ISSN No. 0976-5697

Volume 4, No. 2, Jan-Feb 2013

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Association Rule mining using Apriori Algorithm: A Review

Manisha Bhargava*,Arvind Selwal
Department of Computer Science & Engineering,
Ambala College of Engineering & Applied Research Ambala City, India
Engg.manisha@yahoo.co.in, arvind_ace11@rediffmail.com

Abstract: Data mining or knowledge discovery is the process of discovering patterns in large data sets. In data mining each algorithm has a different objective and to obtain meaningful and previously unknown patterns from large dataset is an emerging and challenging problem. Association rule mining is a technique for discovering unsuspected data dependencies and is one of the best known data mining techniques. The basic Idea to identify from a given database, consisting of item sets (e.g. shopping baskets), whether the occurrence of specific items, implies also the occurrence of other items with a relatively high probability. Apriori algorithm is one of the popular approaches which are used to extract association rules from data sets. One of the most popular data mining approaches is to find frequent item sets from a transaction dataset and derive association rules. In this paper, we describe the association rules which are descriptive data mining technique. This paper also addresses Apriori Algorithm and two other algorithms Record filter and Intersection Approach based on Apriori.

Keywords: Data Mining, Association rules, Apriori Algorithm, Record filter approach, Intersection Approach.

I. INTRODUCTION

Data Mining is the extraction of hidden predictive information from large databases. It is sometimes called data or KDD (knowledge discovery in databases) [1]. Data miming is an essential step in the process of knowledge discovery in databases in which intelligent methods are applied in order to extract patterns. Other steps in the knowledge discovery process include pre-mining tasks such as data cleaning and data integration as well as post-mining tasks such as pattern evaluation and knowledge presentation [2].Data mining software is an important tool for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. These tools predict future trends and can answer the questions that traditionally were too time consuming to resolve [2]. In general, data mining tasks can be descriptive and predictive. Descriptive mining is the process of drawing the essential characteristics of the data in the database. Clustering, Association and Sequential mining are some of the descriptive mining techniques. Predictive mining is the process of inferring patterns form data to make predictions. The predictive mining techniques involve tasks like Regression, classification and Deviation detection [3].One of the popular descriptive data mining technique is Association Rule Mining, owing to its extensive use in marketing and retail communities in addition to many other diverse fields. Apriori algorithm is the algorithm to extract association rules from dataset. It is initially used for Market Basket Analysis to find how items purchased by customers are related [4].

II. ASSOCIATION RULES

Association rule mining is used to find association relationships among large data sets [2]. Mining frequent patterns is an important aspect in association rule mining**.** It is useful for discovering relationships among items from large databases. A standard association rule is a rule of the form X→ Y which says that if X is true of an instance in a database, so is Y true of the same instance, with a certain level of significance as measured by two indicators, support and confidence. The goal of standard association rule mining is to output all rules whose support and confidence are respectively above some given support and coverage thresholds [3].

a. ***Minimum Support Threshold:*** The support [4] of an association pattern is the percentage of task-relevant data transaction for which the pattern is true. An item set satisfies minimum support if the occurrence frequency of the item set (A set of items) is greater than or equal to minimum support. If an item set satisfies minimum support, then it is a frequent item set.

$$\text{Supp}\,(A \Longrightarrow B) = \frac{\text{support containing both A and B}}{\text{total of tuples}}$$

b. ***Minimum Confidence Threshold:*** Confidence [4] is defined as the measure of certainty or trust worthiness associated with each discovered pattern.

$$\text{Conf}\,(A \Longrightarrow B) = \frac{\text{support containing both A and B}}{\text{tuples containing A}}$$

An example [4] of an association rule is:

Contains (T,"baby food") → Contains (T, "diapers") [Support= 4%, Confidence=40%]

The interpretation of such rule is as follows:

a) 40% of transactions that contains baby food also contain diapers;
b) 4% of all transactions contain both of these items.

ARM mainly includes two steps: first, find all frequent patterns; second, generate association rules through frequent patterns [5].

III. APRIORI ALGORITHM

For mining frequent item sets and strong association rules R. Agrawal [6] and R Srikant introduced Apriori [7] algorithm in 1994. Apriori algorithm is, the most classical and important algorithm for mining frequent item sets. It Assume all data are Categorical. It is used for Market Basket

Analysis [4] to find how items purchased by customers are related. Apriori is used to find all frequent item sets in a given database. The key idea of Apriori algorithm is to make multiple passes over the database [5].

It employs an iterative approach known as a breadth-first search (level-wise search) through the search space, where k-item sets are used to explore (k+1) item sets [5]. The algorithm terminates when no further successful extensions are found. But it has to generate a large amount of candidate item sets and scans the data as many times as the length of the longest frequent item sets. The advantage of the algorithm is that before reading the database at every level, it prunes many of the sets which are unlikely to be frequent sets by using the Apriori property, which states that all nonempty subsets of frequent sets must also be frequent. This property belongs to a special category of properties called anti-monotone in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well [8].

