International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

Analysis of TRuleGrowth algorithm for discovery of sequential rules

Mr.Siddharaj D. Pujari
Computer science and technology department
Department of technology
Kolhapur, India

Ms. R.V. Mane
Assistant Professor, Department of Computer Science and Technology,
Shivaji University, Department of technology,
Kolhapur, Maharashtra, India.

Dr.V.R.Ghorpade
Principal, D.Y.Patil college of engineering and Technology
Kolhapur, Maharashtra, India.

Abstract
Association rule mining is a method of finding an interesting relationshipbetween numbers of items in large databases. It is helpful for prediction purpose in various fields. For mining sequential rules efficiently a data mining task called Sequential rule mining is used. It uses ARM principle for discovery of sequential rules. Various algorithms have designed for discovery of sequential rules common to several sequences which are having an important limitation that discovered rules are very particular and therefore many related rules provides equal activity. There are major problems while discovering rules like there might be same rules with different items ordering, rules may not be discovered because they are individually considered uneventful, and the rules which are very particular that are not used for prediction purpose. To solve above issues, a partially ordered sequential rules (POSR), another form of sequential rules is used in which items in the left and right part of each rule are not ordered. For mining POSR efficiently a TRuleGrowth Rule generation algorithm is used which is efficient. A sliding-window, a user defined threshold is used to find out sequential rules appearing within the maximal time. For more accurate prediction and analysis more user interested constraints such as length, aggregate, gap, duration, attribute, super-pattern etc. can be used within user defined thresholds for discovering the sequential rules.

Keywords:ARM, Sequential rule mining, POSR, sliding-window, user interested constraints.

Introduction

Pattern mining is most widely used technique of data mining. It contains discovery of sequential patterns from the sequential database which are similar to multiple sequences. Algorithms with different approaches are available such as GSP [6], Prefix Span [7] and SPADE [8]. It uses support value for discovering frequent patterns. Another approach which includes confidence of sequential pattern is sequential rule mining. Sequential rule (episode rule or temporal rule) mining is used to discover rules appearing in a single sequence or across sequences or common to several sequences.Pattern mining is used for finding sub patterns from the sequential data. Sequential pattern mining does not provide a probability of the pattern will be followed. Sequential patterns only indicate a number of times pattern appearance. They do not provide any indication about a probability of future pattern. Sequential rule mining is used for prediction and used for fast decision making. It shows that if some episode occurs then some other episode will follow with some probability.For mining sequential rule the concept of confidence is used. There are major problems while discovering rules like there might be same rules with different items ordering, rules may not be discovered because they are individually considered uneventful, and the uninteresting or weak rules may not be used for further applications such as prediction. To solve this problem, a partially ordered sequential rule (POSR) more common form of rules is used, in which items in left and right part of rule are not ordered.

Sequential rule mining is used in different domains like management of drought, analysis of market, weather changes and relationship, online learning, and electronic commerce management. In online learning, sequential rule mining is used to find out and predict the learners‟ behavior and to find out patterns common to several learners solution. SRM is applicable in bioinformatics to analyze sequences like DNA sequence, protein sequence etc. or to find correlation among gene expression.

The TRuleGrowth rule generation method first generates rules having length of its left and right side of rule as 1.It keeps each every items position in every sequence. The items position for a sequence is represented as a number which gives the position of an itemset which contains an item. A sliding window constraint is used which discovers rules

appearing within a user specified window size. User interested constraints such as length, attribute, super-pattern, aggregate, duration, gap etc. are added to find out more sequential rules related to these constraints for more accuracy and prediction.

RelatedWork

The Fournier-Viger P et al. proposes a method for mining POSR [1].In this paper, Partially Ordered Sequential Rules(POSR) concept is used to discover sequential rules from various sequences in which the items in left and right part of the rule are not ordered. For this purpose, Rule Growth algorithm is used which is adequate and easily extendable. There are algorithms like CMRules and CMDeo for SRM which are not efficient with respect to time and memory requirements. The Rule Growth algorithm is useful for rule generation in less time as compared to previous algorithms and requires less memory.

Fabregue M et al. proposes an algorithm called as orderspan [2]. In this paper, orderspan is used to discover a group of patterns which are partially ordered and closed from the sequence transaction. It uses known properties of suffix and prefix. The orderspan algorithm uses a pattern-growth approach which uses divide and conquer strategy. The authors‟ experimental studies shows that an algorithm is more space efficient and uses pruning strategy to output complete set of patterns which are partially ordered and closed which eliminates excessive branches.

