Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Novel Method for Refined Association Rule Mining from a Data Set

Anu Singh[1], Abhishek Raghuvanshi [2]

Department of Information Technology

Mahakal Institute of Technology, Ujjain, India

Abstract-Association mining is a fundamental and well researched data mining technique. However, depending on the choice of the parameters (the minimum support and minimum confidence), current algorithms can become very slow and generate an extremely large amount of results or generate none or too few results, eliding useful information. This is a serious problem because in practice users have limited resources for analyzing the results and thus are often only interested in finding a certain number of results, the main constraint here is the time consumed while the parameters are tuned.. In this paper, we propose an effective algorithm to mine the top-k association rules, where 'k' is the number of association rules to be found and is set by the user. Experimental and theoretical results indicate that the algorithm is highly effective and has excellent performance and scalability and that it is an advantageous alternative to classical association rule mining algorithms when the user want to control the number of rules generated.

Keywords: Association rule mining, top-k rules, frequent patterns, Support, Association rule.

I. INTRODUCTION

Data mining is a logical process that is used to search through large amounts of information in order to find important data. The objective of this technique is to find patterns that were previously unknown. Once you have obtain these patterns, you can use them to resolve a number of problems. Data mining (sometimes called data or knowledge discovery, KDD) is the process of analyzing data from different perspectives and summarizing it into useful information. Usually there are three processes. One is called preprocessing, which is executed before data mining techniques are applied to the right data. The pre-processing includes data cleaning, selection, integration, and transformation. The main process of knowledge discovery is the data mining process, in this process different algorithms are applied to produce hidden knowledge. After that the process comes known as post processing , which evaluates the mining result according to users requirements and domain knowledge. Regarding the evaluation results, the data can be presented if the result is adequate, otherwise we have to run some or all of those processes again until we get the satisfactory result. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Data mining is a powerful tool because it can provide you with relevant information that you can use to your own advantage.

The primary goal of data mining research is to attempt to create a system that efficiently navigates through the data complexity to give the user only what they need (Chen et al., 1996).Data mining can be used for many purposes. Here are a few exemplar:

Sales Promotion- Marketing staff may wish to know which products their customers are likely to buy together. With this information, they can bundle these items in a sales drive in order to increase revenue [1] [24].

Fraud Detection- Banks may wish to determine if any of their credit cards are being used for fraudulent purposes. Unusual spikes in a customer's spending pattern may indicate fraud [24].

Intrusion Detection- Running a data mining algorithm on past user logs may reveal that a certain sequence of events always leads to an unauthorized access attempt. The people in charge of security can then be warned of this to prevent future break-ins.

A. *Association Rule*

Association rule mining, one of the most important and well researched techniques of data mining, was first introduced in [1]. It aims to extract interesting frequent patterns, correlations, associations or casual structures among sets of items in the transaction databases or other data repositories. Association mining is a fundamental data mining technique. It identifies items that are associated with one another in data. In a database of transactions D with a set of n binary attributes (items) I, a rule is defined as an implication of the form X =>Y where X, Y are items and X ∩Y =NULL The sets of items (for short item sets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively. The support, supp(X), of an item set X is defined as the proportion of transactions in the data set which contain the item set. The confidence of a rule is defined as conf(X _=>Y) = supp(X UY) / supp(X). Following the original definition given by [1], association rules (ARs) are implication rules that inform the user about items most likely to occur in some transactions of a database. They are advantageous to use because they are simple, intuitive and do not make assumptions of any models. Their mining requires satisfying a user-specified minimum support and a user-specified minimum confidence from a given database at the same time. To accomplish this, association rule generation is a two step process. Firstly, minimum support is applied to find all frequent item-sets in a database. In a

second step, these frequent item-sets and the minimum confidence constraint are used to form rules. While the second step is simple, the first step requires more attention.

B. Measures of Association Rules

Essentially, association mining is about discovering a set of rules that is shared among a large percentage of the data [26]. Association rules mining tend to produce a large amount of rules. The objective is to find the rules that are useful to users.There are two ways of computing usefulness, being subjectively and objectively. Objective measures involve statistical analysis of the data, such as support and confidence [1].

Support- The rule X =>Y holds with support s if s% of transactions in D contain X U Y. Rules that have as greater than a user-specified support is said to have minimum support.

