Fast Mining Algorithm of Association Rules Base on Cloud Computing

Bo He[1, 2]

[1]School of Computer Science and Engineering, ChongQing University of Technology, ChongQing, 400054, China

[2]Shenzhen Key Laboratory of High Performance Data Mining, Shenzhen, 518055, China

heboswnu@sina.com

Keywords: Association rules, Frequent itemsets, FP-tree, FP-growth

Abstract. There were some problems in traditional mining algorithm of association rules: a lot of candidate itemsets and communication traffic. Aiming at these problems, this paper proposed a fast mining algorithm of association rules based on cloud computing, namely, FMAAR algorithm. Firstly, the frequent items were found. Secondly, the FP-tree was created and the frequent itemsets were mined by FP-growth algorithm. Finally, the association rules were got by cloud computing. The experimental results suggest that FMAAR algorithm is fast and effective.

Introduction

There are some mining algorithms of association rules[1], such as Apriori[2] and Partition[3]. However, the database is generally large, traditional mining algorithms consume much time. Aiming at the problem, the paper proposes a fast mining algorithm of association rules based on cloud computing, namely, FMAAR algorithm. The key of mining association rules is to find frequent itemsets.

Related definition and theorem

Description of mining association rules

The database is *DB*, the total number of tuples is *M*. Mining association rules can be described as follows: finding frequent itemsets and getting association rules.

Related Definition

Definition 1 For itemsets *X*, the number of tuples which contain *X* in database is frequency of *X*, symbolized as $X.s$.

Definition 2 For itemsets *X*, if $X.s \geq min\_sup*M$, then *X* are defined as frequent itemsets, symbolized as *F*.

Related Theorem

Theorem 1 If itemsets *X* are frequent itemsets, then any nonempty subsets of *X* are also frequent itemsets.

Proof: If itemsets *X* are frequent itemsets of *DB*, then $X.s \geq min\_sup \times M$, if $Y \subseteq X$, then $Y.s \geq X.s$, hence $Y.s \geq X.s \geq min\,sup \times M$, *Y* are frequent itemsets of *DB*.

Corollary 1 If itemsets *X* are not frequent itemsets, then superset of *X* must not be frequent itemsets.

Theorem 2 If item x is not frequent item, and $\{x\} \subseteq X$, then itemsets *X* must not be frequent itemsets.

Proof: If item x is not frequent item, then itemsets $\{x\}$ are not frequent itemsets. If $x \in X$, then $\{x\} \subseteq X$. According to corollary 1, *X* must not be frequent itemsets.

FP-tree and FP-growth algorithm[4]

Definition 3 FP-tree is a tree structure defined as follow.

(1) It consists of one root labeled as ”null”, a set of itemset prefix subtrees as the children of the root, and a frequent itemset header table.

(2) Each node in the itemsets prefix subtree consists of four fields: item-name, count, parent and node-link.

(3) Each entry in the frequent-item header table consists of three fields: i,Itemname. ii, Side-link, which points to the first node in the FP-tree carrying the item-set. iii, Count, which registers the frequency of the item-name in the transaction database.

FP-growth algorithm adopts a divide-and-conquer strategy. It only scans the database twice and does not generate candidate itemsets. The algorithm substantially reduces the search costs. The study on the performance of the FP-growth shows that it is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm.

FMAAR algorithm

The step of FMAAR is described as follows.

(1) The frequent items are found.

(2) The FP-tree is created and the frequent itemsets are mined by FP-growth algorithm.

(3) The association rules are mined by cloud computing.

The pseudocode of FMAAR is described as follows.

