An Improved Distributed Mining Algorithm of Association Rules

Wang Ailing
Department of Mathematics, Heze University, Heze, 274000, China
Email: fly123406@163.com
doi:10.4156/jcit.vol6. issue4.14

Abstract

Large amount of data is stored in distributed web node as Internet and distributed database develops, it is impossible to store these data in single node on account of communication, efficiency and security. Through research was conducted on mining association rules in the distributed database system and classical Apriori algorithm was extended based on transactional database system. Then an efficient algorithm that get association rules from frequent itemsets was presented. Experiment results show that the algorithm has sound extension, short time complexity and small communication cost.

Keywords: *Distributed Database system, Data Mining, Association Rules, Efficient*

1. Introduction

Automated acquisition of knowledge is most important in Artificial Intelligence. If-then rules are well-known techniques for knowledge representation traditionally, classification methods of machine learning or data mining are always used for knowledge discovering. They commonly deal with attributes without considering ordinal information. However, we may come across many practical applications such as ranking of consumer products, ranking of universities and multi-criteria decision problem and etc. Data mining is the process of non-trivial extraction of implicit, previously unknown and potentially useful information from data [1]. Mining association rules is one of the most important aspects in data mining. Association rules are dependency rules which predict occurrence of an item based on occurrences of other items, it is simple but effective and can help the commercial decision making like the storage layout, appending sale etc.

We usually use distributed system as a solution to mining association rules when mass data is being collected and warehoused. With the development of web and distributed techniques, we begin to store databases in distributed systems. Thus researches on mining association rules algorithm in distributed system become more important, which also have a broad application foreground. Distributed algorithm has characters of high adaptability, high flexibility, low wearing performance, easy to be connected etc.

2. Research methodology

Suppose $I=\{I_1,I_2, I_3, I_4, I_5, I_6 \ldots I_m\}$is the set of the goods, for given transaction database D, each transaction has an unique transaction mark TID and an itemset ($itemset \subseteq I$) . Association rule is an implication expression of the form $X \rightarrow Y$, where X and Y are itemsets and $X \cap Y=\Phi$.

Rule evaluation metrics:

Support is the fraction of transactions that contain both X and Y.

Confidence is to measure how often items in Y appear in transactions that contain X.

Frequent itemset is ann itemset whose support is greater than or equal to a minsup threshold.

Given a transaction database D that consists of a set of transactions T, the goal of association rule mining is to find all rules having support $\geq$ minsup threshold and confidence $\geq$ minconf threshold.

Two-step approach for mining association rules can be concluded as following.

- Frequent Itemset Generation. Generate all Itemsets whose support $\geq$ minsup.
- Rule Generation. Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset.

Frequent itemset generation is still more computationally expensive. So we'll concentrate on the first step in the following discussion.

Suppose a transaction database D is stored distributed in n spots $S_i(1 \le i \le n)$, $D=\{D_1,D_2 , \dots , D_n\}$,D is the global database and $D_i(1 \le i \le n)$ is the local database. $|D_i|$ is the number of transactions in local database and |D| is the number of transactions in global database. For some itemset I, I.sup(i) and I.sup is respectively the support of I in database D_i and D. If $I.sup(i) \ge |D_i| \times$ minsup then I is local frequent itemset. If $I.sup \ge |D| \times$ minsup then I is global frequent itemset.

Theorem: Suppose $LL_i(k)$ is the local frequent K-itemset in S_i, $L_i(k)$ is the global frequent K-itemset then $L_i(k) \subseteq \bigcup_{i=1}^{n} LLi(k)$ [2].

The above theorem indicates that global frequent itemsets must can be derived from local frequent itemsets. In distributed database system, getting global candidate frequent itemsets from local frequent itemsets can greatly reduce the number of candidate frequent itemsets.

