Proceedings of the 2012 2nd International Conference on Computer and Information Application (ICCIA 2012)

Optimized algorithm for Mining Valid and non-Redundant Rules

Naili Liu
Department of Information
Linyi University
Linyi,276005,China
lnl1999@163.com

Lei Ma
Department of Media
Linyi University
Linyi, 276005,China

Abstract—**The traditional algorithm of mining association rules, or slowly produces association rules, or produces too many redundant rules, or it is probable to find an association rule, which posses high support and confidence, but is uninteresting, and even is false. Furthermore, a rule with negative-item can't be produced. This paper puts forward a new algorithm MVNR(Mining Valid and non-Redundant Association Rules Algorithm),which primely solves above problems by using the minimal subset of frequent itemset.**

Keywords-data mining;association rule;correlation; redundancy;efficient;negative

I. INTRODUCTION

Mining association rules is an important part of the data mining research, which reflects interesting association or contact between itemsets of the large amounts of data. The definition of the association rule: Let I = {i_1, i_2, ..., i_m} is composed of a collection of m different items. Given a transaction database D, where each transaction T is a collection of I, that is T⊆I, T has a unique identifier TID. If A⊆I and A⊆T, then transaction T contains itemset A.

Association rules is an implication like $A \Rightarrow B$, A⊂T,B⊂T and A∩B=$\varnothing$, if the percentage of containing A∪B is s in D, then s is the support of $A \Rightarrow B$;if the percentage of containing a and also containing B is c in D,then c is called the credibility of $A \Rightarrow B$. The problem of mining association rule is to find all the association rules with a given minimum support min_sup and minimum confidence min_conf in transaction database D, that is, the support and confidence of association rules, respectively, is not less than min_sup and min_conf.

The discovery of association rules can be decomposed into two steps: (1) to find all frequent itemsets;② to produce credible association rules.At present research of association rule is mainly focused on the first step, the second step is less, but the rules generated contain a large number of redundant rules, especially when the itemset contains many items, the generated redundant rules are growing exponentially. The method of generating rule is simple in Apriori algorithm [1], but it has the computational complexity of the rule and exists redundancy,so it can not guarantee the validity of the rule.The method of using adjacency directed acyclic graph in [2,3,4] obtains frequent itemsets by using ancestors to eliminate the redundant association rules. This method is low efficient because it need establish adjacencies directed acyclic graph of every frequent itemset. This method need large memory space, especially when itemset's length is very long. So it is low efficient and can not eliminate wrong or false rules. The method of using frequent closures itemset in [5, 6, 7, 8, 9] uses itemsets with frequent closures to avoid generating redundant association rules. Some algorithms have made improvements, but they are based on demand closure itemsets to generate association rules, this needs to be repeated to scan the database to remember TID which includes itemset, but also it can not eliminate wrong or false rules. So this method need scan database repetitively and can not eliminate wrong or false rules. The method of using related support and interest to mining association rules in[10,11,12,13,14]can mine correct rules,but can not eliminate redundant rules. The algorithm proposed in this paper not only can mine non-redundant association rules, and mine effective and correct association rules,can also find out association rules containing negation.

II. BASIC CONCEPTS AND THEOREMS

We still adopt the definitions of redundant rules, simple redundancy, strict redundancy in[4] and the definitions of related support and negate itemsets in [12].

Defintion 1 A rule is necessary if it is not simply redundant and strictly redundant relative to any other rules.

Definition 2 Let X,Y be Frequent itemsets,if sup(Y)≤sup(X)/c, and there doesn't exist a frequent itemset Z satisfying Z ⊂ Y and sup (Z)≤sup (X)/c,then Y is a minimal set of X.

Theorem 1 Let X,Y be frequent itemsets, and Y is a minimal set of X, the rule $Y \Rightarrow X - Y$ is not simply redundant relative any other rules.

Proof: Assuming that $Y \Rightarrow X - Y$ is simply redundant relative $C \Rightarrow D$,according to the definition of simple redundancy, there exists C∪D = X and C⊂Y, then C is a proper subset of Y and sup (C) ≤sup (X)/c is correct, but this contradicts that Y is a minimal set of X, so the assumption is not true, the original conclusion is correct.

Definition 3 Let X be a frequent itemset, all minimal sets of X are called the minimal set collection of X, denoted by F(X,c).

Theorem 2 Let X be a frequent itemset, X_1, X_2, ..., X_k is a superset of X and Xi ∈ L(1≤i≤k), Y ∈ F(X,c)-

$\cup_{i=1}^{k} F(X_j, c)$,then the rule $Y \Rightarrow X - Y$ is not strictly redundant relative any other rules.