Using the downward closure property and the Apriori Property, this algorithm works as follows. The first pass of the algorithm counts the number of single item occurrences to determine the L_1 or single member frequent item sets. Each subsequent pass K consists of two phases. First, the frequent item sets L_{k-1} found in the (k-1) Th pass are used to generate the candidate item sets C_k, using the Apriori candidate generation algorithm. Next, the database is scanned and the support of the candidates in C_k is determined to ensure that C_k item sets are frequent item sets [9].

A. Pseudo-code [8] for Apriori:

Initialize: k=1, C_1=all the 1-item sets.
Read the data base to count the support of C_1 to determine L_1.
L_1: = {frequent 1-item sets};
K: =2 //k represent the pass number//
While ($L_{k-1} \neq \emptyset$) do
Begin
C_k : = gen_candidate_item sets with the given L_{k-1}
Prune (C_k)
For all transactions t □ T do
Increment the count of all candidates in C_k that are contained in t;
L_K: = All candidates in C_k with minimum support;
K: = k + 1;
End
Answer: = $\cup_k L_k$;

B. Example of Apriori algorithm:

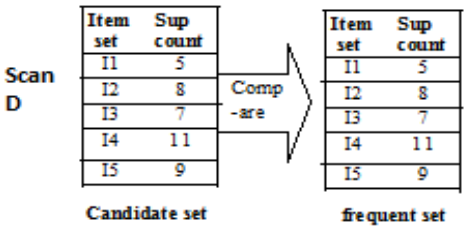
To understand the functioning of classical Apriori algorithm, we consider a database of 15 transactions containing an item set I = {I1, I2, I3, I4, I5} of five items. Before starting the Apriori we assume absolute support count of 3.

Table 1: Database (D)

TID	ITEMS
T1	I1,I3,I5
T2	I1,I4
T3	I4,I5
T4	I2,I3,I4
T5	I1,I2,I3
T6	I2,I4,I5
T7	I2,I5
T8	I2,I3,I4,I5
T9	I4
T10	I2,I3,I4,I5
T11	I3,I4
T12	I1
T13	I2,I4,I5
T14	I4,I5
T15	I1,I2,I3,I4,I5

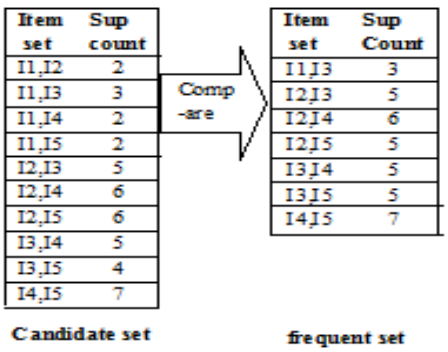
In the first step of Apriori algorithm we take the candidate set of one item and scan the database to count the support of each member of candidate set. After scanning we compare the support count with minimum support count (i.e. 3) and write only those items in the item set which has support count greater than or equal to 3.This process shows in table 2.

Table2



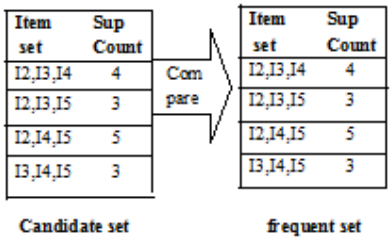
After determining the frequent set of 1 item, we generate the candidate set of 2 items by merging the frequent set of 1 item. After that we again scan the database D to count the support of each element of candidate set and generate the frequent set of 2 items by comparing support count with minimum support count (i.e. 3) and write only those items in the item set which has support count greater than or equal to 3.This process shows in table 3.

Table3



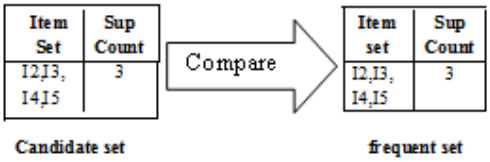
Further we generate a candidate set of 3 items by using frequent 2 item sets and pruning technique. After that we again scan all the transactions in database D to count the support of each element of candidate set in order to get the frequent set by comparing them with the minimum support count (i.e. 3) and write only those items in the item set which has support count greater than or equal to 3.This process shows in table4.

Table 4



In the next step we generate candidate set of 4 items by using frequent 3 item sets and pruning technique and determine the support of candidate set by scanning all the transactions available in the database in order to get frequent set of 4 items. This process shows in table 5.

Table 5



In this way Apriori discover all frequent item set by scanning all the transactions in each repetitive scan and this is the final result.

IV. RECORD FILTER APPROACH BASED ON APRIORI ALGORITHM

This algorithm is also used for finding frequent pattern mining and it is efficient as compare to the apriori algorithm [6]. We have made some changes in the apriori algorithm and made this finest approach which improves the efficiency of Apriori, memory management and remove the complexity of process. In the classical Apriori algorithm, we check the occurrence of candidate item in each transaction of any length. In record filter approach [8] when we count the support of candidate set of length k, we also check its occurrence in transaction whose length may be greater than, less than or equal to the k. This approach has taken very less time as compared to classical Apriori [8].

