COMPUSOFT, An international journal of advanced computer technology, 3 (9), September-2014 (Volume-III, Issue-IX)

ISSN:2320-0790

An International Journal of Advanced Computer Technology

State of The Art - Modern Sequential Rule Mining Techniques

[1]Ms. Anjali Paliwal, [2]Mr. Sourabh Dave

[1]M.E. Student, [2]Asst. Professor
Medicaps Institute of Technology & Management Indore (M.P.)

Abstract: This paper is state of the art of existing sequential rule mining algorithms. Extracting sequential rule is a very popular and computationally expensive task. We also explain the fundamentals of sequential rule mining. We describe today's approaches for sequential rule mining. From the broad variety of efficient algorithms that have been developed we will compare the most important ones. We will systematize the algorithms and analyze their performance based on both their run time performance and theoretical considerations. Their strengths and weaknesses are also investigated.

1. Introduction

Data mining is the process of extracting interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from large information repositories such as: relational database, data warehouses, XML repository, etc. Also data mining is known as one of the core processes of Knowledge Discovery in Database (KDD).

Of all the mining functions in the knowledge discovering process, frequent pattern mining is to find out the frequently occurred patterns. The measure of frequent patterns is a user-specified threshold that indicates the minimum occurring frequency of the pattern. We may categorize recent studies in frequent pattern mining into the discovery of association rules and the discovery of sequential patterns. Association discovery finds closely correlated sets so that the presence of some elements in a frequent set will imply the presence of the remaining elements (in the same set). Sequential pattern discovery finds temporal associations so that not only closely correlated sets but also their relationships in time are uncovered.

In a Sequence Database, each **sequence** is a time-ordered list of itemsets. An **itemset** is an unordered set of items (symbols), considered to occur simultaneously.

ID	Sequences
seq1	$\{a, b\},\{c\},\{f\},\{g\},\{e\}$
seq2	$\{a, d\},\{c\},\{b\},\{a, b, e, f\}$
seq3	$\{a\},\{b\},\{f\},\{e\}$
seq4	$\{b\},\{f, g\}$

Sequential Pattern Mining is probably the most popular set of techniques for discovering temporal patterns in sequence databases. SPM finds subsequences that are common to more than *minSup* sequences. SPM is limited for making **predictions.** For example, consider the pattern {x},{y}. It is possible that *y* appears frequently after an *x* but that there are also many cases where *x* is not followed by *y*. For **prediction**, we need a measurement of the confidence that if *x* occurs, *y* will occur afterward.

A **sequential rule** typically has the form X->Y .A sequential rule X⇒Y has **two properties**:

- **Support:** the number of sequences where X occurs before Y divided by the number of sequences.
- **Confidence** the number of sequences where X occurs before Y, divided by the number of sequences where X occurs.

Sequential Rule Mining finds all **valid rules**, rules with a support and confidence not less than user-defined thresholds *minSup* and *minConf*

For Example: An example of Sequential Rule Mining is as follows:

Consider *minSup*= 0.5 and *minConf*= 0.5:

ID	Sequences
seq1	{*a*, *b*},{*c*},{*f*},{*g*},{*e*}
seq2	{*a*, *d*},{*c*},{*b*},{*a*, *b*, *e*, *f*}
seq3	{*a*},{*b*},{*f*},{*e*}
seq4	{*b*},{*f*, *g*}

Fig: A sequence database

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
…	…	…	…

Fig: some rules found

2. A Survey of SRM Methods

In general, we may categorize the mining approaches into the generate-and-test framework and the pattern-growth one, for sequence databases of horizontal layout. Typifying the former approaches [1,2 , 3], the *GSP* (**G**eneralized **S**equential **P**attern) algorithm [3] generates potential patterns (called *candidates*), scans each data sequence in the database to compute the frequencies of candidates (called *supports*), and then identifies candidates having enough supports as sequential patterns. The sequential patterns in current database pass become seeds for generating candidates in the next pass. This generate-and-test process is repeated until no more new candidates are generated. When candidates cannot fit in memory in a batch, *GSP* re-scans the database to test the remaining candidates that have not been loaded into memory. Consequently, *GSP* scans at least *k* times of the on-disk database if the maximum size of the discovered patterns is *k*, which incurs high cost of disk reading. Despite that *GSP* was good at candidate pruning, the number of candidates is still very huge that might impair the mining efficiency.