Confidence- The rule X=>Y holds with confidence c if c% of the transactions in D that contain X also contain Y .Rules that

have a c greater than a user-specified confidence is said to have minimum confidence.

Association rule mining finds frequent patterns, correlations, associations or causal structures among sets of items or objects in transaction database, relational database, and other information repositories. Frequent patterns are pattern that occurs frequently in a database. The key step of association rule mining is to identify frequent itemset. The problem of mining association rules is to find all association rules in a database having a confidence no less than a user-defined threshold minconf and a support no less than a user-defined threshold minsup. A major problem that has not been addressed is how the user should choose the thresholds to generate a desired amount of rules. To overcome this problem we propose to mine the top k association rules, where 'k' is the number of association rules to be found and is set by the user.

II. RELATED WORK

Top k association discovery: Under this the user specifies three parameters- measure of how potentially interesting an association is, filters for discarding inappropriate associations and number of associations to be discovered ,that is k[12]. With the help of these parameters, we can find the requirements like associations be non redundant, productive or pass statistical evaluation. Also the system finds the k association within the constraints of the user specified filters. This solves directly the problem of controlling the associations discovered, that are likely to be interesting to the user, without any need for a minimum support constraint. But because of its nature, association mining is extremely susceptible to making false discoveries. Three techniques are used for controlling the false discoveries risk, they are within search Bonferroni correction, holdout evaluation and randomization testing. During the search process the within –search approach applies statistical tests to each rule considered during the search process. This approach is used to overcome the multiple testing problem, also it supports top –k mining technique. The holdout evaluation approach splits the available data into two sets, the holdout data and the exploratory data. Associations are discovered using only the exploratory data and statistical tests are then applied using the holdout data. Experimental results shows that it is slightly more powerful than the within- search approach. Randomization testing that sets the discovery process operates by randomly shuffling the data in order to establish the null hypotheses i.e the items are independent of one another. Some features of this approach are it is straightforward to implement, and it automatically consider correlations between associations and the properties of discovery system that may complicate the analysis.

First we will introduce some basic algorithms for association rule mining, apriori series approaches. Then another milestone, tree structured approaches will be explained. Then some special issues of association rule mining, including multiple level ARM, multiple dimension ARM, constraint based ARM and incremental ARM.

A. *AIS Algorithm*:

The AIS[1] algorithm was the First algorithm proposed for mining association rule in [1]. It focus on improving the quality of databases together with necessary functionality to process decision support queries. In this algorithm only one item consequent association rules are generated, which means that the consequent of those rules only contain one item, for example we only generate rules like $X \cap Y \rightarrow Z$ but not those rules as $X \rightarrow Y \cap Z$. The databases were scanned many times to get the frequent itemsets in AIS. The main drawback of the AIS algorithm is too many candidate itemsets that finally turned out to be small are generated, which requires more space and squander much effort that turned out to be useless. At the same time this algorithm needs too many passes over the whole database.

B. *Apriori Algorithm:*

Apriori is a great improvement in the history of association rule mining, Apriori algorithm was first proposed by Agrawal in [2]. The AIS is just a straightforward approach that requires many passes over the database, producing many candidate itemsets and storing counters of each candidate while most of them turn out to be not frequent. Apriori is more effective during the candidate generation process for two reasons, Apriori hires a different candidates generation method and a new pruning technique.

Apriori algorithm still inherits the drawback of scanning the whole data bases many times. Based on Apriori algorithm, many new algorithms were designed with some modifications or improvements. Generally there were two approaches: one is to reduce the number of passes over the whole database or replacing the whole database with only part of it based on the current frequent itemsets, another approach is to explore different kinds of pruning techniques to make the number of candidate itemsets much smaller. Apriori-TID and Apriori-Hybrid [2], DHP [15], SON [18] are modifications of the Apriori algorithm.

Most of the algorithms introduced above are based on the Apriori algorithm and try to improve the efficiency by making some modifications, such as reducing the number of passes over the database; reducing the size of the database to be scanned in every pass; pruning the candidates by different techniques and using sampling technique. However there are two bottlenecks of the Apriori algorithm. One is that uses most of the time, space and memory known as complex candidate generation process. Another bottleneck is the multiple scan of the database.

