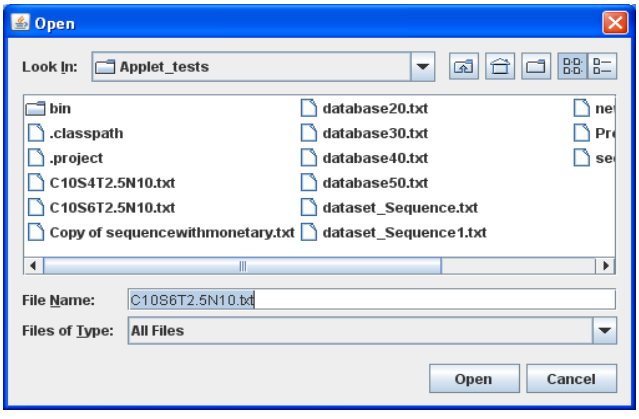

Figure 3.2: Selection of Dataset to operate on

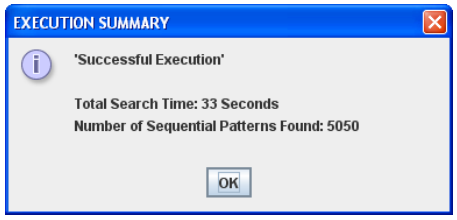

Figure 3.3: Execution Summary

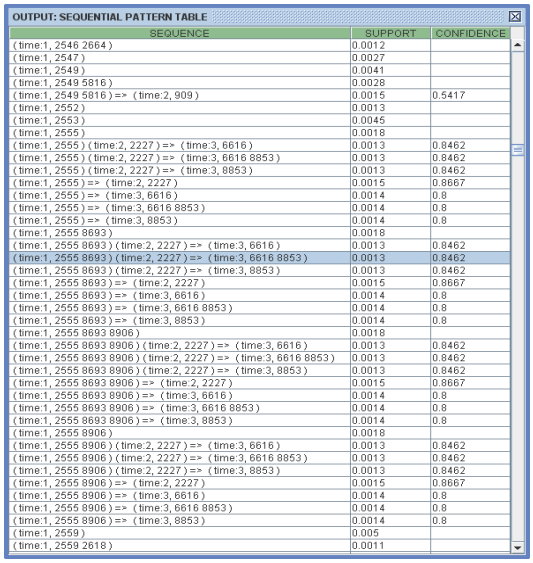

Figure 3.4: Sequential Patterns which satisfies Gap and Frequency Constraints

3.2 Applying Length Constraint

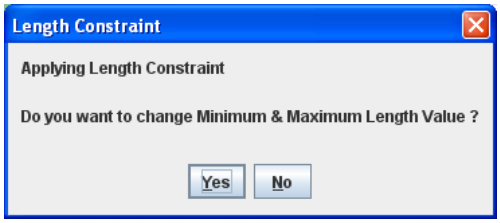

Figure 3.5: change in minimum and maximum length value

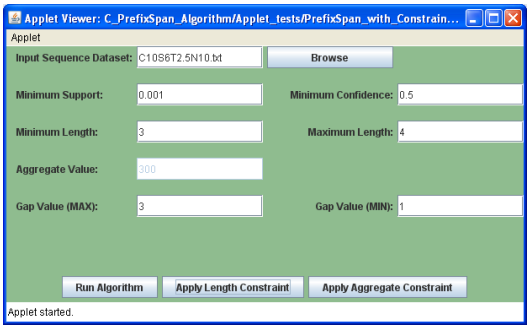

Figure 3.6: Apply Length Constraint

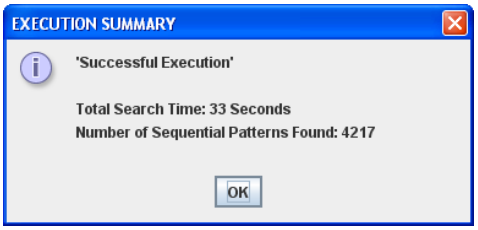

Figure 3.7: Execution summary after applying length constraint

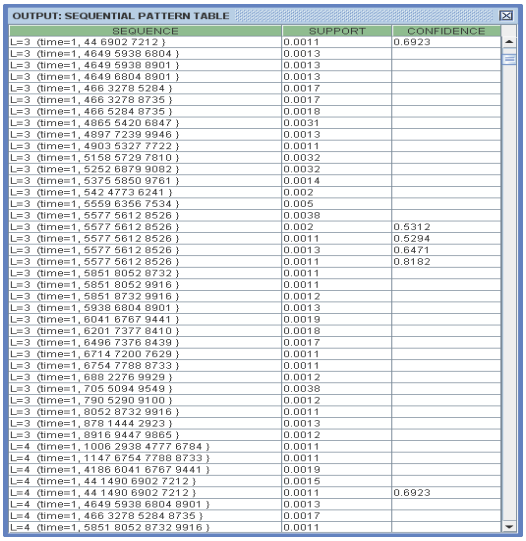

Figure 3.8: Sequential Patterns satisfies Length constraint

3.3. Applying Aggregate Constraint

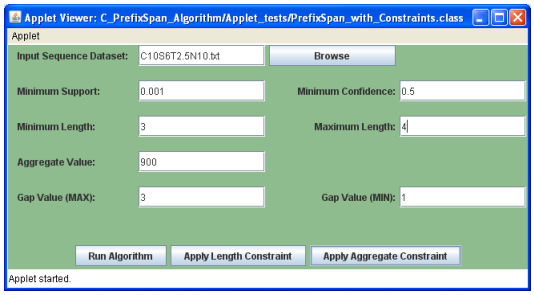

Figure 3.9: Apply Aggregate Constraint

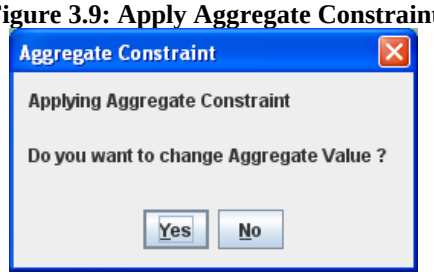

Figure 3.10: change in aggregate value

VII. GSP AND PREFIXSPAN ALGORITHM

Design Modification

In the proposed approach GSP algorithm was introduced to discover RFM sequential patterns from sequence database. But for implementation purpose the design is modified so that instead of GSP algorithm, the Prefixspan algorithm which is based on prefix projected pattern growth approach is proposed.

Comparison between GSP and Prefixspan Algorithms

GSP ALGORITHM	PREFIXSPAN ALGORITHM
Apriori based	Pattern Growth based
Uses Candidate generation and test approach	Uses Projected database concept (doesn't use candidate generation approach)
Requires Multiple database scan	Requires single database scan
To generate long sequential pattern, large number of candidates are generated	To generate long sequential pattern the number of projected databases to be generated are comparatively less.
Generates some candidates which doesn't have any existence in sequence database	Never generates any prefix which is not present in sequence database
Not good for those applications where low support thresholds are used	Good for those applications where low support thresholds are used
Performance is poor than Prefixspan algorithm	Performance is better than GSP algorithm

Comparison between GSP and PrefixSpan Algorithms

The modified approach applies following constraints in sequential pattern mining process.

1. Gap Constraint: A gap constraint is defined in sequence databases where each transaction in every sequence has a timestamp. It requires that the sequential patterns in the sequence database must have the property such that the timestamp difference (difference of days) between every two adjacent transactions in a discovered sequential pattern must not be greater than given gap.

2. Compactness Constraint: A compactness constraint requires that the sequential patterns in the sequence database must have the property such that the time-stamp difference (difference of days) between the first and the last transactions in a discovered sequential pattern must not be greater than given period.

3. Recency Constraint: A recency constraint is defined as the last transaction of the discovered pattern must be greater than or equal to recency minimum support.

