A Review of Modern Sequential Rule Mining Techniques

Pushpraj Patel
PG Student, JIT
Borawan,Khargoan (M.P)

Mahesh Malviya
Asst. Professor, JIT
Borawan,Khargoan (M.P)

ABSTRACT
Sequential rule mining is an important data mining task with wide applications. The current algorithms for discovering sequential rules common to several sequences use very restrictive definitions of sequential rules. Among various data mining objectives the mining of frequent patterns has been the focus of knowledge discovery in databases. In this paper, we aim to investigate efficient algorithm for mining including association rule and sequential patterns. The time and space consumption of proposed algorithm will be lesser in comparison to previous algorithm.

From the broad variety of efficient algorithm that has been developed we will compare the most important ones. We will analyze the performance of various algorithms on the basis of both their run time performance and theoretical considerations. We use pattern growth approach for discovering valid rules such that it can be much more efficient and scalable.

General Terms
Data Mining, Sequential Rule Mining, Market Basket Problem

Keywords
Sequential Rules, Data Mining, Association Rules

1. INTRODUCTION
Mining is a vital step within the method of knowledge discovery in databases, during which intelligent strategies are applied so as to extract patterns. different steps in knowledge discovery method embody pre-mining tasks like data improvement (removing noise and inconsistent knowledge) and data integration (bringing data from multiple sources to one location and into a standard format), furthermore as post mining tasks like pattern analysis (identifying the actually attention-grabbing patterns representing knowledge) and knowledge presentation (presenting the discovered rules victimization visual image and knowledge illustration techniques).

In data mining, association rule learning may be a common and well researched technique for locating fascinating relations between variables in massive databases. Piatetsky-Shapiro describes analyzing and presenting sturdy rules discovered in databases using totally different measures of interest. Supported the idea of sturdy rules, Agrawal et al introduced association rules for locating regularities between product in massive scale transaction data recorded by point-of-sale (POS) systems in supermarkets.

Sequence Database Each sequence is a time-ordered list of item sets. An item set is an unordered set of items (symbols), considered to occur simultaneously.

Table.1: Data Mining Sequences

S.No	ID	Sequences
01	Seq1	{a,b},{c},{f},{g},{e}
02	Seq2	{a,d},{c},{b},{a,b,e,f}
03	Seq3	{a},{b},{f},{e}
04	Seq4	{b},{f,g}

Sequential Pattern Mining (SPM) is perhaps the foremost standard set of techniques for locating temporal patterns in sequence databases. SPM finds sub- sequences that are common to over min sup sequences. SPM is restricted for creating predictions. for instance, take into account a pattern. It's attainable that y seems often when an x but that there are also several cases wherever x isn't followed by y. For prediction, we'd like a mensuration of the confidence that if x happens, y can occur afterward.

A sequential rule usually has the shape X->Y

A sequential rule X⇒Y has 2 property

- Support: the number of sequences where X happens before Y, divided by the number of sequences.

- Confidence the number of sequences where X happens before Y, divided by the number of sequences where X occurs.

The Sequential Mining Rules finds the all needed rules, with support and confidence not less than user-defined thresholds values i.e. minSup and minConf.

For Example: an example of equential Rule Mining is as follows: Consider minSup= 0.5 and minConf= 0.5:

Table 2: A sequence database

S.No	ID	Sequences
01	Seq1	{a,b},{c},{f},{g},{e}
02	Seq2	{a,d},{c},{b},{a,b,e,f}
03	Seq3	{a},{b},{f},{e}
04	Seq4	{b},{f,g}

Table 3 Some sequential rules

ID	Rule	Support	Confidence
r1	{a,b,c}=>{e}	0.5	1.0
r2	{a}=>{c,e,f}	0.5	0.66
r3	{a,b}=>{e,f}	0.5	1.0
r4	{b}=>{e,f}	0.75	0.75
r5	{a}=>{e,f}	0.75	1.0
r6	{c}=>{f}	0.5	1.0
r7	{a}=>{b}	0.5	0.66
…..	…..	…..	…..