Alogrithm FMAAR

Input: The transaction database *DB*, the total number of tuples M , the minimum support threshold *min_sup* and confidence level *cl.*

Output: The frequent itemsets *F* and the association rules *A.*

Step1. /* The frequent items are found.*/

$E=\varnothing$; /**E* is all frequent items*/

Scanning *DB* once;

computing frequency *x.s* of each items *x* ;

if *x.s* >=*min_sup***M*

{*x* are the frequent items;

$E=E\cup\{x\}$;

}

E is sorted in the order of descending support count;

Step2. /* The FP-tree is created and the frequent itemsets are mined by FP-growth algorithm.*/

creating the FP-tree;

F=FP-growth(FP-tree, null);

Step3. /* The association rules are mined by cloud computing.*/

the association rules *A* are mined by *F and* confidence level *cl.*

Example of FMAAR algorithm

The database *DB*, as show in table 1. *Min_sup* is the minimum support threshold, *min_sup*=0.4.

Table 1 database *DB*

database	ID	Transaction
DB	100	*a,b,c,k,m,f,e,l,p*
	101	*c,k,b,m,o,q*
	102	*a,b,c,d*

According table 1 and *min_sup*=0.4, all frequent items can be got. All frequent items are sorted in the order of descending support count. As shown in table 2.

Table 2 The frequent items and support count

Frequent Items	Support count	Frequent Items	Support count
c	3	*b*	3
a	2	*m*	2

k	2		

All frequent items $E=\{c, b, a, m, k\}$,the FP-tree is constructed according to E, as shown in figure 1. According to Theorem 2, If item x is not frequent item, and $\{x\} \subseteq X$, then itemsets X must not be frequent itemsets. Hence the FP-tree only contains frequent items. As show in fig.1.

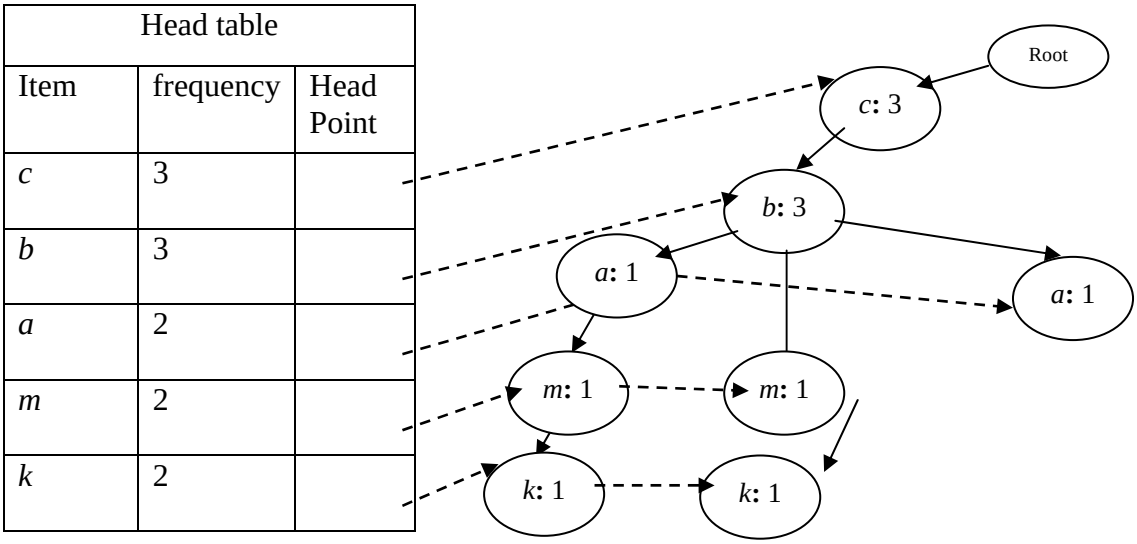

Fig.1 *FP-tree*

The frequent itemsets are computed by FP-growth algorithm and *FP-tree*. $F=\{\{c, b, a\}, \{c, b, m, k\}, \{c, b\}, \{c, a\}, \{b, a\}, \{c, b, m\}, \{c, b, k\}, \{c, m, k\}, \{b, m, k\}, \{c, m\}, \{b, m\}, \{c, k\}, \{b, k\}, \{m, k\}\}$.