4. Frequency Constraint: The frequency constraint is defined as each discovered pattern must satisfy minimum support.

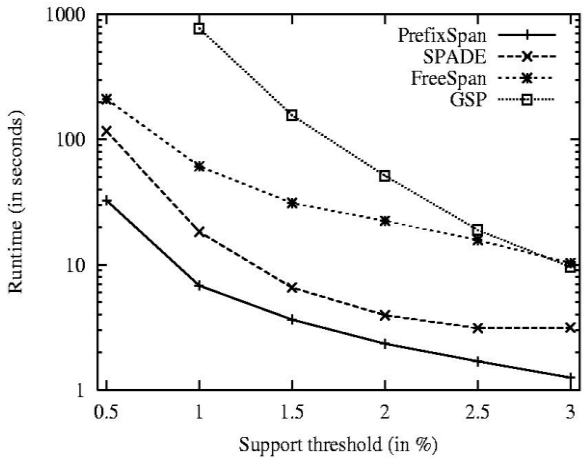

Figure 3.11 Performance on Data Set C10T8S8I8

Justification for choosing the PrefixSpan to modify:

Following graph shows the running time of various Sequential Pattern Mining algorithms when various datasets are used for mining:

Analysis of data from literature:

Dataset format

For the purpose of implementing the Constrained Prefixs pan algorithm for finding sequential patterns, the dataset generated by illimine synthetic data generator is used. The dataset is provided in text format.

Parameter	Description
|C|	Number of Customers
|S|	Average number of transactions per sequence
|T|	Average number of items per transaction
|N|	Number of distinct items

Input file format(.txt)

- ✓ PrefixSpan takes binary input file format. Each integer is represented by 4 bytes, in the endian format of the machine.
- ✓ Usually a sequence database consists of a series of sequences, and each sequence is composed of several transactions (also called events) sorted in time ascending order. Each distinct item in the database should be assigned an itemID numbered beginning from 0, and the items in each transaction should be sorted in ascending order according to their itemID.
- Also, the sequential database being mined is in binary format, each sequence ends with a -2 and every transaction is followed by a -1. Here is a **sample sequence:**

2 5 7 -1 1 2 -1 3 9 -1 4 -1 -2

This sequence equals <(2 5 7)(1 2)(3 9)(4)>, there are totally 4 transactions

(i.e., events), and each transaction contains 3, 2, 2, 1 items respectively.

The example of data generated by Illimine synthetic data generator is given below:

2 3 2635 8073 1933 1 5477
4 1 2479 1 2276 3 1992 2154 9838 3 7775 7926 9241

Here above are two different sequences, for the first sequence

2 - Indicates that this sequence has two transactions.
3 - Indicates that first transaction has three items.
2635, 8073 and 1933 - three items of first transaction
1 - Indicates that second transaction has one item.
5477- One item of second transaction.

The Constrained Prefixspan algorithm uses different dataset format than the dataset generated by illimine generator. So, by java coding above dataset format has been changed so that it can be directly applied to Constrained Prefixspan algorithm. The Constrained Prefixspan algorithm applies Gap, Length, Aggregate and Frequency constraints on input dataset to generate sequential patterns which are based on timestamp at which each transaction occurred. So the modified dataset format is given below:

<1> 2635 8073 1933 -1 <2> 5477 -1 -2
<1> 2479 -1 <2> 2276 -1 <3> 1992 2154 9838 -1 <4> 7775 7926 9241 -1 -2

In above sequences <1>,<2>,<3>,<4> indicates the timestamp(day) at which particular transaction occur. Each transaction is separated by -1(end of transaction) and -2 represents the end of sequence. 2635, 8073,1933... are the actual items in a transaction.

VIII. CONCLUSION

"Discovery consists of seeing what everybody has seen and thinking what nobody has thought." Data mining is towards an effective and efficient tool for discovery. By mining, we can see the patterns hidden behind the data more accurately, more systematically and more efficiently. However, it is the data miner's responsibility to distinguish the gold from the dust.