A. Pseudo-code [9] for Record Filter Approach:

Initialize: K: = 1, C_1 = all the 1- item sets;
Read the database to count the support of $C_1$to
Determine L_1.
L_1: = {frequent 1- item sets};
K: =2; //k represents the pass number//
While ($L_{k-1} \neq \emptyset$) do
Begin
C_K: = gen_candidate_itemsets with the given L_{k-1}
Prune (C_K)
For all transactions t whose length is greater than or Equal to k □ T do
Increment the count of all candidates in C_K that are Contained in t;
L_K : = All candidates in C_K with minimum support;
K: = k + 1;
End
Answer: =$\cup_k L_k$;

V. INTERSECTION APPROACH BASED ON APRIORI ALGORITHM

The intersection approach is made by changes in Apriori. This gives better result as compare to the Record Filter approach. The intersection algorithm [8] is designed to improve the efficiency, memory management and remove the complexity of apriori. Here we are presenting a different approach in apriori algorithm to count the support of candidate item set.

This approach is more appropriate for vertical data layout, since apriori basically works on horizontal data layout. In this new approach, we use the set theory concept of intersection. In classical apriori algorithm, to count the support of candidate set each record is scanned one by one and check the existence of each candidate, if candidate exists then we increase the support by one. This process takes a lot of time, requires iterative scan of whole database for each candidate set, which is equal to the max length of candidate item set. In modified approach, to calculate the support we count the common transaction that contains in each element's of candidate set, by using the intersect query of SQL. This approach requires very less time as compared to Classical Apriori algorithm [8].

A. Pseudo-code for Intersection Approach:

Initialize: K: = 1, C_1 = all the 1- item sets;
Read the database to count the support of C_1 to
Determine L1.
L_1: = {frequent 1- item sets};
K: =2; //k represents the pass number//
While ($L_{k-1} \neq \emptyset$) do
Begin
C_k: = gen_candidate_itemsets with the given L_{k-1}
Prune (C_k)
For all candidates in C_k do
Count the number of transactions that are common in each item □ C_k
L_k : = All candidates in C_k with minimum support;
K: = k + 1;
End
Answer: = $\cup_k L_k$;

VI. LIMITATIONS OF ASSOCIATION RULE MINING

a. End users of ARM encounter problems as the algorithm do not return result in a reasonable time [4].
b. It only tells the presence and absence of an item in transactional database.
c. It is not efficient in case of large dataset.
d. ARM treats all items in database equally by considering only the presence and absence of an item within the transaction. It does not take into account the significance of item to user or business [4].

e. ARM fails to associate user objective and business value with outcome of ARM analysis.

VII. CONCLUSION

Apriori algorithm is applied on the transactional database. By using measures of apriori algorithm, frequent item sets can be generated from the database. Apriori algorithm is associated with certain limitations of large database scans. Record filter approach gives better result than Apriori. This takes less time as compared from Apriori. Intersection approach takes less time as compared to record filter approach.

VIII. REFERENCES

[1]. Textbook Jiawei Han, "Data Mining: concepts and techniques", Morgan Kaufman, 2000

[2]. J.Han and M. Kamber," Data Mining concepts and techniques "Morgan Kaufmann Publisher, CA, UA 2001

[3]. V.Umarani, Dr.M.Punithavalli, "A study on Effective Mining Of Association Rules From Huge DataBases", IJCSR International Journal of Computer Science and Research, Vol. 1 Issue 1, 2010

[4]. Mamta Dhanda, Sonali Guglani and Gaurav Gupta, "Mining Efficient Association Rules through Apriori Algorithm Using Attributes", IJCST Vol. 2, Issue 3, September 2011 ISSN : 2229 - 4333 (Print) | ISSN : 0976 - 8491 (Online)

[5]. M Suman, T Anuradha, K Gowtham, A Ramakrishna," A Frequent Pattern Mining Algorithm Based On FP-Tree Structure And Apriori Algorithm", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 2, Issue 1, Jan-Feb 2012, pp.114-116

[6]. Agrawal, R., Imielinski, T., and Swami, A., "Mining association rules between sets of items in large databases", In Proceedings of ACM SIGMOD International Conference on Management of Data, 207-216.May 1993.

[7]. Agrawal, R., Srikant, R.," Fast Algorithms formining association rules in large databases", Proceedings of 20th Int. Conference on Very Large Databases, Santiago de Chile pp. 487-489, 1994.

[8]. Goswami D.N, Chaturvedi Anshu, Raghuvanshi C.S, "An Algorithm for Frequent Pattern Mining Based On Apriori" ,(IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 942-947

[9]. D.N Goswami, Anshu Chaturvedi, C.S Raghuvanshi," Frequent Pattern Mining Using Record Filter Approach", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 7, July 2010