Proof: Assuming that $Y \Rightarrow X - Y$ is strictly redundant relative $C \Rightarrow D$,according to the definition of strict redundancy, there exists X⊂C∪D and C⊆Y, then C is a subset of Y and C is a proper subset of C∪D, but C can not be a proper subset of Y when the confidence level of c because Y is a minimal subset of X, there must be C=Y, then sup (Y) ≤sup (C ∪ D)/c is correct, but any Xi, $Y \notin F(X_i, c)$,that is, $sup(Y)>sup(X_i)/c$, sup(Y)>sup(C ∪ D)/c is correct because C∪D is a super set of X, so the assumption is not true, the original conclusion is correct.

Theorem 3 Let X be a frequent itemset, $X_1, X_2, ..., X_m$ is a subset of X and $X_i \in L$ (1≤i≤m), if sup (X) /c≥max_sup (max_sup is the maximum value of all frequent 1-itemsets support), then any X_i exists $F(X_i,c) \subseteq F(X,c)$;if sup (X)/c<max_sup, then any Xi which satisfies sup (X) = sup (X_i), $F(X_i,c) \subseteq F(X,c)$ is correct.

Proof: $Y \in F(X_i,c)$,Y is a minimal subset of X_i, then Y is a subset of X, and sup (Y) ≤max_sup≤sup (X)/c≤sup (X_i)/c is established. Let us prove that Y is also a minimal subset of X. Suppose that there exists a set Z which is a proper subset of Y, then sup (Y)≤sup (Z), so sup (Y) ≤sup (Z) ≤max_sup≤sup (X) / c≤sup (X_i) / c, so Z is a minimal subset of X_i, but this contradicts that Y is a minimal set of X_i, so the assumption is not true, Y∈ F(X,c) is correct, that is, $F(X_i,c) \subseteq F(X,c)$ is established. If sup (X) / c <max_sup and $Y \in F(X_i,c)$, then sup(Y) ≤sup(X_i) / c, and also because of sup(X)=sup(X_i), so sup (Y) ≤sup (X) / c is established. Next we prove Y∈ F(X,c), assuming that there exists a proper subset Z of Y,Z is a minimal subset of X, then sup(Z) ≤sup(X) / c is correct, then sup(Y) ≤sup(Z) ≤sup(X) /c =sup(X_i) /c is established, that is ,Z is a minimal subset of X_i, but this contradicts that Y is a minimal subset of X_i, so the assumption is not true, so Y Y∈ F(X,c) is correct, then $F(X_i,c) \subseteq F(X,c)$ is established.

As can be seen by the above definitions and theorems, if X is a frequent itemset and exists sup (X) / c≥max_sup, then we need not consider all subset of X when generate association rules; if there exists sup (X) / c <max_sup, all the subset of X with the same support need not be considered. So, we firstly filter frequent itemsets by theorem when we generate association rules, which can improve the efficiency of generating association rules.

III. MINING MAXIMUM FREQUENT ITEMSETS ALGORITHM

A. The basic idea of the algorithm

The basic idea of the algorithm is: Firstly, filter the frequent itemsets L,delete frequent itemsets which then can only generate redundant association rules, then get new frequent itemsets L'; secondly, producing minimal subset of every frequent itemset in L'; then analyse any frequent itemset Li in L':first delete elements of minimal subset of Li which belong to the elements of mimimal subset of Li's superset ,then the rest of each minimal subset Y∈ L', generate rule $Y \Rightarrow L_i - Y$,if the relevant support of this rule is greater than 1, then this rule is added to the rule set R; if relevant support is less than 1, then negative rule $L_i - Y \Rightarrow \overline{Y}$ is generated, then we determine whether is this negative rule's support and confidence greater than the user-defined minimum support and minimum confidence, and relevant support is greater than 1, then the rule is added to the rule set R.

B. Algorithm decription

1) Main Algorithm

Generate efficient and non-redundant association rules algorithm:

Input: frequent itemsets L, minimum support s, minimum confidence c, max_sup;

Output: effective and non-redundant association rules R;