This thesis is devoted to the sequential pattern mining, which belongs to one of the most frequently used data mining technique. As our world is now in its information era, a huge amount of data is accumulated everyday. A real universal challenge is to find actionable knowledge from a large amount of data. Data mining is an emerging research direction to meet this challenge. Many kinds of knowledge (patterns) can be mined from various data.

In this dissertation, we have focus on the problem of mining Sequential patterns efficiently and effectively, and enhanced a new class of pattern growth algorithm i.e. PrefixSpan. The enhanced algorithm can reduce the number of uninteresting pattern and increase the discovery capacity with a small value of minimum support and minimum confidence.

The C-Prefixspan's performance is better when average number of transactions per sequence increases. The C-Prefixspan algorithm generates more patterns when average number of transactions per sequence increases but these patterns are comparatively very less and more valuable than Prefixspan algorithm.

Most of the existing sequential pattern mining methods, however, focus on the concept of frequency in the sequence database and did not consider other factors like concepts of length, gap and aggregates. To apply these factors in sequential pattern mining process the Prefixspan algorithm which uses prefix projected pattern growth approach is modified to form Constrained Prefixspan algorithm. From the results presented in section VI, it is evident that as database size increases Constrained Prefixspan algorithm generates very less but more valuable patterns than Prefixspan algorithm.

The tests with large dataset showed that algorithm is scalable and the runtimes scaled with the size of the dataset. Since algorithms for frequent pattern enumeration and mining (breadth- first search) are not main memory based due to projected database construction, datasets which are larger than the available physical memory did not suffer due to operating system paging.

FUTURE DIRECTION

This study is still having improvement on it and especially on the researching results. In future, we can implement the C-PrefixSpan SP Miner using WEKA tool in JAVA Technology. Currently, we have performed well upto the feasibility study of the proposed system. Future work will also include evaluating C-PrefixSpan with various datasets, so that we can guarantee the efficiency of the proposed system.

With the success of pattern-growth methods, it is interesting to re-examine and explore many related problems, extensions and applications. This research can be further preceded in several ways. Further we may consider about adding other useful constraints to the C-PrefixSpan, for example the quantitative constraint that the quantity of each item in a sequence must be no less than a given threshold or the number of repetitions of item in a sequence must be no longer than a given threshold. Sequential pattern-based clustering is also possible using C-PrefixSpan, where once a set of frequent patterns are found, we can organize objects into some clusters according to the patterns they share. Also researches could extend it by adding fuzzy constraints, so that the boundary is no longer fixed but flexible. As a future extension, the C-Prefixspan algorithm can also be implemented on multiprocessor system

ACKNOWLEDGEMENT

The authors' wishes to thank all the colleagues for their guidance, encouragement and support in undertaking the research work.