3. AprTidRec algorithm

Main algorithms for mining association rules consist of Apriori[3], DHP[4], Partitionetc[5]. AprTidRec proposed is similar to Apriori in this paper, the difference between them is that Apriori includes joint step and pruning step while ApriTidRec include only joint setp when generate frequent itemset. In AprTidRec, a record structure called tidRec is defined for each candidate frequent itemset. The tidRec of itemset I consist of TID of the transactions who contain itemset I. I.tidRec is the tidRec of itemset I. The tidRec of 1-itemset can be got by scanning the transaction database. The structure of the record in the algorithm is < I , tidRec, count >, I.count is the support of the itemset I, it is equal to the length of tidRec that is count=| tidRec |. When generating candidate frequent k-itemset from frequent k-1-itemset, the tidRec and support of the candidate frequent k-itemset can be derived from the intersection of the tidRec of the two k-1-itemsets.

AprTidRec-algorithm description: (C_k is candidate frequent k-itemset, L_k is frequent k-itemset)

Procedure get_association_rule:

Begin

a) $K=1, Lk= \Phi$
b) for all itemsets $I_1 \in L_{k-1}$ do begin
c) for all itemsets $I_2 \in L_{k-1}$ do begin
d) If $I_1.item_l=I_2.item_l \wedge I_1.item_2=I_2.item_2 \wedge \dots \wedge I_1.item_{k-2}=I_2.item_{k-2} \wedge I_1.item_{k-1} < I_2.item_{k-1}$;
e) then
f) begin
g) $C_k.itemsets = I_1.item_1.I_1.item_2 \dots I_1.item_{k-1}.I_2.item_{k-1}$
h) $C_k.tidRec = I_1.tidRec \cap I_2.tidRec$
i) $C_k.count = | C_k.tidRec |$
j) end
k) if($C_k.count \ge | D | *$ minsup)then
l) $L_k = L_k \cup \{ C_k \}$
m) end
n) end

End

From the above algorithm, we can know that when generate global frequent k-itemset we scan the local databases only once(during constructing the new storage structure) and prune step doesn't need. So I/O spending is saved, and time complexity of the algorithm is reduced and efficiency is improved. But reduction of time complexity is at the cost of increase of space complexity. Each candidate itemset need a tidRec structure in the algorithm so large of memory space is required if transaction database is

huge. How to balance between the time and space complexity to get the best solution need yet further study.

4. System Realization

Using global knowledge database and local global communication mode, distributed association rules mining system can be constructed as shown in Fig. 1.

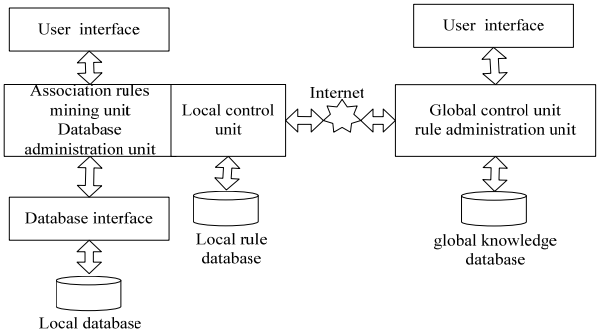

Figure 1. System structure

Local websites generate frequent itemset from local database. Global website is responsible for generating the superset of all local frequent itemsets and computing the global support of each itemset then confirming the global frequent itemsets. Finally, association rules based on the global database can be acquired and stored in the global knowledge database according to the global frequent itemsets and minsup given by the user. Detail realization steps are as follows.

Step 1

- Generate global frequent 1-itemset by scanning all local databases and exchanging support count between local websites.

Step 2

Generate frequent k-itemsets(k>1) by:

- Local websites generate local frequent itemsets based on local database by AprTidRec
- Local websites send local frequent itemsets and their support to global website
- Global website combines all the frequent itemsets from local websites then generates candidate global frequent itemsets
- Global website broadcast candidate global frequent itemsets, and local website determines newly added itemsets (in candidate itemsets but not in local frequent itemsets) and their local support
- Local websites sends newly added itemsets and their local support to global website, support is accumulated at global website thus the global support of each global frequent itemset is acquired
- Global website generate global frequent itemsets according to minsup given by the user.

