Middle-East Journal of Scientific Research 24 (S2): 360-365, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.24.S2.211

A Horizontal Formatting of Partially-Ordered Sequentialmining with Apriori-Based Algorithm

P. Yuvanesh and S. Arunkumar

M. Tech (CSE), Department of Computer Science, SRM University, Chennai, India

Abstract: quences are commonly occurring inany metric space that facilitates either total or partial ordering. Events in time, organic module in an amino acid, website traversal, computer networks and characters in a text string are examples of where the existence of sequences are significant and where the detection of frequent (totally or partially ordered) subsequences are also useful. Sequential pattern mining has arisen as a technology to discover such subsequences. A circumstance of algorithms for mining sequential rules common to multiple sequences is that rules are very specific and therefore many similar rules are represent the same situation. To address these issues, we explore the idea of "partially-ordered sequential rules" (POSR), In general form of sequential rules such that items in the forerunner and resultant of each rule are unordered. We propose the Apriori algorithm for an inherent problem facing different model of pattern mining algorithm either constraints or approximate patterns are commonly used solutions. The algorithm takes the edge of size in the candidate set of k-itemsets and also reduce the I/O layout by fragmenting the transaction records of the database. The performance of Apriori algorithm is to improve efficiency, so that we can used to refer association information from enormous data transfer rate.

Key words: Apriori · Partially-ordered · Sequential minig · Frequent Itemsets

INTRODUCTION

Sequential pattern mining is an important data mining task with wide applications. It consists of discovering sub sequences that are common to multiple sequences. Several algorithms have been proposed for this task such as GSP, Prefix Span, SPADE and CM-SPADE. However, sequential patterns found by these algorithms are often misleading for the user. The reason is that patterns are found solely on the basis of their support (the percentage of sequences in which they occur). For instance, consider the sequential pattern {Vivaldi}, {Handel}, {Berlioz} meaning that customer(s) bought the music of Vivaldi, Handel and Berlioz in that order. This sequential pattern is said to have a support of 50 %.

A solution to this problem would be to add a measure of the confidence or probability that a pattern will be followed. But adding this information to sequential patterns is not straightforward because they can contain multiple items and sequential pattern mining algorithms have just not been designed for that. An alternative is that, consider the confidence of a sequential pattern is sequential rule mining.

A sequential rule (also called episode rule, temporal rule or prediction rule) indicates that if some event(s) occur, some other event(s) are likely to follow with a given confidence or probability. Sequential rule mining has been applied in several domains such as drought management stock market analysis weather observation reverse engineering e-learning and e-commerce. Algorithms for sequential rule mining are designed to either discover rules appearing in a single sequence across sequences or common to multiple sequences.

Association rule mining also has been applied to the learning of sequential patterns mining, which is a restrictive form of association rule mining in the sense that not only the occurrences themselves, but also the order between the occurrences of the items is taken into account. The extra action of sequential patterns has been used in e-learning for evaluating the learners' activities and can be used in adapting and customizing resource delivery [1]; discovering and comparison with expected behavioral patterns specified by the instructor that describes an ideal learning path [2]; giving an indication of how to best organize the educational web space and be able to make suggestions to learners who share similar

Corresponding Author: P. Yuvanesh, M.Tech(CSE), Department of Computer Science, SRM University, Chennai, India.

characteristics [3]; generating personalized activities to different groups of learners [4]; supporting the evaluation and validation of learning site designs [5]; identifying interaction sequences indicative of problems and patterns that are markers of success [6].

The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

- Consider a database, D, consisting of 9 transactions.
- Suppose min. support count required is 2 (i.e. min_sup = 2/9 = 22 %)
- Let minimum confidence required is 70%.
- We have to first find out the frequent itemset using Apriori algorithm.
- Then, Association rules will be generated using min. support & min. confidence.

