ISSN: 2250–3676

[IJESAT] INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE & ADVANCED TECHNOLOGY Volume-2, Issue-5, 1447 – 1453

PERFORMANCE EVALUATION BETWEEN APRIORI AND IMPROVED APRIORI

B.Yugandhar[1], M. Raghu Varma[2], K.Karthik[3], T.Rajkumar[4], G.Bhaskar Phani[5]

[1] *Assistant professor, Dept of Computer Science and Engineering, Swarnandhra Institute of Engg & Technology, Seetharampuram, Narsapur 534280.*

[2, 3, 4,5] *Student,Dept of Computer Science and Engineering, Swarnandhra Institute of Engg.& Technology, Seetharampuram, Narsapur 534280.*

Abstract

With massive amounts of data being collected and stored, many industries are becoming interested in mining association rules from their databases. The discovery of interesting association relationships among huge amounts of business transaction records can help in many business decision making processes such as marketing, catalog design etc.. In this respect Association rule mining is considered as one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories.. This paper presents a comparison between classical frequent pattern mining algorithms that use candidate set generation and test the algorithms without candidate set generation. Here we present three algorithms Apriori, and Improved Apriori.

----------***----------

1. INTRODUCTION

With the development of the information technology and application of database. The collected data far exceed people's ability to analyze it. Thus new and efficient methods are needed to discover knowledge from large databases. And data mining appeared. Data mining is the core of knowledge discovery in databases. It's a procedure to find the useful and potentially knowledge in database. Association rules are one of the most important knowledge of data mining's result which can be defined as the relation and dependency between the itemsets by given support and confidence in database. In the algorithms of the association rules mining, apriori is the ancestor which offered by Agrawal R in 1993. The main idea of the apriori is scanning the database repeatedly. With the theory that the subset of the frequent itemset are frequent itemsets too, you can gain the length of frequent (k+1)-itemsets L_{k+1} from the frequent k-itemsets L_k. At the k time it scanned the database only the candidate itemsets C_{k+1} which generate from the L_k was concerned. Later the appear times of the C_{k+1} can be verified by another scanning database.

There was a drawback that cost much time and memory to generate candidate C_{k+1} . It would be more worst when the length of the frequent itemsets were very long and their support were very small. Example: the number of candidate 2- itemsets will be more than 10^7 if the number of frequent 1-item sets are 10^4 in algorithm apriori. Moreover, In apriori it would generate $2^{100} \approx 10^{30}$ candidate itemsets to find a frequent itesets with the length of 100,also, it needs to scan the database repeatedly for 100 times if the length of the frequent itemsets is 100. From the above it can be concluded that the key problem of the apriori is it take too much time to mining the frequent itemsets. The time mainly costs in two areas, one is the time for scanning the large database repeatedly the other is for generating frequent itemsets with JOIN.

There are a lot of improved algorithm for apriori such as AprioriTID, Apriori Hybri, Multiple oins, Reorder and Direct etc. The main idea of all these algorithm is according the theory that the subset of a frequent itemset is a frequent itemset and the superset of a infrequent itemset is a infrequent itemset. They scan the database repeatdly to mining the association rules. There is another feature for algorithm AprioriTID, the support of the candidate frequent itemsets are calculated only at the first time it scanned the database D and also generated candidate transaction database D' which only includes the candidate frequent itemsets. Then the latter mining are based on the database D', It reduce the time of I/O operation because D' is smaller than D, so, it enhance the efficiency of the algorithm. For algorithm Apriori Hybri, it is the combination of algorithm Apriori and AprioriTID. When the candidate transaction database D' couldn't be contained in the RAM, it mines with algorithm Apriori otherwise with

algorithm AprioriTID. The most important step in mining association is generation frequent itemsets. In algorithm apriori the most time is consumed by scanning the database repeatedly. It would reduce the running time of the algorithm by reducing the times it scans the database far and away. In this paper a method of mining frequent itemsets by evaluating their probability of supports based on association analyzing were mentioned. First , it gained the probability of every 1-itemset by scanning the database, the 1-itemset with the more larger support than the probability the user sets would be frequent 1-itemsets. Second, it evaluates the probability of every 2-itemset, every 3-item set , every k- itemset from the frequent 1-itemsets. Third, it gains all the candidate frequent item sets. Fourth, it scans the database for verifying the support of the candidate frequent itemsets, Last the frequent itemsets are mined and association rules also do. In the method it reduces a lot of times of scanning database and shortened the calculate time of the algorithm.

2. TRADITIONAL APRIORI

Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules.The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties . It is an iterative approach where k-itemsets are used to find out (k+1) itemsets .To improve the efficiency an important property Apriori property is used to reduce the search space .