Pei J et al. Proposes a method for mining patterns with use of user interested constraints [4] is a framework developed for discovery of sequential patterns based on some user interested constraints. For effective and efficient processes such as analysis, prediction and decision making the constraints are very important while mining sequential patterns. Constraint is used to discover sequential patterns which fulfill a given constraint. There are some constraints that are used for sequential pattern mining which are Length, Item, aggregate, super-pattern, duration, regular expression, gap etc.

Analysis of TRuleGrowth Rule Generation algorithm

The TRuleGrowth Rule Generation algorithm takes a sequence database as input with minsup, minconf and sliding window as user defined thresholds. Some user interested constraints like length, attribute, Item, duration, gap etc. are used. First as part of data pre-processing it converts data from text file into proper sequences and stores occurrences of each item by using hash table. With use of mining algorithms such as GSP[6], PrefixSpan[7], SPADE[8] etc. the frequent items are identified and those items having support greater than minsup value are selected for expansion while those items having support less than user provided minsup value are rejected. For a generation of sequential rules TRuleGrowth rule generation algorithm first generates rules of length 1, a user defined constraint as sliding-window. For a generation of valid rules from smaller rules the Left Expansion and Right Expansion, sub processes are used. For effectiveness and efficiency user interested constraints like Length, Item, aggregate, super-pattern, regular expression, gap, and duration are used.

Rule Generation(S, minsup, minconf, sliding window)

Step 1: Scan the whole database, preprocess the data and store it in sequence list.
Step2: Store all occurrences of items for each sequence in hash table.
Step 3: Find out frequent items by calculating their

Figure 1: TRuleGrowth Rule Generation algorithm.

The corresponding Expansion _of_ LHS and Expansion _of_ RHS procedures are explained below.

The LHS expansion and RHS expansion are used to expand the left and right side of the sequential rule.LHS expansion is used for expanding left side of the rule e.g. if item ‘i’ is appended to the left part of the ruleX=>Y then resulting rule becomes X U {i}=> Y. RHS expansion is used for extending the right part of the rule e.g. if item ‘i’ is appended to the right side of the rule X=>Y then resulting rule becomes X => YU{i}. The LHSexpansion and RHS expansions are used for that rule which satisfies minimum support, minimum confidence,sliding_window constraint and user interested constraints.

Expansion _of_ LHS (a=>b, sids (a), sids (a=>b))

Step 1: For each sid∈sids (a=>b), check for each itemset ‘m’ in sequence sid. Initialize ‘Hasha’ and ‘Hashb’ to null.
Step 2: Remove items from hash tables if seen more than sliding_window-1.
Step3: If size of ‘Hashb’ table is equal to size of ‘b’ then add every item c∈a∩m in ‘Hasha’ table.
Step4: If size of ‘Hashb’ table is less than the size of ‘b’ then add every item d∈b∩m in ‘Hashb’ table with position of ‘m’ in sid.
Step5:If size of ‘Hasha’ is equal to size of ‘a’ and size of ‘Hashb’ is equal to size of ‘b’ then add sid to ‘a’ variable sids(a∪{c}⇒b) for every item c∉a∪b which is having id before first item of ‘b’within window size.
Step6: Check for rule a∪{c}=>b having support greater than misup then again check for eachsid∈ sids(a) such that sid∈ sids(c) then again call the same Expansion _of_ LHS procedure.
Step7: If the rule a∪{c}=>b is having confidence greater than minconf then rule is considered as valid rule.

Expansion_of_RHS (a=>b, sids (a), sids (a=>b))

Step 1: For each sid∈sids (a=>b), check for each itemset 'm' in sequence sid. Initialize 'Hasha'and 'Hashb' to null.
Step 2: Remove items from hash tables if seen more than sliding_window-1.
Step3: If size of 'Hasha' table is equal to size of 'a' then add every item c∈b∩m in 'Hashb' table with position of 'm' in sid.
Step4: If size of 'Hasha' table is less than the size of 'a' then add every item d∈a∩m in 'Hashb' table with position of 'm' in sid.
Step5: If size of 'Hashb' is equal to size of 'b' and size of 'Hasha' is equal to size of 'a' then add sidto 'a' variable sids (a=>b∪{c}) for every item c ∉a∪b which is having id before first item of 'a'within window size.
Step6: Check for rule a=>b∪{c} having support greater than minimum support then call theExpansion_of_LHS procedure after that calls the Expansion_of_RHS procedure.
Step7: If the rule a=>b∪{c} is having confidence greater than minimum confidence then the rule is considered as valid rule.