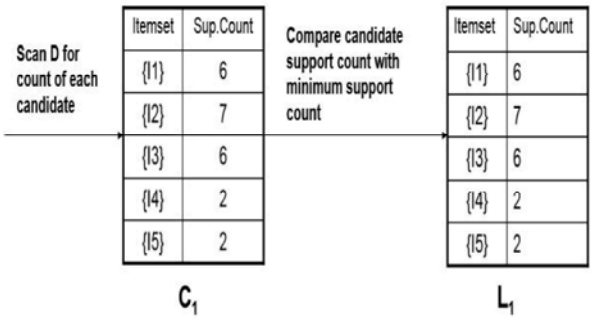

Fig. 1: Influential algorithm

Related Work: In the association rule mining area, most of the research efforts went in the first place for improving the algorithmic performance [7] and in the second place into reducing the output set by allowing the possibility to express constraints on the desired results. Over the past decade a variety of algorithms that address these issues through the refinement of search strategies, pruning techniques and data structures have been developed. While most algorithms focus on the explicit discovery of all rules that satisfy minimal support and confidence constraints for a given dataset, increasing consideration is being given to specialized algorithms that attempt to improve processing time or facilitate user interpretation by reducing the result set size and by incorporating domain knowledge [8]. There are also other specific problems related to the application of association rule mining from e-learning data. When trying to solve these problems, one should consider the purpose of the association models and the data they come from. Nowadays, normally, data mining tools are designed more for power andflexibility than for simplicity. Most of the current data mining tools are too complex for educators to use and their features go well beyond the scope of what an educator might require. As a result, the courses administrator is more likely to apply data mining techniques in order to produce reports for instructors who then use these reports to make decisions about how to improve the student's learning and the online courses. However, it is most desirable that teachers participate directly in the iterative mining process in order to obtain more valuable rules. But normally, teachers only use the feedback provided by the obtained rules to make decisions about modification to improve the course, detect activities or students with problems, etc. Some of the main drawbacks of association rule algorithms in e-learning are: the used algorithms have too many parameters for somebody non expert in data mining and the obtained rules are far too many, most of them non-interesting and with low comprehensibility. In the following subsections, we will tackle these problems.

The application of traditional association algorithms will be simple and efficient. However, association rule mining algorithms normally discover a huge quantity of rules and do not guarantee that all the rules found are relevant. Support and confidence factors can be used for obtaining interesting rules which have values for these factors greater than a threshold value. Although these two parameters allow the pruning of many associations, another common constraint is to indicate the attributes that must or cannot be present in the antecedent or consequent of the discovered rules. Another solution is to evaluate and post-prune the obtained rules in order to find the most interesting rules for a specific problem. Traditionally, the use of objective interestingness measures has been suggested [9], such as support and confidence, mentioned previously, as well as others measures such as Laplace, chi-square statistic, correlation coefficient, entropy gain, gin, interest, conviction, etc. These measures can be used for ranking the obtained rules in order than the user can select the rules with highest values in the measures that he/she is more interested.

Apriori Algorithm: In Classical Apriority algorithm, when candidate item sets are generated, the algorithm needs to test their occurrence frequencies. The manipulation with redundancy will result in high frequency in querying,

So tremendous amount of resources will be expended in time or in space. Therefore the improved algorithm was proposed for mining the association rules in generating frequent k item sets. Instead of judging whether these

candidates are frequent item sets after generating new candidates, this new algorithm finds frequent item sets directly and removes the subset that is not frequent, based on the classical Apriority algorithm. The improvement is for reducing query frequencies and storage resources. The improved Apriority algorithm mines frequent item sets without new candidate generation [10].