ALGORITHM

1) L1 = {large 1-itemsets};
2) **for** (k = 2; Lk-1 ≠ Ø; k++) **do begin**
3) Ck = apriori-gen(Lk-1); // New candidates
4) **forall** transactions t ∈ D **do begin**
5) Ct = subset(Ck, t); Candidates contained in t
6) **forall** candidates c ∈ Ct **do**
7) c.count++;
8) **end**
9) Lk = {c ∈ Ck | c.count ≥ minsup}
10) **end**
11) Answer = ∪k Lk;

Example: In our example we consider a transactional database and apply the apriori algorithm for generating frequent itemsets. There are nine transactions in this database that is D=9. Here we assume the minimum support count to be 2.

Transactional Database

TID	List of item ID'S
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Step1:

Scan D for count

C1

Itemset	Supcount
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate sup count with min sup count →

L1

Itemset	Supcount
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Step2:

Generate c2 candidates from L1

C2 Itemset
{I1,I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I2,I3}
{I2,I4}
{I2,I5}
{I3,I4}
{I3,I5}
{I4,I5}

scan D for count →

C2 Itemset	Supcount
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

Compare candidate sup count with minimum sup Results in L2

Itemset	Supcount
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

Step3:

Generate c3 candidates from L2

C3 Itemset
{I1,I2,I3}
{I1,I2,I5}

→

C3 Itemset	supcount
{I1,I2,I3}	2
{I1,I2,I5}	2

→

L3 Itemset	supcount
{I1,I2,I3}	2
{I1,I2,I5}	2

3 THE IMPROVED ALGORITHM FOR APRIORI

3.1 The Main Idea of The Improved Algorithm

Let P_1, $P_2 \ldots P_n$ are the independent probability of every item A_1, $A_2 \ldots A_n$,the probability for any two item A_k, $A_m (P_k < P_m)$ both appeared in one transaction is P_{km}. If A_k and A_m are total non-correlation, from definition 3 it can be concluded that $P_{km} = P_k * P_m$, if A_k and A_m are total correlation, then P_{km} is the minimum of the P_k and P_m that is P_k, so , $P_k * P_m \le P_{km} \le P_k$ Now the problem is: Given P_k and P_m, also $P_k * P_m \le P_{km} \le P_k$, Please evaluate P_{km}. The problem couldn't be solved with the conditions in mathematics. But in fact, there is a lot of information without accurate mathematic formula which be omitted. In this paper it offered a method by association analysis to confirm the formula. Let parameter a be the probability which A_k and A_m are total correlation, and parameter b for total non-correlation. a+b=1, 0 < a, b < 1, then P_{km} can be defined as formula (2) below:

$$P_{km} = a * P_k + b * P_k * P_m \qquad (2)$$

There are a lot methods to confirm the value of parameter a and b, in this paper it provide a way to define "a" and " b"by association analysis.

3.2 Confirm The Parameter "A", "B" By Association Analysis

There are a series of criterion about environment. The most ingredients of the pollution can be confirmed based on the source. So we can consider the criterion as a referenced list and the list needed to find the correlation as a comparison list. Then we'll get the correlation coefficient which is the parameter "a"in our formula (2), and b=1-a. The details as below: Let $S=\{S_1, S_2, \ldots S_m\}$ be the value list of item A_m, S_1, S_2, $\ldots S_m$ are sample extracted from the DB and $X=\{X_1, X_2, \ldots X_m\}$ be the value list for item A_k , $X_1, X_2, \ldots X_m$ are sample extracted from the DB.

$$a_{km} = \frac{\min_i \min_k \Delta_i(k) + \rho \max_i \max_k \Delta_i(k)}{\Delta_i(k) + \rho \max_i \max_k \Delta_i(k)} \qquad (3)$$

PFA[1], PFA[2], …PFA[n] be the P_1, P_2,…, P_n .which refer to the probability of each itemset A_1, A_2, …, A_n .

The process for calculating the probability of A_i appearing. (a) If it is not the end of database ,then read and get the recorder; (b)If there is an item A_i in the recorder then PFA[i] = PFA[i]+1; (c)Repeat the above procedure until the end of the database then PFA[i]=PFA[i]/(number of records in the database).

Repeat step (a)(b)(c)to calculate the probability of itemset $A_1, A_2, \ldots, A_n$ appearing.