(1) L'=Reduce_L(L,max_sup,c); //delete frequent itemsets in L which can only generate redundant association rules
(2) for each $L_i \in L'$ do
(3) $F(L_i,c)$=FindMinimalSubset(L_i,c);//find minimal subset of every frequent itemset in L'
(4) R=∅;//association rules set
(5) for each $L_i \in L'$ do
(6) {
(7) $P(L_i,c)=F(L_i,c)$;
(8) for each $L_j \in L'$ of L_i's superset do
(9) $P(L_i,c)=P(L_i,c)-F(L_j,c)$;
(10) for each itemset $Y \in P(L_i,c)$ do
(11) if $sup(L_i)/(sup(Y)*sup(L_i-Y))>1$ then
(12) $R=R\cup\{Y \Rightarrow L_i - Y\}$
(13) else if $sup(L_i)/(sup(Y)*sup(L_i-Y))<1$ then
(14) {
(15) compute the support of L_i-Y) ∪ $\overline{Y}$,sup((L_i-Y)∪ $\overline{Y}$);
(16) compute the support of $\overline{Y}$,sup($\overline{Y}$);
(17) conf=sup((L_i-Y)∪ $\overline{Y}$)/sup(L_i-Y);
(18) corr=conf/sup($\overline{Y}$)
(19) if sup((L_i-Y)∪ $\overline{Y}$)≥s and conf≥c and corr>1 then
//if rule's relevant is less than 1,judge its negative rule
(20) $R=R\cup\{L_i - Y \Rightarrow \overline{Y}\}$
(21) }
(22) }

2) Reduce_L algorithm

Simplify the collection of frequent items algorithm Reduce_L:

In algorithm first let L be assigned to H, then delete frequent 1-itemset because frequent 1-itemset can not generate association rules, and then analyze each frequent itemset remaining in H. Let X be frequent itemset, if sup (X) / c <max_sup, then delete all frequent itemset in H with the same support as H; if sup (X) / c max_sup, then delete all subset of X in H, and finally return H which is filtered.

```
procedure Reduce_L(Frequent Itemset:L,max_sup,
confidence:c)
{
    H=L;
    delete frequent itemsets which contain only one
item ,that is, frequent 1-itemset;
    for each h∈ H do
     {
        if sup(l)/c ≥ max_sup then
            delete all subsets of h in H
        else
            delete all subsets with the same support as
h
                        of h in H
    }
    return H;
}
```

3) FindMinimalSubset algorithm

Find a minimal subset of frequent itemsets algorithm FindMinimalSubset:

In FindMinimalSubset algorithm, first find out all subsets of X which its support is less than or equal to sup (X) / c in L, let its result be assigned to H, each frequent itemset h in H, if there does not exist subset of h in H, then h was added to the minimal subset collection.

```
procedure FindMinimalSubset(Frequent item:X,
confidence:c)
{
    MinimailSubset=∅;
    H=the subsets of X which its support is less than or
     equal to sup(X)/c in L;
    for each h∈ H do
      if no subset of h in H then
        MinimalSubset=MinimalSubset ∪ {h}
}
```

4) Algorithm example

Transaction database D in Figure 1, user-defined minimum support s = 2/8, minimum confidence c = 50%.

TID	Items
001	A C D
002	B C E
003	A B C E
004	B E
005	A B C E
006	B E
007	B E
008	C

Figure 1 transaction database D

According to minimum support and mining frequent itemsets' algorithm, we produce frequent itemsets L= {A, B, C, E, AB, AC, AE, BC, BE, CE, ABC, ABE, ACE, BCE, ABCE}. According to the algorithm in this pager, first filter L by Reduce_L algorithm, get new frequent itemsets L'= {AC, BCE, ABCE}, then call FindMinimalSubset algorithm to generate a minimal subset of every frequent itemset in L', get F (AC, 0.5) = {A, C}, F (BCE, 0.5) = {B, C, E}, F, (ABCE, 0.5) = {A, BC, CE,}. According to above results, we generate rules, judge every rule's support and confidence whether or not are greater than or equal to minimum support minimum confidence each other and whether or not its relevant support is greater than 1, if rule's relevant support is less than 1, we judge its negative rule. Finally generated association rules is shown in Figure 2.Among these rules, the rule $C \Rightarrow BE$ relevant support is 4/5 which is less than 1,then we consider its negative rule $BE \Rightarrow \overline{C}$,its support is 3/8 and its confidence is 50%,and its relevant support is 4/3, this rule satisfies requirement, let be added to association rules set R.

Itemset	Rules	Support	Confidence	Correlation
ABCE	A=>BCE	2/8	67%	16/9
	BC=>AE	2/8	67%	8/3
	CE=>AB	2/8	67%	8/3
BCE	B=>CE	3/8	50%	4/3
	E=>BC	3/8	50%	4/3
	$BE \Rightarrow \overline{C}$	3/8	50%	4/3
AC	C=>A	3/8	60%	8/5

Figure 2 generated association rules according to the algorithm

We can generate association rules which these relevant support is greater than 1 according to Apriori algorithm, as is shown in Figure 3.According to definition and theorem, many rules are invalid or redundant, and we can not generate negative rules.