2. LITERATURE SURVEY

Sequential rule mining has been applied in many domains like stock exchange analysis (Das & Lin [4], Hsieh, Wu & Yang [11]), weather observation (Hamilton & Karimi, [8]) and drought management (Harms & Tadesse, [9], Deogun & Jiang, [5]).

The most known approach for sequential rule mining is that of Mannila & Verkano [13] and alternative researchers later on that aim at discovering partly ordered sets of events showing often within a time window during a sequence of events. Given these "frequent episodes", a trivial algorithmic rule will derive sequential rules respecting a lowest confidence and support (Mannila [13]).

However, their work can only get rules during a single sequence of events. Alternative works that extract sequential rules from one sequence of events are the algorithms of Hamilton & Karimi [8], Hsieh, Wu & Yang [11] and Deogun & Jiang [5], that respectively discover rules between many events and one event, between 2 events, and between many events.

Contrarily to those works that discover rules during a single sequence of events, some works are designed for mining sequential rules in many sequences (Das & Lin [4]; Harms & Tadesse [9]). as an example, Das & Lin [4] discovers rules where the left a part of a rule can have multiple events, however the correct half still needs to contain one event. This may be a significant limitation, as in real-life applications, sequential relationships may involve many events. Moreover, the algorithmic rule of Das & Lin [4] is extremely inefficient because it tests all potential rules, with none strategy for pruning the search space. To our information, only the algorithm of Harms & Tadesse [9] discovers sequential rules from sequence databases, and doesn't limit the quantity of events contained in every rule. It searches for rules with a confidence and a support higher or equal to user-specified thresholds. The support of a rule is here outlined because the number of times that the correct half occurs when the left part inside user-defined time windows.

However, one necessary limitation of the algorithms of Das & Lin [4] and Harms & Tadesse [9] comes from the actual fact that they're designed for mining rules occurring often in sequences. As a consequence, these algorithms are inadequate for locating rules common to several sequences. We have a tendency to illustrate this with an example. Think about a sequence information where every sequence corresponds to a client, and every event represents the items bought throughout a specific day. Suppose that one desires to mine sequential rules that are common to several customers. The algorithms of Das & Lin [4] and Harms [9] are inappropriate since a rule that seems again and again within the same sequence might have a high support even though it doesn't seem in any other sequences. A second example is that the application domain of this paper. we've designed an intelligent tutoring agent that records a sequence of events for every of its executions. We want that the tutoring agent discovers sequential rules between events, common to many of its executions, in order that the agent can thereafter use the principles for prediction throughout its following execution.

In general, we might categorise the mining approaches into the generate- and-test framework and also the pattern-growth one, for sequence databases of horizontal layout. Typifying the previous approaches [1,2 , 3], the GSP (Generalized sequential Pattern) algorithm [3] generates potential patterns (called candidates), scans every information sequence within the database to calculate the frequencies of candidates (called supports), then identifies candidates having sufficient supports for sequential patterns. These patterns in current database pass become seeds for generating candidates within the next pass. This generate-and- test method is continual till no additional new candidates are generated. Once candidates cannot fit in memory in a batch, GSP again scans the data to check the remaining candidates that haven't been loaded. Then, GSP scans for more than k times of the on-disk database if the maximum size of the discovered patterns is k, that incurs high value of disk reading. Despite that GSP was smart at candidate pruning, the quantity of candidates continues to be terribly large that might impair the mining efficiency.

The AIS (Agrawal, Imielinski, Swami) algorithm put forth by Agrawal [3] was the forerunner of all the algorithms used to generate the frequent itemsets and assured association rules, the description of that has been given at the side of the introduction of mining problem. The algorithm contains of 2 phases, the primary section constitutes the generation of the frequent itemsets. This is often followed by the generation of the confident and frequent association rules within the second section. The exploitation of the monotonicity property of the support of itemsets and therefore the confidence of association rules led to the improvement of the algorithm and it was renamed Apriori in a later purpose of time by Agrawal [15,16]. Although variety of algorithms were put forth following the introduction of Apriori

algorithm, a majority of them treated the optimisation of one or a lot of steps of the Apriori bearing the similar general structure. aboard Apriori, Agrawal [15] planned the AprioriTid and AprioriHybrid algorithms further. Apriori outperforms AIS on issues of assorted sizes. It beats by a factor of two for high minimum support and more than an order magnitude for low levels of support. SETM (SET-oriented Mining of association rules) [17] was perpetually outperformed by AIS. AprioriTid performed equivalently well as Apriori for smaller problem sizes but performance degraded twice slow when applied to large issues.

