Mining of Rare Itemsets in Distributed Environment

R N Yadawad
Dept of CSE
SDMCET, Dharwad
Karnataka, India

RBV Subramanyam, Ph.D.
Dept of CSE
NIT, Warangal
Telangana, India

U P Kulkarni, Ph.D.
Dept of CSE
SDMCET, Dharwad
Karnataka, India

ABSTRACT

The mining of rare itemsets involves finding rarely occurring items. It is difficult to mine rare itemsets with a single minimum support (minsup) constraint because low minsup can result in generating too many rules in which some of them can be uninteresting [3]. In the literature [4, 5], "multiple minsup framework" was proposed to efficiently discover rare itemsets. However, that model still extracts uninteresting rules if the items' frequencies in a dataset vary widely. In this paper, we are using the notion of "item-to-pattern difference" and multiple minsup based FP-growth-like approach proposed in [6] to efficiently discover rare itemsets in the distributed environment. To discover global rare itemsets in distributed environment, information regarding itemsets of local sites is collected in the form of MIS-tree at one site; that is, each site sends its local MIS-tree to a single site where a global MIS-tree will be constructed from all the MIS-trees received from all the sites. This global MIS-tree is mined to generate global rare itemsets. Experimental results show that this approach is efficient in terms of communication bandwidth consumed.

General Terms

Data mining

Keywords

Association rules, multiple minimum supports, MIS-tree, rare itemsets.

1. INTRODUCTION

It has been estimated that the amount of information in the world doubles every 20 months. The size and number of databases probably increases even faster [1]. So it can be said that in the current era is of data explosion. Because of the technological advances in many fields, especially in marketing, telecom, medicine, and weather forecasting etc, the rate at which the data is getting generated is enormous. Unfortunately, because of the huge data and also nature of data, it is difficult to take correct and timely decisions using such unprocessed data. As a result, we are in data rich but information poor situation. Data mining has recently attracted considerable attention from database practitioners and researchers because of its applicability in many areas such as decision support, market strategy and financial forecasts. Many approaches have been proposed to find out useful and valuable information from huge databases.

Association rules are an important class of regularities that exist in a database. Since the introduction of association rules in [2], the classic application is market basket analysis, where the objective is to analyze how the items purchased by customers are associated. The basic model of mining association rules is as follows:

Let I = {i_1, i_2, i_n} be a set of items. Let T be a set of transactions (dataset), where each transaction t (also called a data case) is a set of items such that t ⊆ I. A pattern (or an itemset) X is a set of items such that X⊆I. Itemset containing k number of items is called k-itemset. An association rule is an implication of the form, A ⇒ B, where A ⊂ I, B ⊂ I and A∩B = Φ. The rule A ⇒ B holds in T with support s, if s% of the transactions in T contain A∩B. This is taken to be the probability, P (AUB). The rule A⇒ B holds in T with confidence c, if c% of transactions in T that support A also support B. Given T, the problem of mining association rules is to discover all rules that satisfy user-specified minimum support (*minsup*) and minimum confidence (*minconf*) constraints. This is taken to be the conditional probability, P (B/A). So

$$S(A \Rightarrow B) = P(A \cup B) = \frac{f(A \cup B)}{|T|} \qquad 1.1$$

$$C(A \Rightarrow B) = P(B/A) = \frac{S(A \cup B)}{S(A)} \qquad 1.2$$

where f (AUB) represents the frequency of AUB in T and |T| represents the total number of transactions in T. Therefore, a common strategy adopted by many association rule mining algorithms is to decompose the problem into two major subtasks [1]:

(i) Frequent Itemset Generation. Here, the goal is to find all the itemsets that satisfy the *minsup* threshold. These itemsets are called *frequent* itemsets.

(ii) Rule Generation. In this step, all the high-confidence rules are generated from frequent itemsets found in the previous step. These rules are called *strong* rules.

The parameter *minsup* controls the minimum number of transactions that a rule must cover in a database. The parameter *minconf* controls the predictive strength of a rule.

Rare association rules [5] are the association rules containing rare items. Rare items are less frequent items. Real-world datasets are mostly non-uniform in nature containing both frequently and relatively rarely occurring entities. By the natures of items, it is meant that some items, by nature, appear less frequently than others. For example, in a supermarket, people buy bed and pillow much less frequently than they buy bread and jam. However, the profit earned on selling bed and pillow is very high as compared selling bread and jam. In general, those durable and/or expensive goods are bought less frequently, but each of them generates more profit. It is thus important to capture those rules involving less frequent items.

