Data mining using Association rule based on APRIORI algorithm and improved approach with illustration

Pratibha Mandave
Department of Master of Computer Application
STES's Sinhgad Institute of Business Administration & Research, Pune, Maharashtra, India

Megha Mane
Department of Master of Computer Application
STES's Sinhgad Institute of Business Administration & Research ,Pune, Maharashtra, India

Prof. Sharada Patil
Department of Master of Computer Application
STES's Sinhgad Institute of Business Administration & Research ,Pune, Maharashtra, India

Abstract- In this paper we have explain one of the useful and efficient algorithms of Association mining named as APRIORI algorithm. *Association rule of data mining is used in all real life applications of business and industry.* Using this we gets an effective results rather than traditional results. Association rules are the main technique for data mining and APRIORI algorithm is a classical algorithm. Lots of algorithms for mining association rules and their mutation (change/transformation) are proposed on basis of APRIORI algorithm, but traditional algorithms are not efficient. The main intension of this paper is to understand the concept of association rule and how to implement the APRIORI algorithm and improved APRIORI algorithm.

Keywords – Data mining, Association Rule; APRIORI Algorithm; Improved APRIORI Algorithm; Frequent Item set ;association mining.

I. INTRODUCTION

Data mining is an important method to increase efficiency, discover hidden (novel), useful, valid and understandable knowledge from a massive databases. Data mining is the process of analyzing data from different perspective and summarizing the data into useful identical format of information that can be used to predict future trends or performances. The ultimate goal of data mining is to recognize pattern full information and predictions.[1]
Association mining is an important component of data mining. But what is *Association Mining?*

II . ASSOCIATION MINING

***Association Mining*: -** Association mining is one of the most popular ways of data mining uses association rules that are an important class of methods of finding regularities/patterns in data. It is perhaps the most important model invented and extensively studied by databases and data mining community. Association mining has been used in many application domains. One of the best known is the business field where discovering of purchase patterns or association between products is very useful for decision making and effective marketing.
Applications domains are like: finding patterns in biological databases, extraction of knowledge from software engineering metrics, web personalization, text mining, telecommunication networks, market and risk management, inventory control etc. Association rule mining can also play an important role in discovering knowledge.[5]
It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories [6]

Definition of an Association Rule: Association rule of data mining involves picking out the unknown inter-dependence of the data and finding out the rules between those items [3]. Agrawal introduced association rules for

point of sale (POS) systems in supermarkets. A rule is defined as an implication of the form A=>B, where A∩ B≠Ø. The left-hand side of the rule is called as antecedent. The right-hand side of the rule is called as consequent. [3]
For example we may find that "95 percent of customers who bought pen (A) also bought paper (B)" A rule may contain more than one item in antecedent and consequent of rule. Every rule must satisfy two users specified constrains: one is measure of statistical significance called support and other is measure of goodness called confidence. [4]

In simple words we can say like
If A and B then C
If A and not B then C
If A and B and C then D etc.

What is support and Confidence?
Support :- the minimum percentage of transactions in the DB containing A and B i.e. A U B
Confidence:- the minimum percentage of those transactions containing A that also contain B .(A intersection B)
Ex. Suppose the DB contains 1 million transactions andt that 10'000 of those transactions contain both A and B.
We can then say that the support of the association if A then B is:
S= 10'000/1'000'000 = 1%.
Likewise, if 50'000 of the transactions contain A and 10'000 out of those 50'000 also contain B then the association rule if A then B has a confidence 10'000/50'000 = 20%.
Confidence is just the conditional probability of B given A.**[8]**
How data mining is associated with association rules through example [7]
Given a data set, find the items in the data that are associated with each other

- Association is measured as frequency of occurrence in the same context

Transaction ID	**items**
1	{ pen, paper }
2	{ pen, erasers, sharpener, ink }
3	{ pen, erasers, sharpener, pencil}
4	{ pen, paper, erasers, sharpener }
5	{pen, paper, erasers, pencil }

The association rule in above transactions is as
{Paper, erasers} -> {sharpener, pencil}

Finding associations in items can reveal interesting relations between items Association mining does identify items that are connected to each other.

Frequent Item set

- **Item set- It is a** collection of one or more items like
  e.g., Pen=5, paper=3, erasers=4, sharpener=3, ink=1, pencil=2

- **Support [count (s)]:** support count is the item set which occurred frequently in transactions like e.g. ({ pen, paper})=3 ,({ pen, erasers, sharpener })=3
  e.g.:- Supp ({ pen, erasers, sharpener })/ N means 3/5=0.6 since it occurs in 60% of all transactions (3 out of 5 transactions).where N = Total no. of transactions
- **Confidence** (c): Measures how often the items are occurred in contexts containing in transaction.
  For example, the rule{pen, paper}=>{pencil}has a confidence of 0.6/0.4= 1.5 in the database.