C. *FP-Treed (Frequent Pattern Tree) Algorithm:*

To break the two bottlenecks of Apriori series algorithms, tree structure have been designed. FP-Tree [11], frequent pattern mining, is another milestone in the development of association rule mining, which breaks the two bottlenecks of the Apriori. The frequent item sets are generated with only two passes over the database and without any candidate generation process. FP-Tree was introduced by Han et al in [11]. By avoiding the candidate generation process and less passes over the database, FP-Tree is an order of magnitude faster than the Apriori algorithm. The frequent patterns generation process includes two sub processes : constructing the FP-Tree, and generating frequent patterns from the FP-Tree.

The efficiency of FP-Tree algorithm account for three reasons. First the FP-Tree is a compressed representation of the original database because only those frequent items are used to construct the tree, other unrelated information are pruned. Also by structuring the items according to their supports the overlapping parts appear only once with different support count. Secondly the FP Tree algorithm only scans the database twice. The frequent patterns are generated by the FP-growth procedure, constructing the conditional FP-Tree which contain patterns with specified suffix patterns. Thirdly, FP-Tree uses a divide and conquer method that considerably reduced the size of the subsequent conditional FP-Tree, longer frequent patterns are generated by adding a suffix to the shorter frequent patterns. Every algorithm has his limitations, for FP-Tree it is difficult to be used in an interactive mining system. During the interactive mining process, users may change the threshold of support according to the rules. However for FP-Tree the changing of support may lead to repetition of the whole mining process. Another limitation is that FP-Tree is that it is not suitable for incremental mining. Since as time goes on databases keep changing, new datasets may be inserted into the database, those insertions may also lead to a replication of the whole process if we employ FP-Tree algorithm.

D. Rapid Association Rule Mining (RARM):

RARM [8] is another association rule mining method that uses the tree structure to represent the original database and avoids candidate generation process. RARM is much faster than FP-Tree algorithm. By using the SOTrieIT structure RARM can generate large 1- itemsets and 2-itemsets quickly without scanning the database for the second time and candidates generation. Similar to the FP-Tree, every node of the SOTrieIT contains one item and the corresponding support count. The large itemsets generation process is as follows. Preprocessing, the database is scanned to construct the TrieIT, the process is similar to the process of generation the FP-Tree. For each transaction all the possible itemsets combinations are extracted and for those items that are already in the TrieIT increase their support count by 1, for those that still do not exist in the TrieIT the itemsets are inserted to the TrieIT with the corresponding support count be 1.

The difference between FP Tree and TrieIT is that TrieIT only increases the support count of the leaf node items while FP-Tree increases all the support counts along the path of the itemsets. Since the TrieIT stores the support counts individually, it requires large memory space which may not be satisfied, SOTrieIT (Support Ordered Trie Itemset) is introduced. To construct the SOTrieIT only 1-itemsets and 2-itemsets are extracted from each transaction, the building process is the same as in the construction of TrieIT, after all the itemsets of the same transaction were inserted the tree is ordered in a descending order of support count, the SOTrieIT has only two levels one is for 1-itemsets, another is for 2-itemsets.Since generating the large 2-itemsets is the most costly process during the mining process, experiments in [8] showed that the efficiency of generating large 1-itemsets and 2-itemsets in the SOTrieIT algorithm improves the performance drastically, SOTrieIT is much faster than FP-Tree, but SOTrieIT also faces the same problem as FP-Tree

E. Multiple Concept Level ARM:

In real life, for many applications, it is difficult to find strong association rules between data items at low or primitive level of abstraction due to the sparsity of data in multidimensional space [11]. While strong association rules generated at a higher concept level may be common sense to some users but it also can be novel to other users. Multiple level association rule mining is trying to mine strong association rules among intra and inter different levels of abstraction. For example, besides the association rules between milk and ham, it can generalize those rules to relation between drink and meat, at the same time it can also specify relation between certain brand of milk and ham. Research have been done in mining association rule at multiple concept levels [10], [17].

F. *Multiple Dimensional ARM:*

Multiple dimensional association rule mining is to discover the correlation between different predicts/attributes. Each predict/attribute is called a dimension, such as: age, occupation and buys in this example. At the same time multiple dimensional association rule mining concerns all types of data such as boolean data, categorical data and numerical data [4] The mining process is similar to the process of multiple level association rule mining. Firstly the frequent 1-dimensions are generated, then all frequent dimensions are generated based on the Apriori algorithm. A handful research literature exists in the study of constraints based association rule mining [14], [16], [5], [20], [9], [23], [6], [19]. Constraints based association rule mining is to find all rules from a given data-set meeting all the user specified constraints. Apriori and its variants only hire two basic constraints: minimal support and minimal confidence. However there are two points, one is some of the generated rules may be usefulness or not informative to individual users; another point is that with the constraints of minimal support and confidence those algorithms may miss some interesting

information that may not satisfy them. Some works have used the term "top K association rules". But they are applied to mining streams or mining non-standard rules.