But, it must be done so without allowing frequent items to produce too many meaningless rules with very low supports (causing combinatorial explosion). The rare cases are more difficult to detect and generalize because they contain fewer data.

Mining of rare associations (association rules involving rare items) with single minsup approach may cause "rare item problem" [3]. This problem is as follows: "If *minsup* is high, frequent itemsets involving rare items are missed as the support of the rare items is less than the given *minsup*. In order to find frequent itemsets involving rare items, the *minsup* value should be fixed at low value. But the problem is the numbers of frequent itemsets explodes"

Thus, it is difficult to mine rare association rule by single *minsup* value. So to improve the performance of mining frequent patterns consisting of both frequent and rare items, efforts have been made to discover frequent patterns using "multiple minsup framework"[3,7,8]. In these approaches, technique used to mine both (frequent and rare) patterns is as follows:

(i) Each item in the transaction dataset is specified with a support constraint called minimum item support (MIS).

(ii) A pattern is defined as frequent, if its support is greater than or equal to the minimum MIS value among all of its items.

Generally, item's MIS value is specified based on support value of the item. So, as compared with frequent items, rare items are specified with relatively lower MIS value. Thus, these models prune patterns consisting of those uninteresting patterns which have low support and contain only frequent items. But they cannot prune uninteresting patterns with low support and contain highly frequent and rare items.

2. LITERATURE SURVEY

2.1 Centralized Algorithms for Mining Rare Itemsets

In the technique [2], minimum item support specified by the user reflects the natures of items and varied frequencies for each item in the database; hence in mining rule, different rules may need to satisfy different minimum supports. Minimum item supports thus enable us to achieve the goal of having higher minimum supports for rules that only involve frequent items, and having lower minimum supports for rules that involve less frequent items.

This approach efficiently prunes uninteresting patterns which have low support and contain only frequent itemsets. The disadvantage of this algorithm is that the user has to specify the values of minimum item support for each item. For this, the user needs to have the domain knowledge. Also, this approach cannot prune uninteresting patterns which have low support and contain both highly frequent and rare items. It still suffers from "rare item problem".

The authors in [5] have introduced "support difference" (SD) for calculating minsup value for each item. SD refers to the acceptable deviation of an item from its frequency so that an itemset involving that item can be considered as a frequent itemset. The disadvantage of this approach is that they cannot prune uninteresting patterns which have low support and contain both highly frequent and rare items.

The approach in [7] uses same steps as that of apriori approach for generating the candidate and frequent itemsets. Frequent itemsets are generated by using user specified percentile and least support. For calculating multiple minimum supports by using a single statistic measurement, percentile value is used. By using the statistic percentile value, it is more appropriate for automatic generation of multiple minimum support value from supports characteristic of its size.

2.2 Distributed Algorithm for Mining Rare Itemsets

In [8], rare association rules among items are discovered over the sites distributed geographically across the network. It utilizes the idea of using statistic percentile to produce multiple minimum supports to mine rare association rules. The algorithm AprioriMSD recognizes a distribution of data and handles the data in an appropriate way; that is, if a distribution of data is skewed, the algorithm applies natural logarithm on the data; thus, it helps to determine frequent itemsets. The author claims that AprioriMSD can discover more rare association rules with an optimized communication cost. However, it is evident from the work carried out that frequent itemsets, rather than rare itemsets, are found. In the literature, no other algorithms which deal with mining rare itemsets in distributed systems are present.

3. PROBLEM DEFINITION

In a transactional dataset T at local site, given LS value and mipd value, MIS values are calculated for items using Equation 1.3 (given in Section 4) and then, complete set of local frequent/rare patterns is discovered at that site that satisfy

(i) lowest MIS value among all its items and

(ii) ipd value less than or equal to the user-specified mipd.

Similarly at global site, given global LS value and global mipd value, and after receiving information regarding datasets in the form of MIS-tree from all clients, global MIS values are calculated and global rare itemsets are found that satisfy

(i) lowest MIS value among all its items across all sites

(ii) ipd value less than or equal to the user-specified global mipd.

Hence, pruning uninteresting patterns i.e. patterns with low supports contain only frequent items and patterns with low support and contain highly frequent and rare items.

4. DISTRIBUTED MINING OF RARE ITEMSETS

The idea behind the proposed approach is as follows:

In the proposed work, both frequent itemsets as well as rare itemsets are generated. However, the main objective is towards mining these rare itemsets, as there are many algorithms to mine only frequent itemsets. The proposed algorithm makes use of the approach given by [4] to prune the items which are uninteresting patterns.