2.Set a minimum value V_1 for the probability of A_i appearing, if the probability of A_i appearing PFA[i] is larger than V_1 then

itemset A_i is a frequent 1-itemset. so, you get some frequent 1-itemset, let "m" be the number, and PFA[1],PFA[j],…PFA[m] be the probability of 1-itemset appearing respectively.
3 Due to the probability of 1-itemset appearing PFA[1], PFA[j], …PFA[m] ,base on the formula (2),then the probability of any two itemset appeared in one recorder can be evaluated. Set a minimum value V_2 for the probability of A_i and A_j appeared synchronously in one record, if the probability is larger than V_2 then itemset A_iA_j is a candidate frequent 2-itemset.otherwise set the value of the probability is zero to predigest the later calculation. Let the element of array PUA_2 [i] record the value of candidate frequent 2-itemsets.

Let V_2 be the minimum probability for candidate frequent 2-itemset, and V_3 for candidate frequent 3-itemset, V_{k-1} for candidate frequent (k-1)-itemset
Set minimum probability V_{k-1} :

V_{k-1}=a*min($PFA_{k-1}[1],PFA_{k-1}[2],…PFA_{k-1}[m]$)+b*min($PFA_{k-1}[1],PFA_{k-1}[2],…PFA_{k-1}[m]$)*
max($PFA_{k-1}[1],PFA_{k-1}[2],…PFA_{k-1}[m]$)

4. Recur the above step 1.2.3., from k=2 to n to calculate the probability of k-itemsets $A_1,A_2,…$,A_k appearing in one recorder;
5.Scan the database another time to calculate the support of the candidate frequent itemsets which is the result of step 4.

(a)Create a new array DMA[m] with each element's original value is zero.(m =number of candidate frequent itemsets);

(b)Read and get the recorder of the database until the end of the database.

(c)If there are itemsets $A_i,A_j, … ,A_k$ in any recorder synchronously and $A_i≠0,A_j≠0…A_k≠0$; then the support for $A_iA_j … A_k$ DMA[k]= DMA[k]+1.

recur the above step(b)(c)to calculate the actual support of every candidate frequent itemsets until the end of the database.
6. Find out the frequent itemsets from the candidate frequent itemsets. If DMA[k] is larger than the minimum support which the user set, then output the frequent itemsets.
Step 5.,6. is used to confirm the probability and support of the candidate frequent itemsets which come out by the method of probability evaluation whether satisfy the request of the user.
7. Output the association rule from the result of the step 6.

3.4 Explanation and simulation of the algorithm

The transaction database of some sub-office in All Electronics is choosed to compare the efficiency of our algorithm and Apriori. The detailed data is presented in fig1(A). In the algorithm the paper providing the process of finding frequent itemsets include 3 steps:

Firstly it scans the database to come out the probability of frequent 1-itemsets;

Then evaluates the probability of candidate frequent 2-itemsets,3-itemsets …m-itemsets, based on the probability of frequent 1-itemsets;

Last it scans the database another time to confirm the frequent itemsets from the candidate frequent itemsets.
There is a database with 9 transaction and 5 itemsets. The parameter "a","b"is 5/9 and 4/9 respectively in the simulation of the algorithm.

T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I5
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

(1A)

Itemsets	Probability
I1	6/9
I2	7/9
I3	5/9
I4	2/9
I5	3/9

(1B)

Set a=5/9,b=4/9;
Threshold V_2=(5*(2/9)+4*(2/9)*(7/9))/9
=146/729

From frequent C1(fig 1B)to evaluate the probability of candidate frequent2-itemstes(fig 1C)

Itemsets	Probability
{I1,I2}	438/729
{I1,I3}	345/729
{I1,I4}	138/729
{I1,I5}	207/729
{I2,I3}	365/729
{I2,I4}	146/729
{I2,I5}	219/729
{I3,I4}	130/729
{I3,I5}	195/729
{I4,I5}	114/729

Get the candidate frequent 2-itemsets below(1D):

2-itemsets	Probability
{I1,I2}	438/729
{I1,I3}	345/729
{I1,I5}	207/729
{I2,I3}	365/729
{I2,I4}	146/729
{I2,I5}	219/729
{I3,I5}	195/729

(1D)

Set a=5/9,b=4/9;
Threshold V_3=(5*(146/729)+4*(146/729)*(7/9))/9
=73*146/59049

From frequent C2 to evaluatethe probability of candidatefrequent 3-itemstes(fig 1E)

Itemset	Probabilty
{I1,I2,I3}	69*345/59049
{I1,I2,I4}	73*138/59049
{I1,I2,I5}	73*207/59049
{I1,I3,I4}	69*130/59049
{I1,I3,I5}	69*195/59049
{I1,I4,I5}	69*114/59049
{I4,I2,I3}	73*130/59049
{I3,I4,I5}	65*114/59049
{I2,I4,I5}	73*114/59049
{I2,I3,I5}	73*195/59049