The association rules are mined by *F and* confidence level.

Experiments of FMAAR

This paper compares FMAAR with classical algorithm Apriori. The experimental data comes from the sales data in July 2010 of a supermarket.

Comparison experiment: It is a way of changing the minimum support threshold. FMAAR compares with Apriori in terms of communication traffic and runtime. The results are reported in Fig.2 and Fig.3.

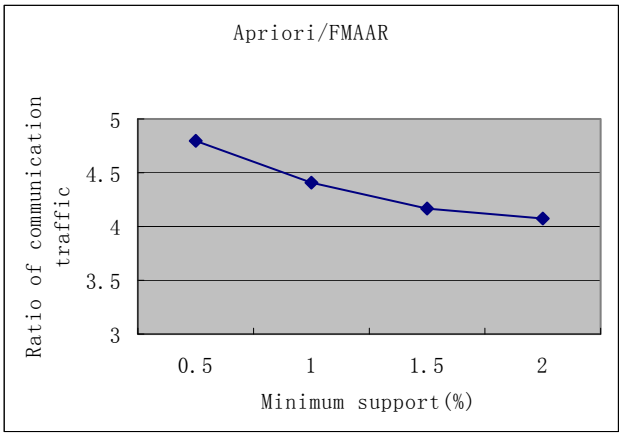

Fig.2. Comparison of communication traff

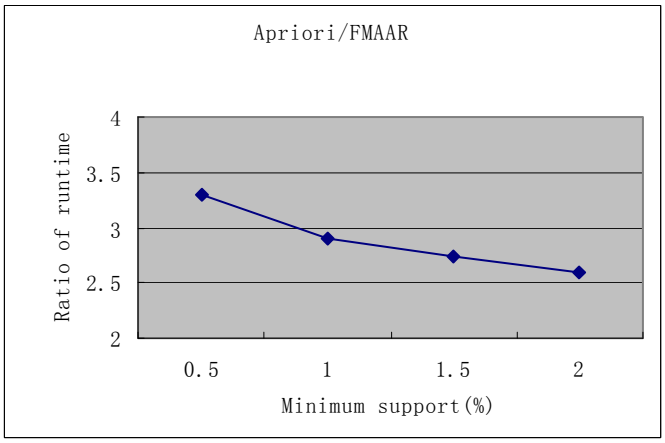

Fig.3. Comparison of runtime

The comparison experiment results indicate that under the same minimum support threshold, communication traffic and runtime of FMAAR decreases while comparing with Apriori.

Conclusion

The paper proposes a fast mining algorithm of association rules based on cloud computing, namely, FMAAR algorithm. The experimental results suggest that FMAAR algorithm is fast.

This research is supported by shenzhen key laboratory for high performance data mining with shenzhen new industry development fund under grant No.CXB201005250021A.

References

[1] Chen ZB, Han H, Wang JX. Data warehouse and data mining[M]. Beijing: Tsinghua University Press, 2009. (in Chinese)

[2] Agrawal R, Srikant R. Fast algorithms for mining frequent itemsets[C]. In: Proceedings of the 20th International Conference Very Large DataBase, Santiago, Chile, 1994, pp. 487-499.

[3] Savasere A, Omiecinski E, Navathe SM. An efficient algorithm for mining frequent itemsets. In: Proceedings of the 21th International Conference on VLDB, Zurich, 1995, pp.432-444.

[4] Han J W, Pei J, and Yin Y. Mining frequent patterns without Candidate Generation[C]. Proceedings of the 2000 ACM SIGMOD international conference on Management of data. ACM Press, 2000, pp.1-12.

[5] He B, Wang HQ, Liu Z, Wang Y. A fast and parallel algorithm for mining frequent itemsets[J]. Journal of Computer Applications, 2006, 26(2) 391-392. (in Chinese with English abstract)