Incremental Mining of Association Rules based on Enumeration Constraints

Pooja Dubey[1], R.K. Gupta[2]
Department of CSE and IT
[1, 2] MITS, Gwalior

Abstract- *Business analysis is a subject of apprehension when Apriori was developed. As time advances, the techniques are developing and concerning on minimizing complexity, number of database scans, using definite checking to produce only practical rules. For producing rules, firstly by support value, algorithms can take out frequent itemsets. After specifying confidence, some rules are extracted. For producing buying pattern, a large number of areas are making use of association rule as cross marketing, profit loss calculation and store administration. These profits of association rule mining are still need to be enhanced. More user specific conditions can be included by constraints.*

Association rule mining using constraints is not only about applying constraints on the given dataset, but also a methodical way of constraints imposition and mining results based on these conditions. Specifying taxonomies, non-uniform support constraints, and user specified constraints , small number of rules will be generated, so extratcted rules can be more useful and can be of significance. In this paper we presented our work on incremental mining using enumeration constraints.

Keywords- Constrained association rules, Generalized association rules, Taxonomy based constraints, Nonuniform constraints

I. INTRODUCTION

Large data repositories gathers vast amount of data and it is required to devise an efficient information extraction technique. Suppoose milk, bread, butter items are there in a store and we want to know relative sale of these items is dependent on one another or not. For this purpose association rule mining is used. If support is 80% and bread and butter comes in database 85% times, they are frequent itemsets. A rule can be specified as Bread → Butter . It means if bread is purchased, butter will definitely be purchased 95% times. If confidence is 95% then this rule will be a valid rule, So frequent itemsets from database of transactions can be extracted by Apriori and interesting rules are generated. For association rule mining to be more specific, constraints are included. Taxonomy constraint is hierarchical approach of organizing itemsets. So more precise and specific rules can be generated.

II. RELATED WORK

In [8, 9, 10] number of techniques have been presented for constraint based rule mining . In many research works[11, 12, 13, 14, 15], efficient algorithms have been invented for incremental mining of association rules, which focus on the mining of rules when datasets are added.

Thomas et al. [4] proposed the utilization of incremental mining methods in association rule mining with constraints. They proposed the negative border theory for incremental mining. The negative border concept is used as a constraint relaxation technique, so that incremental mining can easily be executed. The central idea of this method is that the negative border concept is evolved to a larger class of constraints as well as enhancing the incremental frequent itemset mining algorithm to meet the constraints, increasing efficiency of the generalized incremental mining algorithm in the relational dataset and performance gains.

Several types of constraints are applied in this work as frequency constraint in which upward closure property is used that is, if an itemset is frequent, then so all its subsets are also frequent. Attribute constraint is used so that the itemsets which satisfy a Boolean function B are found. Suppose a constraint is applied that average of sold product price should be more than some value, this is known as aggregation constraint. Explicit filters can also be utilized. Different constraints are applied at different steps of associtin rule mining method. Attribute constraints and composite constraints are applied in candidate generation phase so useless itemsets can be pruned. Two types of concepts are used here as Subquery Approach and Vertical Approach. In first approach, support counting is done by a set of k nested subqueries where k is the size of the largest itemset. For every transaction that supports an itemset we find itemset and transaction id tuples, so a large table is formed. The Vertical approach ignores this by collecting all tids that support an itemset into a binary large object and creates itemset, tid-list. The tid-list for an itemset is obtained by finding the common the tid-lists of its items using a user-specified function.

Yafi et al. [7] presented the idea of Shocking associations. It is a self-upgrading filter that maintains

previously found knowledge as new shocking rules are found. The cause for applying shocking measures is to simulte the real time incidents as tsunami, earthquakes. The interesting rules are which are unpredictable. They are new because they are not present in previously discovered rules. The shocking measure is computed by the degree of shockingness of antecedent and consequent at conjunct level and then combined at rule level. This shocking measure is pushed into Apriori. The strong partial rules meet the minimum confidence and partial shocking rules are having interestingness higher than a user specified value. The usefulness of this is that lessnumber of rules are generated.

III. CONSTRAINT BASED MINING ON INCREMENTAL DATASETS

Incremental datasets necessitate an proficient technique for rule mining so as to lessen number of scans. In this work, we have proposed Incremental Mining of Association rules based on Constraints , it utilizes hash function [7] by which the algorithm finds the address of an itemset with no collision and increases the count value explicitly. A vector field is kept in hash table by which we can get which itemsets are fequent. It uses large space. For evolving efficiency of incremental constrained association rule mining, we used minimal perfect hashing for incremental mining.

Let an itemset is presented as {i, j, k,……,n} . Let hash address of an itemset is represented as H(i, j, k,…,n). Perfect hash function for 3-itemset can be expressed as:

If i=1, j=2, k=3, H(i, j, k) = 1

Otherwise, $H(i, j, k) = {}^{i-1}C_3 + {}^{j-1}C_2 + k$ (I)

As an example a hash table following this rule is shown in Fig. 2

Itemsets	Numerical Representation	Perfect Hash Address
{A, B}	{1, 2}	1
{A, D}	{1, 4}	4
{A, F}	{1, 6}	11
{B, D}	{2, 4}	5
{B, F}	{2, 6}	12
{D, F}	{4, 6}	14
{A, B, D}	{1, 2, 4}	2

Figure 2. Hash Table for 2 and 3 size itemsets

In this research work, the enumeration constraints are applied in the starting phase of data extraction from database. There are various classes of constraints carried out in this paper as frequency constraint, enumeration based specification. These are described as follows:

1. Frequency based constraint enables users to apply different support value on different items as per the requirement. For example, some items are purchased very less, still they are important and valuable item of the store so support is less for them while support value is higher for the items which are sold in large quantity. So varying support value should be specified for different set of items.