The support count procedure of the Apriori algorithm has attracted voluminous research due to the very fact that the performance of the algorithm principally depends on this aspect. Park et al. proposed an optimisation, known as DHP (Direct Hashing and Pruning) supposed towards limiting the number of candidate itemstes, shortly following the Apriori algorithms mentioned above. Brin et al place forth the DIC algorithm that partitions the database into intervals of a fixed size therefore to reduce the number of traversals through the database [18]. Another algorithm known as the CARMA algorithm (Continuous Association Rule Mining Algorithm) employs the same technique so as to limit the interval size to one.

The PrefixSpan (Prefix-projected sequential pattern mining) algorithm [4], representing the pattern-growth methodology [5, 4, 6], finds the frequent items after scanning the sequence data for a single time. The data is now projected, according to the frequent items, into many small size databases. Finally, the whole set of sequential patterns is found by recursively growing subsequence fragments in every projected database. Two optimizations for minimizing disk projections were represented in [4]. The bi-level projection technique, handling large databases, scans every data sequence two times within the (projected) information in order that fewer and smaller projected databases are produced. The pseudo- projection method, avoiding real projections, maintains the sequence-postfix of every data sequence during a projection by a pointer-offset pair. However, according to [4], most of the mining performance may be achieved only if the database size is reduced to the dimensions accommodable by the main memory by using pseudo-projection after using bi-level optimisation. Though PrefixSpan with success discovered patterns using the divide-and-conquer strategy, the value of disk I/O could be high because of the creation and process of the projected sub- databases.

Besides the horizontal layout, the sequence database is reworked into a vertical format consisting of items' id-lists [7, 8, 9]. The id list of any item could be a list of (sequence id & timestamp) pairs showing the occurring timestamps of the item in this sequence. Looking within the lattice shaped by id-list intersections, the SPADE (Sequential Pattern Discovery using Equivalence classes) algorithm [9] completed the mining in three passes of database scanning. Nevertheless, extra computation time is needed to rework a database of horizontal layout to vertical format that additionally needs extra space for storing many times larger than that of the initial sequence database.

CMRules: an association rule mining based mostly algorithm for the invention of sequential rules[24].

The users will specify min_sup as a parameter to a sequential pattern mining algorithm. There are 2 major difficulties in sequential pattern mining:

(1) effectiveness: the mining could return a large variety of patterns, several of that can be uninteresting to users, and

(2) efficiency: it usually takes substantial process time and space for mining the whole set of sequential patterns during a large sequence database.

In the context of constraint-based sequential pattern mining, (Srikant & Agrawal, [23]) generalized the scope of the Apriori-based sequential pattern mining to incorporate the shortage of time, sliding time windows, and user- defined taxonomy. Mining frequent episodes during a sequence of events studied by (Mannila, Toivonen, & Verkamo, [13]) also can be viewed as a constrained mining problem, since episodes are basically constraints on events in the acyclic graphs type. The classical framework on sequential pattern mining and frequent pattern mining relies on the anti-monotonic Apriori property of frequent patterns. A breadth-first, level-by-level search may be conducted to search out the whole set of patterns.

3. CONCLUSION

Sequential rule mining is used in many applications like market basket analysis, forecasting etc. In this paper, we analyzed some modern methods for sequential rule mining. There advantages and drawbacks are analyzed. In next paper, we will propose a more efficient method for sequential rule mining.

4. REFERENCES

[1] Rakesh Agrawal, Swami, A., & T. Imielminski, 1993, Mining Association Rules Between Sets of Items in Large Databases, *SIGMOD Conference*, pp. 207-216

[2] Rakesh Agrawal, & Ramakrishnan Srikant, 1995, Mining Sequential Patterns. Proc. Int. Conf. on Data Engineering, pp. 3-14.