III. PROPOSED ALGORITHM

The proposed algorithm is as follows:

Step 1: Input for algorithm is transaction database, minimum confidence (minconf) and a number of K rules

Step 2: Set internal minimum support (minsup) variable to 0.

Step 3: The algorithm looks for the rules as follows:

i. First it finds all valid rules of size 1*1, where supp(rule) > minsup and conf(rule)>minconf.
It is obtained by

$$sup(A \rightarrow C) = sup(A \cup \{C\})$$
$$conf(A \rightarrow C) = sup(\{C\})/sup(A).$$

Where A is an itemset and C is an item.

Then for each valid rule, a procedure is used whose role is to raise minsup called with the rule and item of list as parameters so that the rule is recorded in the list of the current top k rules found. (The procedure is in place which raises the minsup and updates and saves the list whenever new rule is found and a list contains top n rules ordered by support.)If the rule appears frequently then it is added to a set, to be later considered for expansion and a special flag is set to true for each such rule.

ii. After that a loop is performed to recursively select the rule with the highest support in the set (containing frequently appear rules) such that support of rule is greater or equal to minsup. The idea is to always expand the rule having the highest support because it is more likely to generate rules having a high support and thus to allow to raise minsup more quickly. The loop terminates when there is no more rule in the set with a support higher than minsup.

iii. Then for each rule, a flag indicates if the rule should be left or right

Instead of using classical approach, top K rules recursively grown its rule by adding items to the antecedent or consequent. To select the items that are added to a rule to grow it, top K rules scans the transactions containing the rule to find single items that could expand its left or right part. We name the two processes for expanding rules in top K rules left expansion and right expansion

Step 4: Set minsup = support of rule with the lowest support in list.

Step 5: Rules not respecting minsup anymore are removed from the list.

IV. RESULT

An example data set

TABLE I
SHOWS DATA SET [12]

1.	1 2 4 5
2.	2 3 5
3.	1 2 4 5
4.	1 2 3 5
5.	1 2 3 4 5
6.	2 3 4

Suppose Number of rules to be mined is 25 and minimum confidence is 0.5. Then our proposed algorithm generates following rules.

TABLE II
. ASSOCIATION RULES GENERATED

Rule	Support	Confidence
4, ==> 1,2	3	0.75
2, ==> 1,4,5	3	0.5
1,5, ==> 2,4	3	0.75
1, ==> 2,4,5	3	0.75

1,4,5, ==> 2	3	1.0
2,5, ==> 1,4	3	0.6
4, ==> 1,5	3	0.75
5, ==> 4	3	0.6
1, ==> 4,5	3	0.75
4, ==> 1,2,5	3	0.75
1,2,5, ==> 4	3	0.75
1,2, ==> 5	4	1.0
2, ==> 4,5	3	0.5
2,5, ==> 4	3	0.6
1,2,4, ==> 5	3	1.0
3, ==> 2,5	3	0.75
2,5, ==> 4	3	0.6
1,2,4, ==> 5	3	1.0
3, ==> 2,5	3	0.75
2,4, ==> 1,5	3	0.75
5, ==> 1,4	3	0.6
2, ==> 1,5	4	0.6666
4, ==> 2	4	1.0
1, ==> 2	4	1.0
1,5, ==> 2	4	1.0
1, ==> 2,4	3	0.75
5, ==> 1,2,4	3	0.6
4, ==> 1	3	0.75
1, ==> 2,5	4	1.0
4,5, ==> 1	3	1.0
5, ==> 1,2	4	0.8
2, ==> 1	4	0.66666
4,5, ==> 1,2	4	1.0
5, ==> 2	3	1.0
2, ==> 3	5	0.66666

V. CONCLUSION

We proposed an algorithm that let the user set *k*, the number of rules to be found. It has excellent scalability and execution time linearly increases with *k*. When the data set is too large, the general association rule mining algorithm can generate an extremely large amount of rules. It takes a lot of execution time and also consumes huge memory. In another case the association rule mining algorithm may generate too few rules. In this particular case we will loss valuable information. To overcome these above mentioned problems, we have proposed a novel algorithm for mining top ranked data from any standard data set. The algorithm has no problem running in reasonable time and memory limits for *k* values. We have tested our proposed algorithm on a standard data set (Market Basket Data set).