III. INTRODUCTION OF AN APRIORI ALGORITHM

This is also referred as a fast algorithm in mining frequent item sets and associations. The main objective of APRIORI algorithm is to uncover hidden information that is the major goal of data mining. It was first introduced in

1993 ASSOCIATION rules mining is a very popular data mining technique and it finds relationships among the different entities of records (for example, transaction records). Since the introduction of frequent item sets in 1993 by Agrawal et al. [1], it has received a great deal of attention in the field of knowledge discovery and data mining.[2]
Association mining using APRIORI algorithm is fundamentally based on the principle of
1. Frequent item set generation: Generate all item sets with support>= min-support
2. Rule generation: Generate high confidence rules from each frequent item set

Classical APRIORI Algorithm:-

Join Step: - Ck is generated by joining Lk-1 with itself.
Prune step: - Any (k-1) –itemset that is not frequent cannot be a subset of a frequent k-1 itemset
Where,
Ck: candidate itemset of size k
Lk:- frequent itemset of size k
L1={frequent items};
For (k=1; Lk! =Ø;k++)
do begin
Ck-1=candidates generated from Lk;
For each **transaction t** in database do
Increment the count of all candidates in Ck-1 that are contained in t
Lk-1 =candidates in Ck-1 with min_support
end
Return UkLk [3]
Example:-Assume that we are having 5 transactions in a database i.e. D=5.

Transaction ID	Item ID's
1	I1,I2,I3,I5
2	I3,I4,I5
3	I3,I5
4	I1,I3,I4
5	I2,I4

Step 1-Find frequent items in above transactions to count support of each item.

TID	Item ID's	Support count
1	I1	2
2	I2	2
3	I3	4
4	I4	3
5	I5	3

Step 2-Join the table with itself.
In step 2 we are building item set of two items. And these item set can be counted with the original table.
Note – In step 1 we are assuming minimum support count=2 and maximum support count =4.So, all items are greater than 2 .As a result no pruning is required.

TID	Item ID's	Support count
1	I1,I2	1

2	I1,I3	2
3	I1,I4	1
4	I1,I5	1
5	I2,I3	1
6	I2,I4	1
7	I2,I5	1
8	I3,I4	2
9	I3,I5	3
10	I4,I5	1

Step 3- In this step we are pruning the above table on the basis of association rule which pursue equal or more than 2 support counts. But above table contains many item set having less than 2 support counts. So here pruning is required. After pruning the resultant table will be as

TID	Item ID's	Support count
1	I1,I3	2
2	I3,I4	2
3	I3,I5	3

Step 4- Join the table with itself

TID	Item ID's	Support count
1	I1,I3,I4	1
2	I3,I4,I5	1

If we compare with support of the above table's item set with minimum support (i.e 2) then none of the transaction sets are qualified.

The result of applying APRIORI algorithm on above item sets with minimum support=2 .So, We get 3 frequent item sets as {I1, I3}, {I3, I4} and {I3,I5}.

IV. IMPROVED APRIORI ALGORITHM

APRIORI algorithm may generate ample number of candidate generations. Every time algorithm needs to judge whether these candidate generations are frequent item sets or not. This results into high frequency in querying, so huge amount of resources are spent may be in terms of time or space.
In order to reduce this drawback, Improved APRIORI Algorithm is one of the best solution in reducing querying frequencies and storage resources. The designed IMPROVED APRIORI ALGORITHM that mines frequent item sets without new candidate generation.
For example, in this algorithm we compute the frequency of frequent k-item sets when k-item sets are generated from (k-1)-item sets. If k is greater than the size of transaction T, there is no need to scan Transaction T which is generated by (k-1)-item sets according to the nature of Apriori algorithm, and we can remove it.[9]

Improved APRIORI Algorithm

Input: transaction database D; min-sup.
Output: the set of Frequent L in the database D
(1)min-sup-count=min-sup*|D|
(2)L1-candidates=find all one itemsets(D) //Scan D and produce L1-candidates
(3)L1={<X1,TID-set(X1)>∈L1-candidates |sup-count≥min-sup-count}
(4)for (k=2; Lk-1≠ϕ;k++) do {
(5) {for each k-itemset (xi,TID-set(xi)∈Lk-1 do

(6) for each k-itemset (xj,TID-set(xj)∈Lk-1 do
(7) if (xi[1]=xj[1])∧(x i[2]=xj[2])∧…∧(xi[k-2]=xj[k-2] then
(8) {Lk-candidates.Xk= Xi* Xj;
(9) Lk-candidates.TID-set(Xk) = TID-set(Xi)∩TID-set(Xj)
(10) }}
(11) for each k-itemset<Xk,TID(Xk)>∈Lk-candidates do
(12) sup-count=|TID-set|
(13) Lk={<Xk,TID-set(Xk)>∈Lk-candidates|sup-count≥min-sup-count}
(14) }
(15) set-count=Lk.itemcount //updata set-count
(16) return L=∪kLk; **[8]**

Experimental result:-Assume that we are having 10 transactions in a database i.e. D=10.