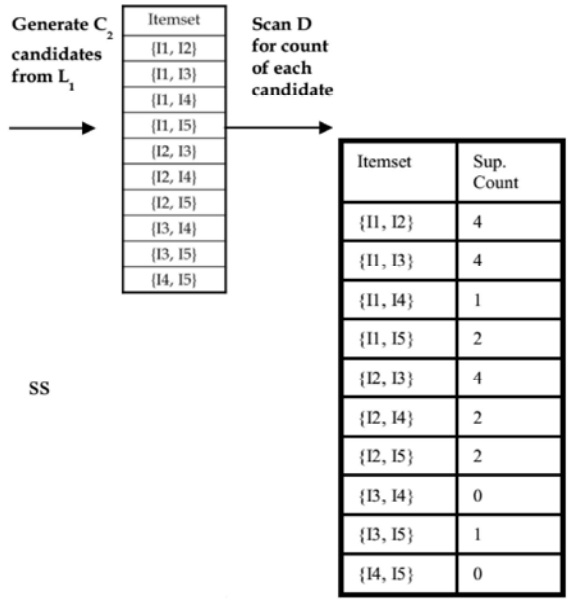

Fig. 2: Apriori Algorithm

Association rule mining also has been applied to the learning of sequential, patterns mining, which is a restrictive form of association rule mining in the sense that not only the occurrences themselves, but also the order between the occurrences of the items is taken into account. The extraction of sequential patterns has been used in learning for evaluating the learners' activities and can be used in adapting and customizing resource delivery discovering and comparison with expected behavioral patterns specified by the instructor that describes an ideal learning path giving an indication of how to best organize the educational web space and be able to make suggestions to learners who share similar characteristics generating personalized activities to different groups of learners ;supporting the evaluation and validation of learning site designs identifying interaction sequences indicative of problems and patterns that are markers of success. Finally, association rule mining has been used in the e-learning for classification. From a syntactic point of view, the main difference to general association rules is that classification rules have a single condition in the consequent which is the class identifier name. They have been applied in learning material organization, student learning assessments, course adaptation to the students' behavior and evaluation of educational web sites. This paper is organized in the following way: Section 2 describes the KDD process for association rule mining in e-learning. Section 3 describes the main drawbacks and solutions of applying association rule algorithms in LMS. Finally, in section 4, the conclusions and further research are outlined.

Pseudo Code:

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) do begin
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
return $\cup_k L_k$;

Proposed Method: The method we propose involves the mapping of the In items and Tm transaction from the database into a matrix A with size mix. The rows of the matrix represent the transaction and the columns of the matrix represent the items. The elements of matrix A are:

A= [aij] = 1,

if transaction i has item j = 0, otherwise

We assume that minimum support and minimum confidence is provided beforehand. In matrix A, The sum of the jth column vector gives the support of jthitem. And the sum of the ith row vector gives the S-O-T, that is, size of ith transaction (no. of items in the transaction).Now we generate the item sets. For, 1–frequent item set, we check if the column sum of each column is greater than minimum support. If not, the column is deleted. All rows with row sum=1 (S-O-T) are also deleted. Resultant matrix will represent the 1frequent item set.

Now, to find 2-frequent item sets, columns are merged by AND-ing their values. The resultant matrix will have only those columns whose column sum>=min_support. Additionally, all rows with row sum=2 are deleted. Similarly the kth frequent item is

found by merging columns and deleting all resultant columns with column sum<min_support and row sum=k.When matrix A has 1 column remaining, that will give the kth frequent item set.

The improvement of algorithm can be described as follows:

```
//Generate items, items support, their transaction ID
(1) L1 = find_frequent_1_itemsets (T);
(2) For (k = 2; Lk-1 ?Ö; k++) { //Generate the Ck from the
    LK-1
(3) Ck = candidates generated from Lk-1; //get the item
    Iw with minimum support in Ck using L1, (1=w=k).
(4) x = Get _item_min_sup(Ck, L1); // get the target
    transaction IDs that contain item x.
(5) Tgt = get_Transaction_ID(x);
(6) For each transaction t in Tgt Do
(7) Increment the count of all items in Ck that are found
    in Tgt;
(8) Lk= items in Ck = min_support;
(9) End;
(10) }
```

The algorithm is used to find out all the frequent item sets. In the first iteration, item set A directly constitutes the first candidate item set C1. Assume that A={a1,a2,…,am}, then C1={{a1},{a2},…, {am } }. In the Kth iteration, firstly, the candidate item set Ck of this iteration emerges according to the frequent item set Lk-1 found in the last iteration. (The candidate item set is the potential frequent item set and is the superset of the K-1th frequent item set. Item set with k candidate item sets is expressed as Ck, which was consisted by k frequent item sets Elk.) Then distribute a counter which has a initial value equals to zero to ever item set and scan affairs in database D in proper order. Make sure every affairs belongs to each item sets and the counter of these item sets will increase. When all the affairs have been scan, the support level can be gotten according to the actual value of |D| and the minimum support level of the certain Ck of the frequent item set. Repeat the process until no mew item occurs.