In the datasets where items' frequencies vary widely, it generates uninteresting frequent patterns which have low support and contain highly frequent and rare items. So the main issue is to develop a model to filter such uninteresting frequent patterns. One of the characteristic features of an uninteresting frequent pattern generated in model [7] is that the support of a pattern is much less than the support of maximal frequent item within it. So here uninteresting patterns are filtered by limiting the difference between the support of a pattern, and the support of the maximal frequent item in that pattern [4].Equation 1.3 is used to calculate MIS value for an item given LS(Least Support).

$$\mathrm{MIS}(ij) = S(ij)-SD \quad \text{when } (S(ij)-SD)>LS \qquad 1.3$$

$$= LS \text{ otherwise}$$

At each site, given the transactional dataset T, items' MIS values and mipd value, the proposed approach utilizes the prior knowledge regarding the items' MIS values and discovers frequent and rare patterns with a single scan on the transactional dataset. The approach involves the following three steps:

1. Construction of a tree,called MIS-tree.

2. Generating compact MIS-tree from MIS-tree.

3. Mining compact MIS-tree using conditional pattern bases to discover complete set of frequent patterns.

At global site rare itemsets are generated as follows

1. Each site sends a MIS-tree to server.

2. Each MIS-tree converted back to transactional dataset.

3. Combine all transactional dataset into single transactional dataset.

4. Construction of global MIS-tree from transactional dataset.

5. Generating compact MIS-tree from MIS-tree.

6. Mining compact MIS-tree using conditional pattern base to discover complete set of frequent patterns.

Explanation of each step is as follows

4.1 Construction of a MIS-Tree

Initially, the items in the transactional dataset are sorted in descending order of their MIS values. Let this sorted order of items be L. Next, MIS-list is populated with all the items in L order. The support values of the items are set to 0. The MIS values of the items are set with their respective MIS values. A root node labeled "null" is created in the prefix-tree. Next, each transaction in the transactional dataset is scanned and MIS-tree is updated as follows:

(i) Items in the respective transaction are ordered in L order.

(ii) For these items, their frequencies (or supports) are updated by 1 in the MIS-list.

(iii) In L order, a branch which consists of these items is created in the prefix-tree.

To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links.

4.2 Generating compact MIS-Tree

The MIS-tree is constructed with every item in the transactional database. There may be items which do not generate any frequent pattern. We identify all those items which have support less than the lowest MIS value among all frequent items (or frequent 1-patterns) and prune them from the MIS-tree. In [7], it was observed that depending on the items' MIS values there exists a scenario where child nodes of a parent node can share a same item after pruning operation. So, tree-merging operation is performed on the pruned MIS-tree to merge such child nodes. The resultant MIS-tree is called compact MIS-tree.

4.3 Mining the Compact MIS Tree

Briefly, mining of frequent patterns from the compact MIS-tree is as follows. Choose each frequent length-1 pattern (or item) in the compact MIS-tree as the suffix pattern.

For this suffix-pattern construct its conditional pattern bases. From the conditional pattern bases, construct MIS-tree, called conditional MIS-tree, with all those prefix-subpaths that have satisfied the MIS value of the suffix-pattern and mipd. Finally, recursive mining on conditional MIS-tree results in generating all frequent patterns.

At global site

To find out global itemsets,the client-server architecture is used.

1. Each client has to register with the server and, after successful registration, it can send the MIS-tree object to server.

2. Each MIS-tree object is converted back to transactional dataset, which is reduced in size as compared to original transaction dataset.

```
for each item in header table
    while nextNode!=null
        if it is leafNode
            write currentNode counter into buffer
            while(currentNode.parent!=null)
            write currentNode into buffer
            assign currentNode=currentNode.parent
        else
            count all child counts
            remainingcount = currentNodecount-all child counts
          write remainingcount into buffer
          if(remainingcount!=0)
            {
              while(currentNode.parent!=null)
                  write currentNode into buffer
                assign currentNode=currentNode.parent
            }
        currentNode=nextNode.link
```

3. Combine all the transactional dataset into single dataset.

4. Construction of MIS-tree on global dataset. This step is same as followed in client side

Steps 5 and 6 are same as steps 2 and 3 of client side

5. IMPLEMENTATION DETAILS

The work proposed has been implemented in Java. The inputs to be mentioned are transactional dataset, least support (LS) and mipd. The model of distributed system followed is client-server architecture. Socket programming is used for communication. The details are as follows:

A server (program) runs on a specific computer and has a socket that is bound to a specific port. The server listens to the socket for a client to make a connection request. If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to a different port. It needs a new socket, hence, a different port number, so that it can continue to listen to the original socket for connection requests while serving the connected client.