The *PrefixSpan* (**Prefix**-projected **S**equential **pattern** mining) algorithm [4], representing the pattern-growth methodology [5, 4, 6], finds the frequent items after scanning the sequence database once. The database is then projected, according to the frequent items, into several smaller databases. Finally, the complete set of sequential patterns is found by recursively growing subsequence fragments in each projected database. Two optimizations for minimizing disk projections were described in [4]. The *bi-level projection* technique, dealing with huge databases, scans each data sequence twice in the (projected) database so that fewer and smaller projected databases are generated. The *pseudo-projection* technique, avoiding physical projections, maintains the sequence-postfix of each data sequence in a projection by a pointer-offset pair. However, according to [4], maximum mining performance can be achieved only when the database size is reduced to the size accommodable by the main memory by employing *pseudo-projection* after using *bi-level* optimization. Although *PrefixSpan* successfully discovered patterns employing the divide-and-conquer strategy, the cost of disk I/O might be high due to the creation and processing of the projected sub-databases.

Besides the horizontal layout, the sequence database can be transformed into a vertical format consisting of items' id-lists [7, 8, 9]. The id-list of an item is a list of (*sequence-id*, *timestamp*) pairs indicating the occurring timestamps of the item in that *sequence*. Searching in the lattice formed by id-list intersections, the *SPADE* (**S**equential **PA**ttern **D**iscovery using **E**quivalence classes) algorithm [9] completed the mining in three passes of database scanning. Nevertheless, additional computation time is required to transform a database of horizontal layout to vertical format, which also requires additional storage space several times larger than that of the original sequence database.

With rapid cost down and the evidence of the increase in installed memory size, many small or medium sized databases will fit into the main memory. For example, a platform with 256MB memory may hold a database with one million sequences of total size 189MB. Pattern mining performed directly in memory now becomes possible. However, current approaches discover the patterns either through multiple scans of the database or by iterative database projections, thereby requiring abundant disk operations. The mining efficiency could be improved if the excessive disk I/O is reduced by enhancing memory utilization in the discovering process.

3. Conclusion

In this paper, we surveyed the list of existing sequential rule mining techniques. We restricted ourselves to the classic sequential rule mining problem. It is the generation of all sequential rules that exists in market basket like data with respect to minimal thresholds for support & confidence.

In a forthcoming paper, we pursue the development of a novel algorithm that efficiently mines sequential association rules from a market basket data set.

References

1. R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, pp. 3-14, March 1995.
2. F. Masseglia, F. Cathala, and P. Poncelet, "The PSP Approach for Mining Sequential Patterns," *Proceedings of 1998 2nd European Symposium on Principles of Data Mining and Knowledge Discovery*, Vol. 1510, Nantes, France, pp. 176-184, Sep. 1998.
3. R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," *Proceedings* of the *5th International Conference on Extending Database Technology*, Avignon, France, pp. 3-17, 1996. (An extended version is the IBM Research Report RJ 9994)
4. J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-projected Pattern Growth," *Proceedings of 2001 International Conference on Data Engineering*, pp. 215-224, 2001.
5. J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal and M.-C. Hsu, "FreeSpan: Frequent Pattern-projected Sequential Pattern Mining," *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 355-359, 2000.
6. H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal, "Multi-Dimensional Sequential Pattern Mining," *Proceedings of the 10th International Conference on Information and Knowledge Management*, pp. 81-88, 2001.

7. J. Ayres, J. E. Gehrke, T. Yiu, and J. Flannick, "Sequential PAttern Mining Using Bitmaps," *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton*, Alberta, Canada, July 2002.

8. S. Parthasarathy, M. J. Zaki, M. Ogihara, and S. Dwarkadas, "Incremental and Interactive Sequence Mining," *Proceedings of the 8th International Conference on Information and Knowledge Management*, Kansas, Missouri, USA, pp. 251-258, Nov. 1999.

9. M. J. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," *Machine Learning Journal*, Vol. 42, No. 1/2, pp. 31-60, 2001.