According to the properties of frequent item sets, this algorithm declines the number of candidate item sets further. In other words, prune Lk-1 before Ck occur using Lk-1. This algorithm can also be described as following: Count the number of the times of items occurs in Lk-1 (this process can be done while scan data D); Delete item sets with this number less than k-1 in Lk-1 to get Lk-1. To distinguish, this process is called Prune 1 in this study, which is the prune before candidate item sets occur; the process in Apriority algorithm is called Prune 2, which is the prune after candidate item sets occur. These rules seek to find the combination of clusters common in every cluster. More improved clustering techniques in Apriority algorithm with its rule formation can be found on the works of D. Karana mining frequent item sets from k-1 item sets. If k is greater than the size of the transaction T, there is no need to scan the transaction T which is generated by k-1 item sets.

The Partition Algorithm for Frequent Items (PAFI) reduces the scans of the database thereby improving the efficiency of Apriority algorithm and this is possible with the implementation of clustering method. With the work of S. Mural, due to the increase of data, mining frequent item sets even in the text mining domain have been amplified. Murali also made used of cluster analysis, the mined frequent item sets that were derived from meeting the defined threshold were arranged in descending order. Then, splitting the documents into partition using the resulting frequent item sets arrived at an ensuing cluster using the derived keyword. In the implementation of the Apriority algorithm in mining association rules from a dataset containing cases of different crimes against women as dataset available in Session court. Extraction of frequent item sets inessential towards mining useful and relevant patterns from datasets.

The improved algorithm was proposed for mining the association rules in generating frequent k-item sets. Instead of judging whether these candidates are frequent item sets after generating new candidates, this new algorithm finds frequent item sets directly and removes the subset that is not frequent, based on the classical Apriority algorithm. The improvement is for reducing query frequencies and storage resources. The improved Apriority algorithm mines frequent item sets without new candidate generation.

Apriority is a classic algorithm for learning association rules. Apriority is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions or having no timestamps. As it is common in association rule mining, given a set of item sets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets are extended one item at a time (a step known as candidate generation) and groups of candidates are tested against

the data. The algorithm terminates when no further successful extensions are found. The purpose of the Apriority Algorithm is to find associations between different sets of data. It is sometimes referred to as "Market Basket Analysis". Each set of data has a number of items and is called a transaction. The output of Apriority is sets of rules that tell us how often items are contained in sets of data.

DISCUSSION

Extracting rules directly from time-series data involves two coupled problems. First, since rules are inherently a symbolic representation, one must transform the low-level signal data into a more abstract symbolic alphabet. In this paper this is achieved by data-driven clustel'ing of signal windows in a similar way to that used in VQ data compression. The second problem is that of rule induction from symbolic sequences. Naturally, there is a trade-off between the quality of the abstraction and the quality of the rules induced using this abstraction. Parameters such as cluster window width, clustering methodology, number of clusters and so forth, may well affect the types of rules which are induced.

In this context it is important to keep in mind that the proposed technique is essentially intended as an exploratory method and thus, iterative and interactive application of the method coupled with human interpretation of the rules is likely to lead the most useful results (rather than any fully automated approach). Our methods are first steps and additional experimentation is needed to estimate the strengths and weaknesses of the method. Clearly there are several directions for generalizing the concepts introduced here, such as alternative abstractions (rather than pattern cancroids).

For example, the hierarchical piecewise linear representation introduced in (Keogh & Smyth 1997) may provide computationally efficient way to increase the expressive power of the underlying signal representation. The piecewise linear data structure implicitly handles variability in "warping" of signal structure (e.g., signal peaks which may be amplitude-scaled and/or stretched in time), a feature which is absent in the "fixed window" method described here.

Furthermore, a hierarchical representation may provide a practical way to incorporate the notion of multi-resolution scale into the representation in a natural manner, allowing for rules which relate events at different scales in the signal structure.