5. Experiment and analysis

A *Experiment*

Experiment was carried out on Pentium IV 1.6G , 512M PC to improve efficiency of AprTidRec algorithm. Database in the experiment is junk mail database that get from the webisite http://www.ics.uci.edu/~mlearn. Name and corresponding parameters of two different test datasets are shown in Table 1 and Table 2.

Table 1. Parameters Definition

|D|	Number of records in the dataset
|I|	Total number of itemsets
|T|	Average number of itemsets in each record

Table 2. Parameters' setting

dataset	**|D|**	**|I|**	**|T|**
T1	1805	20	8

Run time of these algorithms based on dataset T1 is shown in Figure 2.

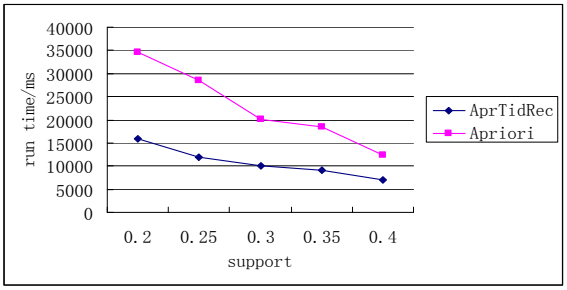

Figure 2. Run time based on T1

B *Result*

From the above experiment, we can see that run time of Apriori and AprTidRec algorithm both increase with the decreasing of support, while run time of AprTidRec increase less than Apriori. Generally speaking, the ApriTidRec has less run time than Apriori totally.

6. Conclusion

New approach for mining rules in order from ordinal distributed database system was demonstrated in the paper. Association rules mining in distributed database is an important aspect in the field of data mining. A new algorithm based on AprTidRec was discussed in the paper, the efficiency of which is verified. Realization of overall system was also given based on the presented algorithm. Improvement on how to improve efficiency and deal with distributed non-isomorphic data sources is to be studied in the future. And practical applications will be discussed.

7. References

[1] Lan H. Witten, Eibe Frank, "Data Mining",China Machine Press, Beijing, 2003.
[2] ShiZhongZhi ,"Knowledge Discovery", Tsinghua Press, Beijing, 2002.
[3] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", Proceedings of the ACM SIGMOD Conference on Management of data, Washington D C, pp.207-216,1993.
[4] J.S. Park, M. Chen , and P.S Yu, "An effective hash-based algorithm for mining association rules", Proc of the ACM SIGMOD Int'l Conf on Management of Data, Jose, CA, pp.175-186,1995.

[5] Ashok Savasere, Edward Omiecinski, and Shamkant Navathe,"An efficient algorithm for mining association rules in large database", Proceeding of the VLDB conference, Zurich Switzerland, pp.432-444,1995.
[6] KARGUPTA H, PARK B, and HERSHBERGER D, "Collective datamining:a new perspective toward distributed data mining", Proc of Advanced in distributed data mining, AAAI/MIT, pp.133-184 , 2000.
[7] Fayyad U, Piatetsky-Shapiro G, "From data mining to knowledge discovery in databases", AI Magazine, vol.17, no.3, pp.37~54 ,1996.
[8] Wilson R, Marinez T, "Improved heterogeneous distance functions", Journal of Artificial Intelligence Research, vol.6, no.1, pp.1-34, 1997.
[9] Li Fan, Yudong Zhang, Zhenyu Zhou, David P. Semanek, Shuihua Wang, Lenan Wu, "An Improved Image Fusion Algorithm Based on Wavelet Decomposition", JCIT: Journal of Convergence Information Technology, Vol. 5, No. 10, pp. 15 ~ 21, 2010.
[10] Huang Jie, Huang Bei, Huang Qiucen, "An Improved Dynamic Load Balancing Algorithm for a Distributed System in LAN", JCIT: Journal of Convergence Information Technology, Vol. 5, No. 10, pp. 91 ~ 98, 2010.
[11] Massimo Orazio Spata, Salvatore Rinaudo , "A scheduling Algorithm based on Potential Game for a Cluster Grid ", JCIT: Journal of Convergence Information Technology, Vol. 4, No. 3, pp. 34 ~ 37, 2009.