Figure 3:RHS expansion of rule.

Example 1. Figure 4 shows the execution of TRuleGrowth Rule Generation algorithm for sequence database for weather relation having attributes like outlook, temperature, humidity, Windy and decision about whether play will happen or not with above certain conditions.

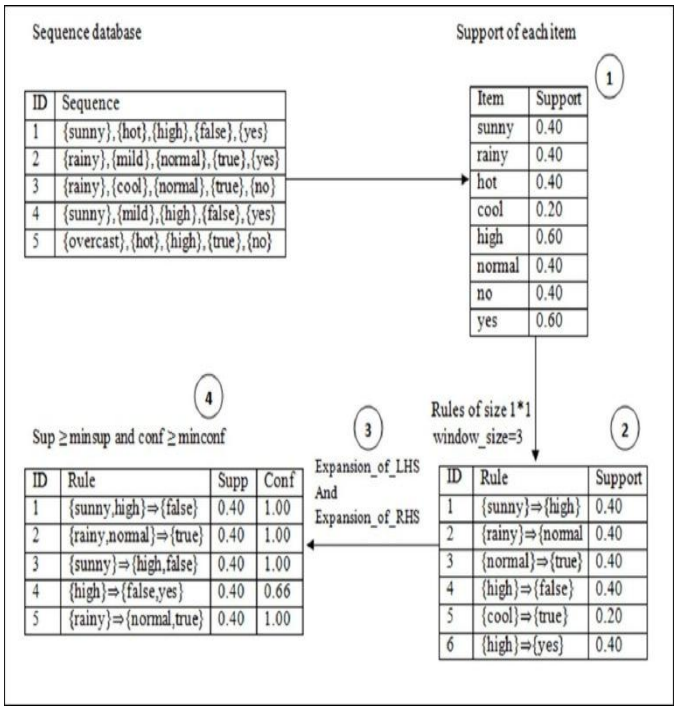

Figure4: Execution of Rule Generation algorithm for window size=3.

The above example shows execution of TRuleGrowth algorithm for weather data set. As part of data preprocessing the algorithm converts data from text file into proper sequences and stores value of position of every item in every sequence with use of hash table.

The numbers in the above figure shows below processes-

1) Those items having support greater than user defined minimum support are selected as valid frequent items.

2) After that algorithm discovers rules of length 1 value which occurs within user defined window size.

3) Then with use of two recursive procedures as Expansion_of_LHS and Expansion_of_RHS of rule and by using the hash tables it starts generation of valid rules from smaller rules.

4) Those rules having confidence greater than minimum confidence that rules are selected as valid sequential rules remaining rules are rejected.

By using above steps the valid sequential rules are generated, suppose 'n' is the number of sequential rules discovered, then the time complexity of this algorithm is O (n), which is linear.

Experimental Setup

The graphs shown below provide statistics of algorithm with and without use of constraint. The time required for constraint based discovery of sequential rules, the number of sequential rules and memory required to store rules is less as compared to discovery of sequential rules without constraint.

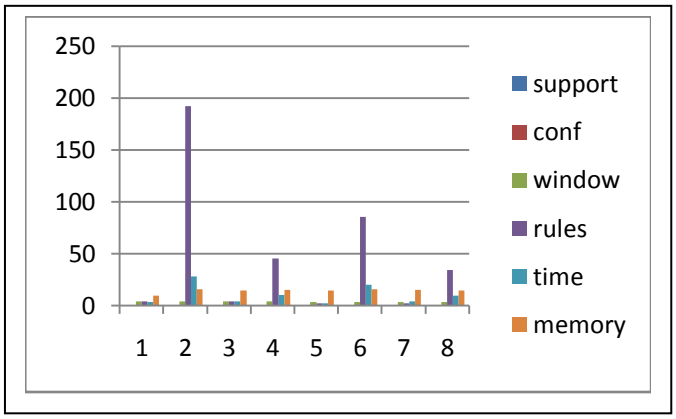

Figure 5: Rule Generation Statistics without Constraint.

Use of various constraints like length, item, duration, attribute, gap etc. helps for fast analysis and it improves overall decision making process in many application areas.

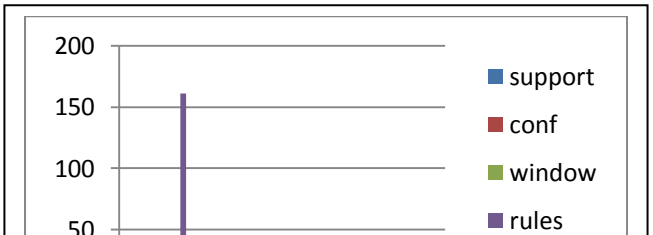

Figure 6: Rule Generation Statistics with length constraint.