Each of the clients has to register with Server. If server accepts request, then client can send a MIS-tree; else, it can not send a MIS-tree to a server. Upon receiving the MIS-tree from all clients, the server will convert tree back to transactional dataset and will combine all transactional datasets to produce global MIS-tree. The process of mining on global MIS –tree produces global rare and frequent itemsets. The advantage of the proposed algorithm is that instead of sending itemsets to the server, a MIS-tree is sent; hence, reduced bandwidth consumption. Only interesting itemsets will be generated hence uninteresting itemsets will not be generated.

6. RESULTS AND DISCUSSION

Here we have considered, for experimental purposes, 9 sites out of which 8 are clients and 1 is server. We are using dataset T10.14.D100K.

Table 1. Characteristics of dataset T10.14.D100K [8]

DATASET	No. of Transactions	No of Itemsets	Maximum No of Items in transaction	Average number of items in transaction
T10.14. D100K	100000	870	29	12

This dataset is horizontally distributed across 8 sites. At each site, MIS-tree is constructed.MIS values are calculated using the following formula [4]:

MIS(ij) = S(ij)-SD when (S(ij)-SD)>LS

= LS otherwise

Each site will send MIS-tree to the server. At server, each tree is combined to get a single transaction dataset. In the proposed work, user defined support difference and least support were used.

At global site:

Table 2. Itemsets generated for LS=0.1% and SD=0.25%

Item sets generated	1355	2668	3771	4717	4928	5140	5171	5171	5171	5171
mipd %	1	2	3	4	5	6	7	8	9	10

As mipd value increases, the number of itemsets generated increases. In the previous approaches [4, 5], only the minimum constraints of the itemsets are considered.

7. ACKNOWLEDGEMENT

The authors sincerely acknowledge the contribution of Ms. Kavita Kori (of SKSVMACET, Laxmeshwar, Karnataka, India) towards the implementation of the algorithm proposed. Also, the authors would like to convey thanks to anonymous reviewers for their valuable comments and suggestions.

8. CONCLUSION

In this work, an efficient distributed algorithm for mining rare itemsets has been proposed. The algorithm considers both minimum and maximum constraints of the itemsets to reflect the nature of the itemsets. In the previous approaches [4, 5], only the minimum constraints of the itemsets are considered. As compared to the algorithm [8], less number of rare itemsets is generated at each site in the proposed work. Also, bandwidth consumption is less. The reason is transmission of MIS-tree rather than entire dataset. Also, the proposed algorithm is scalable. It is planned to develop more compact form of MIS-tree which can in turn further reduce the bandwidth consumption.

9. REFERENCES

[1] Agrawal, R., Imielinski, T., Swami, A. 1993: *Mining association rules between sets of items in large databases.* In: ACM SIGMOD International Conference on Management of Data, vol. 22, pp. 207–216. ACM Press, Washington.

[2] Liu, B., Hsu, W., Ma, Y. 1999: *Mining Association Rules with Multiple Minimum Supports*. In: ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations, pp. 337–341

[3] R. Uday Kiran, P Krishna Reddy. 2010: *Mining Rare Association Rules in the Datasets with Widely Varying Items Frequencies* The 15th International Conference on Database Systems for Advanced Applications Tsukuba, Japan, April 1-4, 2010

[4] Hu, Y.-H., Chen, Y.-L, 2006: *Mining Association Rules with Multiple Minimum Supports: A New Algorithm and a Support Tuning Mechanism*. Decision Support Systems 42(1), 1–24

[5] Uday Kiran, R., Krishna Reddy, P. 2009: *An Improved Multiple Minimum Support Based Approach to Mine Rare Association Rules*. In: IEEE Symposium on Computational Intelligence and Data Mining, pp. 340–347

[6] J. Han, J. Pei, Y. Yin. 2000: *Mining frequent patterns without candidate generation*, Proceedings 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD' 00), Dallas, TX, USA,

[7] Taweechai Ouypornkochagorn Kitsana Waiyamai *Apriori_MSG-P* 2011*: A Statistic-Based Multiple Minimum Support Approach to Mine Rare Association Rules*. Proceedings of the Third International Conference on Knowledge and Smart Technologies

[8] Jutamas Tempaiboolkul. 2013: *Mining rare association rules in a distributed environment using multiple minimum supports* IEEE

[9] Frequent Itemset Mining Repository, http://fimi.cs.helsinki.fi/data/