Itemset	Rules	Support	Confidence	Correlation
AC	A=>C	3/8	100%	1.6
	C=>A	3/8	60%	1.6
BE	B=>E	3/8	100%	1.3
	E=>B	3/8	100%	1.3
ABC	A=>BC	2/8	67%	1.78
	AB=>C	2/8	100%	1.6
	BC=>A	2/8	67%	1.78
ABE	AB=>E	2/8	100%	1.3
	AE=>B	2/8	100%	1.3
ACE	A=>EC	2/8	67%	1.78
	AE=>C	2/8	100%	1.6
	EC=>A	2/8	67%	1.78
BCE	B=>CE	3/8	50%	1.3
	E=>CB	3/8	50%	1.3
	CB=>E	3/8	100%	1.3
	CE=>B	3/8	100%	1.3
ABCE	A=>BCE	2/8	67%	1.78
	AB=>CE	2/8	100%	2.67
	AE=>BC	2/8	100%	2.67
	BC=>AE	2/8	67%	2.67
	CE=>AB	2/8	67%	2.67
	ABE=>C	2/8	100%	1.6
	ABC=>E	2/8	100%	1.3
	ACE=>B	2/8	100%	1.3
	BCE=>A	2/8	67%	1.78

Figure 3 generated association rules according to Apriori algorithm.

IV. Conclusion

In this paper we presented a new algorithm, which can not only generate non-redundant and valid association rules, but also can not generate false or erroneous rules, and can generate negative rules, generated rules can help analyzing problems and making decisions. In this algorithm, first analyze frequent itemsets and delete frequent itemsets which only can generate redundant rules, we adopt the inclusion relations between collection when we find out the minimal subset of frequent itemset. According to the experimental results show that the algorithm is efficient and saves memory space, and generated rules are valid and non-redundant.

References

[1] R.Agrawal, Imielinski T, Swami A. Mining association rules between sets of items in large databases (C). In: Buneman P, Jajodia S,eds. Proc. of the ACM SIGMOD Conf. on Management of Data (SIGMOD'93). New York: ACM Press, 1993. 207~216.

[2] C.C.Aggarwal, Z.Sun,and P.S. Yu,Online Generation of Profile Association Rules,Proc. KDD Conf.,1998.

[3] Charu C. Aggarwal , Philip S. Yu, A New Approach to Online Generation of Association Rules, IEEE Transactions on Knowledge and Data Engineering, v.13 n.4, p.527-540, July 2001

[4] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In 7th Intl. Conf.on Database Theory, Jan. 1999.

[5] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices.Information Systems, 24(1):25–46, 1999.

[6] ZHAO, Y., ZHANG, C., AND ZHANG, S. 2006. Efficient frequent itemsets mining by sampling. In Proceeding of the 2006conference on Advances in Intelligent IT: Active Media Technology 2006. IOS Press, Amsterdam, The Netherlands, 112–117.

[7] M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed association rule mining. Technical Report 99-10,Computer Science Dept., Rensselaer Polytechnic Institute,October 1999.

[8] Mohammed J. Zaki, Generating non-redundant association rules, Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, p.34-43, August 20-23, 2000, Boston, Massachusetts, United States.

[9] Yves Bastide , Nicolas Pasquier , Rafik Taouil , Gerd Stumme , Lotfi Lakhal, Mining Minimal Non-redundant Association Rules Using Frequent Closed Itemsets, Proceedings of the First International Conference on Computational Logic, p.972-986, July 01, 2000

[10] R. J. Bayardo and R. Agrawal. Mining the most interesting rules. In 5th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, Aug. 1999.

[11] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. In 3rd Intl. Conf. Information and Knowledge Management, pages 401–407, Nov. 1994.

[12] Sergey Brin, Rajeev Motwani, Craig Silverstein. Beyond Market Baskets: Generalizing Association Rules to Correlations.In Proc.1997 ACM - SIGMOD Int. Conf. Management of Data (SIGMOD'97), pages 265-276.

[13] CHANDRA, B. AND BHASKAR, S. 2011. A new approach for generating efficient sample from market basket data. Expert Systems with Applications 38, 3, 1321-1325.

[14] CHAKARAVARTHY, V. T., PANDIT, V., AND SABHARWAL, Y. 2009. Analysis of sampling techniques for association rule mining. In Proceedings of the 12th International Conference on Database Theory. ICDT '09. ACM, New York, NY, USA,276–283.

Published by Atlantis Press, Paris, France.