Implementation of an Algorithm for Mining Association Rules

Mr. Akshay R. Ingle[1], Mr. Amay A. Ambekar[2], Mr. Swapnil M .Chawre[3], Mr. Swapnil N. Kedar[4], Mr. Lokesh S. Mundafale[5].
Information Technology, DBACER, Wanadongari,
Nagpur, India

Abstract

The Purchasing of one product when another is purchased represents an association. Association rule are use to show the relationship between data-items.[5] Association rule are frequently used by retail stores to assist in marketing, advertising, floor placement, and inventory control. The discovery of such association can helps retailers develop marketing strategies by gaining insight into which item are frequently purchased together by the customers. In addition to their application in retail business, association rule can be used for other purposes as well such as prediction fault in telecommunication networks.[5]
We consider the problem of discovering association rule between item in a large database of sales-transactions.[9] For solving this problem we present an approach using fast algorithms, which are fundamentally different from known algorithm.

Keywords : Data mining, Association. Apriori Algorithm, Association Rule, Distributed Data Mining.

I. INTRODUCTION

Data mining has attracted a great deal of attention in the information industry and in the society as a whole in recent year, due to the wide availability of huge amount of data and the imminent need for turning such data into useful information and knowledge.[10]

The information and knowledge gained can be used for application ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. Data mining can be viewed as a result of the natural evolution of information technology.[10] The database system industry has witnessed an evolutionary path in the development of the following functionalities: Data collection and database creation, data management(including data storage and retrieval, and database transaction processing), and advanced data analysis (involving data warehousing and data mining). For instance, the early development of data collection and database creation mechanisms served as a prerequisite for later development of effective mechanisms for data storage and retrieval, and query and transaction processing. With numerous database system offering query and transaction processing as common practice, advanced data analysis has naturally become the next target.[3][10]

Data mining refers to extracting or "mining" knowledge from large amounts of data. The term is actually a misnomer. Just as the mining of gold from rocks and sand is referred to as gold mining rather than rock or sand mining, similarly data mining should be more appropriately named as "knowledge mining from data", which is unfortunately somewhat long. "Knowledge mining", a shorter term, may not reflect the emphasis on mining from large amount data.[6] Nevertheless, mining is vivid characterizing the process that finds a small set of precious nuggets from a great deal of raw material.[3][10] Thus, such misnomer that carries both "Data" and "Mining" became a popular choice. Many other terms carry a similar mining such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging.[3]

II. ASSOCIATION RULES : SUPPORT & CONFIDENCE

Now, if we think of the universe as the set of items available at the store, then each item has a Boolean variable representing the presence or absence of that item. Each set of item can be represented by a Boolean vector of values assigned to these variables. The Boolean vectors can be analyzed for buying patterns that reflect items that are frequently associated or purchased together.[7]
This patterns can be represented in the form of association rules. Consider the following example where a customer buys a computer and then tends to buy antivirus software along with it.[5] Association rule can be shown as:

Computer $\longrightarrow$ Antivirus_software [support=2%, confidence = 60%].

Rule support and confidence are two majors of rule interestingness. They respectively reflect the usefulness and certainty of discovered rule. A support of 2% for Association Rule means that 2% of all transactions under analysis show that the computer and antivirus are purchased together. A confidence of 60% shows that 60% of the customers who purchased a computer also purchased antivirus software. Typically association rules are considered interesting only if they satisfy a minimum support threshold and a minimum confidence threshold. Such thresholds are set by users.[7][9]

III. APRIORI ALGORITHM

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rule[1]. The name of the algorithm is based on the fact the algorithm uses prior knowledge of frequent item set properties, we shall see following. Apriori employs an iterative approach known as a level – wise search, where K – item sets are used to explore (K+1) – item sets. First, the set of frequent 1-item sets is found by scanning the database to accumulate the count for each item, and collecting those item that satisfy minimum supports. The resulting set is denoted by L1. Next, L1 is use to find L2, the set of frequent 2 - -item sets, which is used to find L3, and so on, until no more frequent K- item sets can be found.[1][2]

To improve the efficiency of the level – wise generation of frequent item sets, an important property called the Apriori property is used to reduce the search space.

Apriori property: All non empty subsets of a frequent item sets must also be frequent. The Apriori property is based on the following observation. By definition, if an item set I does not satisfy the minimum support threshold, min sup, the I is not frequent; that is P (I) < min sup. If an item A is added to the item set I, then the resulting item set (i.e. I [A]) can not occur more frequently than I. therefore I(A) is not frequent either ; i.e. P (I [A < min sup]).

The algorithm is subdivided into two phases:

1. Join Phase

2. Prune Phase

1) GENERATING K-ITEM SET:

In order to understand how the Apriori property is used in the algorithm, we need to first understand how the frequent K-item set (Lk) is found out using the previously found frequent (K-1)-item set Lk-1. for the same purpose , a two step process is followed, consisting of the following action:

JOIN STEP: To find Lk, a set of candidate K-item set is generated by joining Lk-1 with itself. This set of candidate is denoted as Ck. Let I1 and I2 be item set in the frequent (K-1) item set. By convention, the Apriori assumes that the item sets in the frequent item set are stored in lexicographic order. The join, Lk-1 X Lk-1, is performed, where member of Lk-1 are joinable if their first (K-2) items are in common. That isI1 and I2 of Lk-1should have all but the last element common and then only can they be joined. This condition only ensure that no duplicate are generated. The resulting item set thus formed is the candidate item set Ck containing all the possible candidate elements. [2]