[3] Han, J. Cheung, Wong, Y., , Ng. V., & D.W. 1996, Maintenance of discovered association rules in large databases: An incremental updating technique. *Proc. ICDE 1996*, 106-114.

[4] King Ip Lin., Heikki Mannila, Gautam Das, Gopal Renganathan, & Padhraic Smyth, 1998. Rule Discovery from Time Series. *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining*.

[5] Liying Jiang & Jiternder S Deogun, 2005. Prediction Mining – An Approach to Mining Association Rules for Prediction. *Proceeding of RSFDGrC 2005 Conference*, pp.98-108.

[6] Faghihi, U., Fournier-Viger, P., Nkambou, R. & Poirier, P., 2010. The Combination of a Causal Learning and an Emotional Learning Mechanism for Improved Cognitive Tutoring Agent. *Proceedings of IEA-AIE 2010* (in press).

[7] Kabanza, F., Nkambou, R. & Belghith, K. 2005. Path-planning for Autonomous Training on Robot Manipulators in Space. *Proc. 19th Intern. Joint Conf. on Artificial Intelligence*, 35-38.

[8] Hamilton, H. J. & Karimi, K. 2005. The TIMERS II Algorithm for the Discovery of Causality. *Proc. 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 744-750.

[9] Harms, S. K., Deogun, J. & Tadesse, T. 2002.

Discovering Sequential Association Rules with Constraints and Time Lags in Multiple Sequences. Proc. 13th Int. Symp. on Methodologies for Intelligent Systems, pp. .373- 376.

[10] Hegland, M. 2007. The Apriori Algorithm – A Tutorial. Mathematics and Computation. *Imaging Science and Information Processing*, 11:209-262. [11] Hsieh, Y. L., Yang, D.-L. & Wu, J. 2006. Using Data Mining to Study Upstream and Downstream Causal Realtionship in Stock Market. *Proc.2006 Joint Conference on Information Sciences*.

[12] Laxman, S. & Sastry, P. 2006. A survey of temporal data mining. *Sadhana* 3: 173-198.

[13] Mannila, H., Toivonen & H., Verkano, A.I. 1997. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(1):259-289.

[14] Gregory Piatetsky-Shapiro and William Frawley, Discovery in Databases, AAAI/MIT Press, 1991.

[15] Rakesh Agrawal, and Ramakrishnan Srikant, 1994. "Fast Algorithms for Mining Association Rules", In Proceedings of the 20th Int. Conf. Very Large Data Bases, pp. 487-499.

[16] Rakesh Agrawal, & Ramakrishnan Srikant., 1995. "Mining generalized association rules". In: Dayal U, Gray P M D, Nishio Seds. Proceedings of the International Conference on Very Large Databases. San Francisco, CA: Morgan Kanfman Press, pp. 406-419.

[17] M. Houtsma, and Arun Swami, 1995. "Set-Oriented Mining for Association Rules in Relational Databases". IEEE International Conference on Data Engineering, pp. 25–33.

[18] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, 1997. "Dynamic itemset counting and implication rules for market basket data". In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, volume 26(2) of SIGMOD Record, pp. 255–264. ACM Press.

[19] Chen, E., Cao, H., Li, Q., & Qian, T. (2008). Efficient strategies for tough aggregate constraint-based sequential pattern mining. Inf. Sci., 178(6), 1498-1518.

[20] Masseglia, F., Poncelet, P., & Teisseire, M. (2003). Incremental mining of sequential patterns in large databases. Data Knowl. Eng., 46(1), 97–121.

[21] Wang, J. L., Chirn, G., Marr, T., Shapiro, B., Shasha, D., & Zhang, K. (1994). Combinatorial pattern discovery for scientific data: Some preliminary results. Proc. ACM SIGMOD Int'l Conf. Management of Data, (pp. 115-125).

[22] Yang, J., Wang, W., & Yu, P. S. (2001). Infominer: mining surprising periodic patterns. Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[23] Srikant, R.,& Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements. Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology.

[24] Philippe Fournier-Viger,Usef Faghihi,Roger Nkambou, Engelbert Mephu Nguifo-"CMRULES: An Efficient Algorithm for Mining Sequential Rules Common to Several Sequences"