(1E)

Get the candidate frequent 3-itemsets below(1F):

3-itemsets	Probability
{I1,I2,I3}	69*345/59049
{I1,I2,I5}	73*138/59049
{I2,I3,I5}	73*195/59049

Fig1 The process of producing candidate itemsets

3.5 Comparison for the Algorithms

It was compared between our algorithm and apriori in this section ,the number of the candidate frequent itemsets and times of scanning the database. Which also were the linchpin of the efficiency in a algorithm. The algorithm in this paper ahead apriori in reducing the times of scanning the database. It would scan the database k times to find out frequent k-itemsets in apriori while only 2 times in our algorithm by putting forward a concept of candidate frequent itemset.

In the table1 below, it listed the frequent 1-itemsets, 2-itemsets, 3-itemsets for both of the two algorithm respectively.

Candidate frequent itemsets	Algorithm Apriori
1-Itemsets	(I1,I2,I3,I4,I5}
2-Itemsets	{I1,I2},{I1,I3},{I1,I3},{I1,I5},{I2,I3},{I2,I4},{I2,I5}

Table 1 The Comparison of the frequent itemsets

4. CONCLUSION

Based on the previous studies on algorithm apriori, we proposed a method for mining association rules in large databases with association analysis and probability evaluating. The method developed in this paper explores efficient mining of association rules by probability evaluating . First it scans the database to filter frequent 1-itemsets and gets their probability respectively. Then it gets the candidate frequent 2-itemset,3-itemsets up to n-itemsets by evaluating their probability on formula (2) in the paper and the result of the first step. Last it scans the database for another time to refine the candidate frequent itemsets to the frequent itemsets. Data mining in association rule is an important research topic and apriori is the core algorithm in mining association rule. We propose a method that enhances the efficient of algorithm by evaluating the probability of candidate frequent itemsets. It shortens the runtime of algorithm by reducing the times of scanning database. A formula is provided in this paper. We also present an efficient method for confirming the parameter "a","b" in formula (2) by association analysis. If the parameter "a","b "are unseemliness, it will omit some association rules which users are interested in. The solution for this problem is multi enactment for parameter "a" and "b",then choose the best. The method of grey relational analysis is widely used in the association analysis of environment databases. There are a lot of other methods to confirm the parameter "a","b". In this paper we implemented the algorithm and the performed experiments show significant benefits for different number of the itemsets and tuples in the database.

REFERENCES

[1] R. Agrawal, R. Srikant. "Fast algorithms for mining association rules in large databases". *Proc. of 20th Int'l conf. on VLDB*: 487-499, 1994.
[2] J. Han, J. Pei, Y. Yin. "Mining Frequent Patterns without Candidate Generation". *Proc. of ACM-SIGMOD*, 2000.
[3] C. Győrodi, R. Győrodi. "Mining Association Rules in L arge Databases". *Proc. of Oradea EMES'02*: 45-50, Oradea, Romania, 2002.
[4]C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98, 594-605, New York, NY.
[5]R. Srikantand R. Agrawal. Mining generalized association rules. VLDB'95, 407-419, Zurich, Switzerland, Sept. 1995.
[6]R. Srikantand R. Agrawal. Mining quantitative association rules in large relational tables. SIGMOD'96, 1-12, Montreal, Canada.
[7]H. Toivonen. Sampling large databases for association rules. VLDB'96, 134-145, Bombay, India, Sept. 1996.
[8]Ashrafi, M., Taniar, D., Smith, K., Redundant Association Rules Reduction Techniques, Lecture Notes in Computer Science, Volume 3809, 2005, pp. 254- 263
[9] Baralis, E., Psaila, G., Designing templates for mining association rules. Journal of Intelligent Information Systems, 9(1):7-32, July 1997.
[10] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. 1997. Dynamic itemset
ounting and implication rules for market basket data. In SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, 255-264.
[11] Jiawei Han , Micheline Kamber , Data mining concepts and techniques. C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98, 594-605, New York, NY.
[12]R. Srikantand R. Agrawal. Mining generalized association rules. VLDB'95, 407-419, Zurich, Switzerland, Sept. 1995.
[13]R. Srikantand R. Agrawal. Mining quantitative association rules in large relational tables. SIGMOD'96, 1-12, Montreal, Canada.
[14]R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. KDD'97, 67-73, Newport Beach, California.
[15]H. Toivonen. Sampling large databases for association rules. VLDB'96, 134-145, Bombay, India, Sept. 1996.
[16]D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining.
SIGMOD'98, 1-12, Seattle, Washington.
[17]K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. KDD'97, 96- 103, Newport Beach, CA, Aug. 1997.
[18]M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules. Data Mining and Knowledge Discovery, 1:343-374, 1997.
[19]M. Zaki. Generating Non-Redundant Association Rules. KDD'00. Boston, MA. Aug. 2000.
[20]O. R. Zaiane, J. Han, and H. Zhu. Mining Recurrent Items in Multimedia withProgressive Resolution Refinement. ICDE'00, 461-470, San Diego, CA, Feb. 2000