PRUNE STEP : Ck is super set of Lk, that is, its members may or may not be frequent but all the frequent K-item sets are included in Ck. A scan of the database to determine the count of each candidate in Ck would result in the determination of Lk. Ck, however, can be huge, and so this could involve heavy computation to reduce the size of Ck, the Apriori property is as follows. Hence, if any (K-1) subset of candidate K-item set is not in Lk-1, then the candidate cannot be frequent either and so can be removed from Ck. This subset testing can be quickly maintaining a hash tree of all frequent item set.[1]

2) THE APRIORI ALGORITHM: EXAMPLE

Consider an example based on the transaction database 'D' shown in table 1. There are 9 transaction in the database, i.e. |D| = 9. We shall carry out the following steps one by one in order to find the frequent item set in D.

TID	List of item_IDs
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}

Table 1: Transaction database D

In the first iteration of the algorithm, each item is a member of the set of candidate 1-item set, C1. The algorithm simply scans all the transaction in order to count the number of occurrence of each item.

Suppose the minimum support count required is 2. The set of frequent 1-item set, I4, can then be determined. It consists of the candidate 1-item sets satisfying minimum support. In our example, all of the candidates in C1 satisfy minimum support

Scan D for count of each candidates

.

Compare candidate support count with minimum support count

Item set	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Figure 1: Transaction Count

IV. RESULT

Snapshots of Project :-

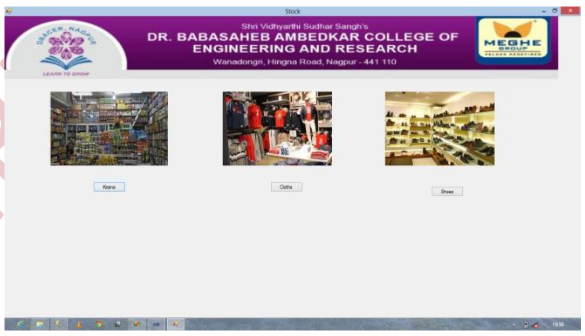
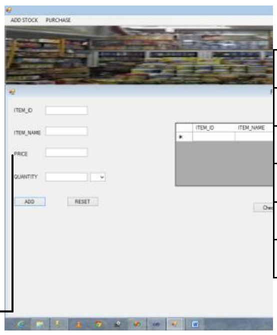

Stock
Shri Vidhyarthi Sudhar Sangh's
DR. BABASAHEB AMBEDKAR COLLEGE OF ENGINEERING AND RESEARCH
Wanadongri, Hingna Road, Nagpur - 441 110
MEGHE GROUP
Kirana
Cloths
Shoes

Figure 2: Stocks On Shop

Above snapshot shows three shops Kirana Shop, Clothes Shop and Shoes Shop which contain all stocks of products.



Item set	Sup. Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Figure 3:- Add Stoke

Above snapshot shows two tabs Add stoke and Purchase. In Add Stock we can add items by entering Item_id, Item_name, Item_price and Quantity.

The right hand side panel display all product in grid view which is entered by user.

V. CONCLUSION

This project study implemented Apriori algorithm in C# for finding frequent K-item set in a given large dataset. The Apriori algorithm is the seminal algorithm for discovering the frequent item set for a given a minimum support threshold. This threshold can be set by the user. The generated output contains all the frequent item sets generated along with their support. Also, a large dataset with about 10k rows and 1000 transaction items is fed as input which produces an appropriate output. Hence the implementation also works on the large datasets as inputs.

VI. REFERENCES

[1] CristianAflori, MiticaCraus, "Grid implementation of the Apriori algorithm", Science direct, Oct 2006.

[2] Aparna S Varde, Makiko Takakshi, Elke A Rundensteiner, Mathew O ward, "Apriori algorithm and game-of-life for predictive analysis in materials science", International journal of knowledge based and intelligent engineering systems 8(2004) .

[3] Jiawei Han and Micheline Kamber, "Data mining, Concepts and techniques", Morgan Kaufmann publishers, 2006.

[4] Azzedine Boukerche Kaiyuan Lu, "Optimized Dynamic Grid-based DDM Protocol for Large-scale Distributed simulation Systems", Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, 2005.

[5] U. Sakthi, R. Hemalatha, and R. S. Bhuvaneswaran, "Parallel and Distributed Mining of Association Rule on Knowledge Grid", World Academy of Science, Engineering and,Technology 42 2008.

[6] Lamine M. Aouad, Nhien-An Le-Khac and Tahar M. Kechadi, "Grid-based Approach for Distributed Frequent Itemsets Mining Using Dynamic Workload Management", School of Computer Science and Informatics University College Dublin.

[7] Albrecht Zimmermann and Luc De Raedt, "CorClass: Correlated Association Rule Mining for Classification", Institute of Computer Science, Machine Learning Lab, Albert-Ludwigs-University, 2004.

[8] Ian Foster, Carl Kesselman, Jeffreu M Nick and Steven Tneeke, "The physiology of the grid", University of Chicago, IBM corporation, NY.

[9] Schender, T, " Finding association rules that trade support optimally against confidence", PKDD 2001.

[10] Anupkumar,MehmedKantardzic and Samuel Madden, "Distributed data mining: framework and implementation", University of Louisville, Masachusetts institute of technology,Aug 2006.