REFERENCES

[1] Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, P. Buneman and S. Jajodia, Eds. Washington, D.C., 207-216.

[2] Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In Proc. 20th Int. Conf. Very Large Data Bases, VLDB, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 487-499.

[3] Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. In Eleventh International Conference on Data Engineering, P. S. Yu and A.

[4] S. P. Chen, Eds. IEEE Computer Society Press, Taipei, Taiwan, 3-14.

[5] Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo: 1996, 'Fast Discovery of Association Rules'. In: U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.): Advances in Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press, pp. 307–328.

[6] Bayardo, R., Agrawal, R., and Gunopulos, D. 1999. Constraint-based rule mining in large, dense databases.

[7] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. 1997. Dynamic itemset counting and implication rules for market basket data. In SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA, J. Peckham, Ed. ACM Press, 255-264.

[8] Das, A., Ng, W.-K., and Woon, Y.-K. 2001. Rapid association rule mining. In Proceedings of the tenth international conference on Information and knowledge management. ACM Press, 474-481.

[9] Garofalakis, M. N., Rastogi, R., and Shim, K. 1999. SPIRIT: Sequential pattern mining with regular expression constraints. In The VLDB Journal. 223-234.

[10] Han, J. and Fu, Y. 1995. Discovery of multiple-level association rules from large databases. In Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95), ZÄurich, Switzerland, Septem- ber 1995.

[11] Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2, 2, 14-20.

[12] Webb,G.I. (2011). Filtered-top-k Association discovery. WIREs Data mining and knowledge discovery. Wiley,Pages 183-192.

[13] Ng, R. T., Lakshmanan, L. V. S., Han, J., and Pang, A. 1998. Exploratory mining and pruning optimizations of constrained associations rules. 13-24

[14] Park, J. S., Chen, M.-S., and Yu, P. S. 1995. An e®ective hash based algorithm for mining association rules. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, M. J. Carey and D. A. Schneider, Eds. San Jose, California, 175-186.

[15] Pei, J. and Han, J. 2000. Can we push more constraints into frequent pattern mining? In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, 350-354.

[16] Psaila, G. and Lanzi, P. L. 2000. Hierarchy-based mining of association rules in data warehouses. In Proceedings of the 2000 ACM symposium on Applied computing 2000. ACM Press, 307-312.

[17] Savesere, A., Omiecinski, E., and Navathe, S. 1995. An efficient algorithm for mining association rules in large databases. In Proceedings of 20th International Conference on VLDB.

[18] Smythe and Goodman. 1992. An information theoretic approach to rule induction from databases. In IEEE Transactions on Knowledge and Data Engineering. IEEE Computer Society Press

[19] Srikant, R., Vu, Q., and Agrawal, R. 1997. Mining association rules with item constraints. In Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD, D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, Eds. AAAI Press, 67-73.

[20] Webb and S. Zhang, "k-Optimal-Rule-Discovery," Data Mining and Knowledge Discovery, vol. 10, no. 1, 2005, pp. 39-79.

[21] Y. You, J. Zhang, Z. Yang and G. Liu, "Mining Top-k Fault Tolerant Association Rules by Redundant Pattern Disambiguation in Data Streams," Proc. 2010 Intern. Conf. Intelligent Computing and Cognitive Informatics, March 2010, IEEE Press, pp. 470-473

[22] Klemettinen M, Mannila H, Ronkainen P, Toivonen H, Verkamo A: Finding interesting rules from large sets of discovered association rules. In: Proceedings of the Third International Conference on Information and Knowledge Management, p. 401-407 1999.

[23] Webb GI: Magnum Opus. p. GI Webb & Associates: Melbourne, 2010.

[24] Pazzani M.J.,Knowledge discovery from data IEEE Intelligent System, Vol. 15Issue 12,pages 10-13, March – April 2000.

[25] Zaki M.J.,Scalable Algorithms for Association Mining.IEEE Transaction on Knowledge and Data Engineering, Vol. 12,No. 3, pages 372-390, May –June 2000.