D1

TID	Items
T1	I1,I2,I4
T2	I2,I5
T3	I2,I3
T4	I1,I2,I5
T5	I1,I2
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I4
T9	I1 ,I2,I3
T10	I1,I4

Step 1 Scan D for count of each candidate
In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, C1. The algorithm scans all of the transactions in order to count the number of occurrences of each item
C1

Item Set	Support Count
I1	7
I2	8
I3	4
I4	3
I5	2

Step 2 Compare candidate support count with minsup.
Suppose that the minimum transaction support count required is 2.The set of frequent 1-itemsets, L1, can then be determined. It consists of the candidate1-itemsets satisfying minimum support count i.e 2
L1

Item Set	Support Count
I1	7
I2	8
I3	4
I4	3
I5	2

Step 3: Generate C2 candidates from L1 and scan D for count of each candidate.
To discover the set of frequent 2-itemsets, L2, the algorithm generates a candidate set of 2-itemsets, C2.And then the transactions in D are scanned and the support count of each candidate itemset in C2 is accumulated, as shown in the table of C2.

C2

Item Set	Support Count
I1,I2	5
I1,I3	3
I1,I4	3
I1,I5	1
I2,I3	4
I2,I4	2
I2,I5	1
I3,I4	1
I3,I5	0
I4,I5	0

Step 4: Compare candidate support count with minsup.
The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 having minimum support. Then D2 was determined from L2.

L2

Item Set	Support Count
I1,I2	5
I1,I3	3
I1,I4	3
I2,I3	4
I2,I4	2

D2

TID	Items
T1	I1,I2,I4
T4	I1,I2,I5
T8	I1,I2,I3,I4
T9	I1 ,I2,I3

Step 5: Generate C3 candidates from L2 and scan D2 for count of each candidate.
First let C3=L2 ∞ L2= {{I1, I2, I3}, {I1, I2, I4}, {I2, I3, I4}}. Based on the APRIORI property that all subsets of a frequent item set must also be frequent, it can determine that the four-letter candidates cannot possibly be frequent, therefore remove them from C3, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of D2 to determine L3.

C3

Items	Support Count
I1,I2,I4	2
I1,I2,I5	0
I1,I2,I3,I4	1
I1 ,I2,I3	2

C3

Items	Support Count
I1,I2,I4	2
I1 ,I2,I3	2

Step 6: Compare candidate support count with minsup.
The transactions in D2 are scanned in order to determine L3, consisting of those candidate 3-itemsets in C3 having minimum support

L3

Items	Support Count
I1,I2,I4	2
I1 ,I2,I3	2

Step 7:The algorithm uses L3 ∞ L3 to generate a candidate set of 4-itemsets, C4. Although the join results in {{I1, I2, I3, I4}}, this item set is pruned since its subset {{I1, I2, I5}}is not frequent. Thus, C4= Ø,

and the algorithm terminates, having found all of the frequent item sets. The efficiency of algorithm is also being by finding execution time for different dataset with different support count.[9]

D3

TID	Items
T8	I1,I2,I3,I4

C4

Items	Support count
I1,I2,I3,I4	0

V. CONCLUSION

Association rule is one of efficient technique of data mining for finding out frequent item set in transaction database D. Using Classical APRIORI Algorithm the efficiency of execution turns very low whereas improvement in the algorithm helps us for reducing querying frequencies and storage spaces. Also it greatly helps in increasing the efficiency and reduces I/O load.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A.N. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 207-216, May 1993.

[2] A Transaction Mapping Algorithm for Frequent Itemsets Mining Mingjun Song and Sanguthevar Rajasekaran, Senior Member, IEEE

[3] Divya Bansal*1 Lekha Bhambhu ,"Execution of APRIORI Algorithm of Data Mining Directed Towards Tumultuous Crimes Concerning Women" ² *[1]M.Tech Scholar ² Assistant Professor Department of Computer Science Department of Computer Science. J.C.D college of Engineering and Technology J.C.D College of Engineering and Technology G.J.U. University of Science & Technology, India G.J.U. University of Science & Technology, India.*

[4] Association Rule Mining.PDF (Winter School on "Data Mining Techniques and Tools for Knowledge Discovery in Agricultural Datasets)

[5] Sotiris Kotsiantis, Dimitris Kanellopoulos ," Association Rules Mining: A Recent Overview "GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82

[6] Qiankun Zhao and Sourav S. Bhowmick," Association Rule Mining: A Survey",Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003116 , 2003

[7] Christof Monz ,Queen Mary University of London, Machine Learning for Data Mining, Week 10: Association Analysis

[8] Xiang Fang, "An Improved Apriori Algorithm on the Frequent Item set", International Conference on Education Technology and Information System (ICETIS 2013)

[9] SurajP. Patil1, U. M. Patil2 and Sonali Borse ,"The novel approach for improving apriori algorithm for mining association rule", Proceedings of "National Conference on Emerging Trends in Computer Technology (NCETCT-2012)"Held at R.C.Patel Institute of Technology, Shirpur, Dist. Dhule, Maharashtra, India. April 21, 2012