REFERENCES

1. Xiaogang Wang, Yan Bai, Yue Li,"An Information Retrieval method based on sequential access pat terns" in 2010 Asia-Pecificconference on Wearable computing system, 2010.
2. Erich Allen Peterson, Peiyi Tang, "A Hybrid approach to mining frequent sequential patterns", ACMSE, CLEMSON, SC, USA, 2009, Unpublished.
3. Jean Francois Boulicaut, "If constraint based mining is the answer: what is the constraint? (Invited Talk)", IEEE International conference on data mining workshops, 2008.
4. Incrementally Fast Updated Sequential Pattern Trees, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008
5. Shigeaki Sakurai, Youichi Kitahata and Ryohei Orihara , "Discovery of Sequential Patterns based on Constraint Patterns", international journal of computational intelligence,2008
6. An Efficient Frequent Item set Mining Algorithm, Proceedings of Sixth International Conference on Machine Learning Cybernetics, Hong K-ong, 19-22 August 2007.
7. Jian Pei, Jiawei Han, Wei Wang, "Constraint-based sequential pattern mining : the pattern growth methods", *J Intell Inf Syst* , Vol. 28, No.2, pp. 133 –160 , 2007.
8. Yen-Liang Chen, Ya-Han Hu, "The consideration of recency and compactness in sequential pattern mining", *In Proceedings of the second workshop on Knowledge Economy and Electronic Commerce*, Vol. 42, Iss. 2 ,pp. 1203-1215, 2006.
9. Yu Hirate, Hayato Yamana, "Generalized Sequential Pattern Mining with Item Intervals", Journal of Computers, Vol. 1, No. 3, June 2006.
10. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu, *IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 10, October 2004*.
11. N Ju-Dong Ren, Yin-Bo Cheng, Lung-Lung Yang, "An Algorithm for Mining Generalized Sequential Patterns", Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.
12. Ron Kohavi, Rajesh Parekh, "Visualizing RFM Segmentation", *In Proceedings of the Fourth SIAM International Conference on Data Mining*, San Mateo, CA, 2004.
13. Ming-Yen Lin and Suh-Yin Lee, "Incremental update on sequential patterns in large databases by implicit merging and efficient counting", *Information Systems*, Vol. 29, No. 5, pp. 385-404, 2004.
14. Show-Jane Yen and Yue-Shi Lee, "Mining Sequential Patterns with Item Constraints", *DaWaK 2004: data warehousing and knowledge discovery: International conference on data warehousing and knowledge discovery*, Zaragoza, ESPAGNE, vol. 3181, pp. 381-390, 2004.
15. Wang, J., & Han J., "BIDE: Efficient Mining of Frequent Closed Sequences", *Proceedings of the 20th International Conference of Data Engineering*, Boston, USA, 2004, pp. 79-90.
16. C. Antunes, A. L. Oliveira, "Generalization of Pattern-growth Methods for Sequential Pattern Mining with Gap Constraints", *Machine Learning and Data Mining in Pattern Recognition, Third International Conference*, MLDM 2003, Leipzig, Germany, July 5-7, 2003, Proceedings.
17. Y. L. Chen, M. C. Chiang, and M. T. Kao, "Discovering time-interval sequential patterns in sequence databases", *Expert Systems with Applications*, Vol. 25, No. 3, pp. 343-354, 2003.
18. M. J. Zaki, "SPADE: an efficient algorithm for mining frequent sequences", *Machine Learning Journal*, Vol. 42, Iss. (1-2), pp. 31-60, 2001.
19. Sahista Machchhar, C.K. Bhensdadia and A.M. Ganatra, "Scientific Understanding, Comprehensive Evolution and More Informed Evaluation of Various Sequential Pattern Mining Algorithms",CiiT International journal of Data Mining & Knowledge engineering,pp. - Jan-2011
20. Helen Pinto, and Jiawei Han , "Multidimensional Sequential Pattern Mining", *In Proceedings of the 10th International Conference on Information and Knowledge Management, pp 81 - 88 , Atlanta, Georgia, USA , 2001.*
21. Ching-Yao-Wang, Tzung-Pei Hong, S.S.T. 2001 "Maintainance of sequential pattern for record deletion", International conference on data mining pp. 536-541.
22. A. Mortazavi, Q. Chen, U. Dayal, M. Hsu, J. Han and T. Pei, FREESPAN: Frequent Pattern Projected sequential pattern mining" Proc ACM SIGMOD, 2000.
23. M. N. Garofalakis, R. Rastogi, K. Shim, "SPIRIT: Sequential Pattern Mining with Regular Expression Constraints", *In Proceedings of 25th VLDB Conference*, pp. 223-234, San Francisco, California, 1999.
24. R. Agrawal and R. Srikant, "Mining sequential patterns: generalizations and performance improvements", *In Proceedings of the 5th International Conference on Extending Database Technology*, pp. 3-17, Avignon, France, 1996.
25. R. Agrawal and R. Srikant, "Mining sequential patterns", *In Proceedings of the 1995 International Conference on Data Engineering*, pp. 3-14, 1995.
26. Agrawal, R. and Srikant, R. 1994. "Fast algorithms for mining association rules", In Proceeding of 20th international conferenceon very large databases, VLDB, J.B. Bocca, M. Jarke, and C. Zaniolo.. Eds. Morgan Kauffman pp. 487-499.