Comparison

An existing system generates sequential rules based on user defined thresholds minsup and minconf are used for making prediction, analysis and decision making. The use of other threshold sliding_window and user interested constraints such as length, item, attribute, gap, duration etc. improves prediction accuracy by several orders of magnitude and requires less amount of time and memory than existing system, It also improves decision making and analysis process with use of proposed system.

Conclusion

An earlier study of sequential rule mining proposes algorithms like CMRules, CMDeo [1]. The drawbacks of these algorithms are similar to Apriori approach, as it uses generation of candidate items and uses test method. With this approach huge amount of uninteresting rules will generate, this makes this algorithm less efficient. It requires a large amount of memory to store rules and it is time consuming.

The TRuleGrowth algorithm is used which discovers sequential rules within user defined threshold sliding_window. It uses pattern-growth approach which avoids unnecessary candidate generation. With use of user interested constraints the generated rules will require less amount of memory for storage. Addition of user interested constraints like attribute, length, duration, gap etc. helps to improve prediction and analysis as well as decision making. The use of this algorithm in weather data improves prediction of future episode and it is useful in fast decision making process in disaster like situation.

References

[1] Fournier-Viger P, Wu C, Tseng V, Cao L, Nkambou R (2015) Mining Partially-Ordered Sequential Rules Common to Multiple Sequences. IEEE Transactions on Knowledge and Data Engineering 27:2203-2216. Doi: 10.1109/tkde.2015.2405509.

[2] Fabregue M, Braud A, Bringay S, Le Ber F, Teisseire M (2015) Mining closed partially ordered patterns, a new optimized algorithm. Knowledge-Based Systems 79:68-79. Doi: 10.1016/j.knosys.2014.12.027.

[3] Philippe Fournier, Cheng-Wei Wu, Vincent S. Tseng, Roger Nkambou, Mining Sequential Rules Common to Several Sequences with the Window Size Constraint. Advances Volume 7310 of the series pp 299-304.

[4] Pei J, Han J, Wang W (2007) Constraint-based sequential pattern mining: the pattern-growth methods. Journal of Intelligent Information Systems 28:133-160. doi: 10.1007/s10844-006-0006z.

[5] Hoang Thi Hong Van , Vo Thi Ngoc Chau , Nguyen Hua Phung, An Efficient Tree-based Rule Mining Algorithm for Sequential Rules with Explicit Timestamps in Temporal Educational Databases. Recent Developments in Intelligent Information and Database Systems Volume 642 of the series Studies in Computational Intelligence pp 411-421.

[6] R. Agrawal, R. Srikant, "Mining Sequential Patterns," Proc. 11th Intern. Conf.on Data Eng., pp. 3-14, 1995.

[7] J. Pei, J. Han et al., "Mining Sequential Patterns by Pattern-Growth:ThePrefixSpanApproach,"IEEE,Transactions Knowledge and Data Eng., vol. 16, no. 10, pp. 1-17, 2004.

[8] M. J. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," Machine Learning, vol. 42, no.1-2, pp. 31-60, 2001.

[9] "An Approach to Mining Sequential Rules Common to Several Sequences" G.Jayagopi, Dr.S.Pushpa1st International Conference on Innovations in Computing & Networking (ICICN16) 12 & 13, May"2016.

[10] Fournier-Viger P, Faghihi U, Nkambou R, Nguifo E (2012) CMRules: Mining sequential rules common to several sequences. Knowledge-Based Systems 25:63-76. Doi: 10.1016/j.knosys.2011.07.005.

[11] Railean I, Lenca P, Moga S, Borda M (2013) Closeness Preference – A new interestingness measure for sequential rules mining. Knowledge-Based Systems 44:48-56. doi: 10.1016/j.knosys.2013.01.025.

[12] Han J, Pei J, Yan X (2004) from sequential pattern mining to structured pattern mining: A pattern-growth approach. Journal of Computer Science and Technology 19:257-279. Doi: 10.1007/bf02944897.

[13] Uppal V (2015) An Efficient Algorithm for Approximate Frequent Intemset Mining. International Journal of Database Theory and Application 8:279-288. Doi: 10.14257/ijdta.2015.8.3.24.

[14] "Data Mining Introductory and Advanced Topics" - Margaret H. Dunham.2002.

[15] Jiawei Han, Micheline Kamber. "Data Mining: Concepts and Techniques".2000.