Comparative Study: We have discussed different algorithms for association rule mining on different size of database, First we have seen the improved Apriority algorithm which takes less time for generating frequent item set. Second we have seen the Feature Based Association Rule Mining Algorithm which is efficient than other algorithms and it speeds up the data mining process. Third we have seen the Optimized Distributed Association Rule Mining Algorithm which works on distributed database.The comparative study of all these algorithms is given in tabular form as below:

Table 1: The comparative study of Algorithms

No.	Parameters	Improved Apriori Algorithm	Farma	ODAM
1	Database Size	Small	Large	Very Large (Distributed)
2	Database Scan	N times	At most Once	N times on different database server.
3	Efficiency	More efficient than classical apriority and less efficient than FARMA	More Efficient than previous approach h.	More efficient for distributed end database .
4	Memory requirement	Large	Less	Less than FARMA
5	Speed	Slow	High	High

The Apriority algorithm is most representative algorithm for association mining. The classical Apriority algorithm has some disadvantages therefore in this paper we have studied different algorithms from which the Feature Based Association Rule Mining Algorithm works best for the large database and distributed database. Optimized Distributed Association Rule Mining Algorithm (ODAM) gives work properly. ODAM removes infrequent items and inserts each transaction into the main memory. While inserting the transactions, it checks whether they are already in memory. If yes, it increases that transaction's counter by one. Otherwise, it inserts that transaction into the main memory with a count equal to one. Finally, it writes all main-memory entries for this partition into a temp file.

CONCLUSION AND FUTURE SCOPE

In this paper, Apriority algorithm is improved based on the properties of cutting database. The typical Apriority algorithm has performance bottleneck in the massive data processing so that we need to optimize the

algorithm with variety of methods. The improved algorithm we proposed in this paper not only optimizes the algorithm of reducing the size of the candidate set of k-item sets, but also reduce the I/O spending by cutting down transaction records in the database. The performance of Apriority algorithm is optimized so that we can mine association information from massive data faster and better. Although this improved algorithm has optimized and efficient but it has overhead to manage the new database after every generation of Matrix. So, there should be some approach which has very less number of scans of database. Another solution might be division of large database among processors.

REFERENCES

1. Agrawal, R., T. Imielminski and A. Swami, 1993. Mining Association Rules Between Sets of Items in Large Databases," Proc. 13th ACMSIGMOD Intern. Conf. on Management of Data, pp: 207-216.
2. Cheung, D.W., J. Han, V. Ng. and Y. Wong, 1996. Maintenance of discovered association rules in large databases: An incremental updating technique, Proc. 12th Intern. Conf. on Data Eng., pp: 106-114.
3. Das, G., K.I. Lin, H. Mannila, G. Renganathan and P. Smyth, 1998. Rule Discovery from Time Series," Proc. 4th ACM Intern. Conf. Know.Discovery and Data Mining, pp: 16-22.
4. Faghihi, U., P. Fournier-Viger and R. Nkambou, 2012. A Computational Model for Causal Learning in Cognitive Agents, Knowledge Based Systems, 30: 48-56.
5. Fournier-Viger, P., U. Faghihi, R. Nkambou and E. Mephu Nguifo, 2012. CMRules: An Efficient Algorithm for Mining Sequential Rules Common to Several Sequences, Knowledge Based Systems, 25(1): 63-76, 2012.
6. Ms.Aarti Patil, Ms. Seem Kolkur, Ms.Deepali Patil: Advanced Apriori Algorithmss, International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August-2013.
7. Zaïane, O. and J. Luo, 2011. Web usage mining for a better web-based learning environment. In: Proc. of Int. Conf. on Advanced Technology for Education. pp: 60-64.
8. Pahl, C. and C. Donnellan, 2003. Data mining technology for the evaluation of web-based teaching and learning systems. In: Proc.of Int. Conf. E-learning, pp: 1-7.
9. Ha, S., S. Bae and S. Park, 2000. Web Mining for Distance Education. In: IEEE Int. Conf. on Management of Innovation and Technology, pp: 715-719.
10. Wang, W., J. Weng, J. Su and S. Tseng, 2004. Learning portfolio analysis and mining in scorn compliant environment. In: ASEE/IEEE Frontiers in Education Conf., pp: 17-24.