[21] Brin, S., Motwani, R. and Silverstein, C., "Beyond Market Baskets: Generalizing Association Rules to Correlations," Proc. ACM SIGMOD Conf.,
pp. 265-276, May 1997.
[22] Cheung, D., Han, J., Ng, V., Fu, A. and Fu, Y. (1996), A fast d stributed algorithm for mining association rules, in `Proc. of 1996 Int'l. Conf. on Parallel and Distributed Information Systems', Miami Beach, Florida, pp. 31 - 44.
[23] Cheung, D., Xiao, Y., Effect of data skewness in parallel mining of association rules, Lecture Notes in Computer Science, Volume 1394, Aug 1998, Pages 48 - 60
[24] Chuang, K., Chen, M., Yang, W., Progressive Sampling for Association Rules Based on Sampling Error Estimation, Lecture Notes in Computer Science, Volume 3518, Jun 2005, Pages 505 - 515
[25] Cristofor, L., Simovici, D., Generating an informative cover for association rules. In Proc. of the IEEE International Conference on Data Mining, 2002.
[26] Das, A., Ng, W.-K., and Woon, Y.-K. 2001. Rapid association rule mining. In Proceedings of the tenth international conference on Information and knowledge management. ACM Press, 474-481.
[27] Tien Dung Do, Siu Cheung Hui, Alvis Fong, Mining Frequent Itemsets with Category- Based Constraints, Lecture Notes in Computer Science, Volume 2843, 2003, pp. 76 - 86
[28] Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2, 2, 14-20.
[16] Hegland, M., Algorithms for Association Rules, Lecture Notes in Computer Science, Volume 2600, Jan 2003, Pages 226 - 234
[17] Hilderman R. J., Hamilton H. J., Knowledge Discovery and Interest Measures, Kluwer Academic, Boston, 2002.
[18] Jaroszewicz, S., Simovici, D., Pruning Redundant Association Rules Using Maximum Entropy Principle, Lecture Notes in Computer Science, Volume 2336, Jan 2002, pp 135-142
[19] Li, Y., Gopalan, R., Effective Sampling for Mining Association Rules, Lecture Notes in Computer Science, Volume 3339, Jan 2004, Pages 391 - 401
[20] Liu, B. Hsu, W., Ma, Y., "Mining Association Rules with Multiple Minimum Supports," Proc. Knowledge Discovery and Data Mining Conf., pp. 337-341, Aug. 1999. Manning, A., Keane, J., Data Allocation Algorithm for Parallel Association Rule Discovery, Lecture Notes in Computer Science, Volume 2035, Page 413-420.
[22] Omiecinski, E. (2003), Alternative Interest Measures for Mining Associations in Databases, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 1, pp. 57-69.
[23] Parthasarathy, S., Zaki, M. J., Ogihara, M., Parallel data mining for association rules on shared-memory systems. Knowledge and Information Systems: An International Journal, 3(1):1–29, February 2001.
[24] Parthasarathy, S., Efficient Progressive Sampling for Association Rules. ICDM 2002: 354-361.
[25] Tang, P., Turkia, M., Parallelizing frequent itemset mining with FP-trees. Technical Report titus.compsci.ualr.edu/~ptang/papers/par-fi.pdf, Department of Computer Science, University of Arkansas at Little Rock, 2005.
[26] Ramaswamy, S., Mahajan, S., Silbershatz, A., "On the Discovery of Interesting Patterns in
Association Rules," Proc. Very Large Databases Conf., pp. 368-379, Sept. 1998.
[27] Sarawagi, S., Thomas, S., "Mining Generalized Association Rules and Sequential Patterns Using SQL Queries". In Proc. of KDD Conference, 1998.
[28] Savasere, A., Omiecinski, E., Navathe, S.: Mining for strong negative associations in a large database of customer transactions. In: Proc. of ICDE. (1998) 494–502
[29] Schuster, A. and Wolff, R. (2001), Communication-efficient distributed mining of association rules, in `Proc. of the 2001 ACM SIGMOD Int'l. Conference on Management of Data', Santa Barbara, California, pp. 473-