2. Enumeration based specification means enumerating in a group. For example, Supp(Y, Z) >= 0.3, where Y = {scorpio, hyundai} AND Z = {biscuit, cookies}. It specifies that any itemset containing at least one item in Y and one item in Z has minimum support 0.3. So the customer is interested in scorpio and hyundai rather than other brands of vehicles, and only biscuit and cookies rather than other products.

IV. PROPOSED ALGORITHM

The steps in the proposed algorithm are demonstrated as follows:

Step 1: Begin by getting the items from the database and put them into a local variable. Create the hash table for putting itemsets and calculating count of all the attributes .

Step 2: Use the enumeration based constraint and enumeration based support constraints to find the items in starting step. So rules will be extracted for only those items which will satisfy these conditions, these itemsets are called winner itemsets. Itemsets which will not satisfy the constraints will not be the part of finally generated rules.

Step 3: Use formula (I) to calculate the address of the candidate itemsets and store the itemset into that address.Database is scanned and count value of itemsets are incremented.

Step 4: If new items are added, the hash address will be calculated for them and they will be stored in the hash table. The updated database will also be scanned and count value of itemsets will be updated whether they were frequent in older database or not.

Step 5: At last, final frequent itemsets will be found and generate rules for these frequent itemsets. Calculate confidence for these rules and store in a file with confidence and support of the rule.

V. EXPERIMENTAL RESULTS

To evaluate the performance of proposed algorithm, we implemented it on Matlab. The system we used is the PC with Intel Core i3-4005U,1.70 GHz , 1.70 GHz 64 bit processor with 4 GB RAM. The environment we used is Matlab 7.8.0 (R2009a). The transaction database has 26 transations with 11 attributes. The time consumed in this program to run is 0.7962 seconds.

VI. CONCLUSION

We proposed an algorithm in this research work which utilizes the property of minimal perfect hash function, so there is no collision and imposed constraints for generation of rules desired by user. It is an incremental association rule mining algorithm based on constraints which is using the best technique of incremental rule mining so it gives the better result.

It is an improvement of this algorithm that hash table will not be constructed for every extended dataset. Only the count value of itemsets will be updated. It is improved over incremental mining algorithm FUP in terms of time complexity and effective rule extraction.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. “Mining association rules between sets of items in large databases”, Proc. of the ACM SIGMOD Conference on Management of Data, pp. 207–216, Washington, D.C., May 1993.

[2] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules”, Proc. of the 20th VLDB, Santiago, Chile, September 1994.

[3] R. Agrawal and K. Shim, “Developing tightly-coupled data mining applications on a relational database system”, Proc. of the KDD Conference, Portland, Oregon, August 1996.

[4] Shiby Thomas and Sharma Chakravarthy, “Incremental Mining of Constrained Associations”, Proceedings of 7th International Conference High Performance Computing — HiPC 2000,December 17–20, 2000 Proceedings, Vol. 1970, 2000, pp. 547-558.

[5] S. Sarawagi, S. Thomas, and R. Agrawal, “Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications”, Proc. of the ACM SIGMOD Conference on Management of Data, Seattle, Washington, June 1998.

[6] Chang, C. C.: The Study of an Ordered Minimal Perfect Hashing Scheme. In: Communications of the ACM, Vol. 27, No. 4, pp. 384-387. ACM Press. Washington, DC (1984).

[7] Elena Baralis, Luca Cagliero, Tania Cerquitelli, Paolo Garza, “Generalized association rule mining with constraints”, Information Sciences (194) pp. 68–84, 2012.

[8] Ramakrishnan Srikant and Quoc Vu and Rakesh Agrawal, “Mining Association Rules with Item Constraints”, American Association for Artificial Intelligence, 1997.

[9] Roberto J. Bayardo Jr., Rakesh Agrawal, Dimitrios Gunopulos, “Constraint-Based Rule Mining in Large, Dense Databases”, Proceedings of the 15th International Conference on Data Engineering, pp. 188-197, 1999.

[10] D. Cheung, J. Han, V. Ng, and C. Y. Wong. Large Databases: An Incremental Updating Technique. Proceedings of the 12th International Conference on Data Engineering, pp. 106—114, February 1996.

[11] N. F. Ayan, A. U. Tansel, and M. E. Arkun, “An Efficient Algorithm to Update Large Itemsets with Early Pruning”, Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 287—291, August 1999.

[12] Z. Zhou and C. I. Ezeife, “A Low-Scan Incremental Association Rule Maintenance Method”, Proceedings of the 14th Canadian Conference on Artificial Intelligence, June 2001.

[13] W. Cheung and O. R. Zaiane, “ Incremental Mining of Frequent Patterns without Candidate Generation or Support Constraint”, Proceedings of the 7th International Database Engineering and Application Symposium, July 2003.

[14] C. K. Leung, Q. I. Khan and T. Hoque, “CanTree: A Tree Structure for Efficient Incremental Mining of Frequent Patterns”, Proceedings of the Fifth IEEE

International Conference on Data Mining (ICDM'05), 2005.