Using Association Rule Mining for Extracting Product Sales Patterns in Retail Store Transactions

Pramod Prasad, Research Scholar
Department of Computer Science and Engineering
G. H. Raisoni College of Engineering
Nagpur, India
Email: pmpramod@gmail.com

Dr. Latesh Malik, Professor
Department of Computer Science and Engineering
G. H. Raisoni College of Engineering
Nagpur, India

***Abstract* – Computers and software play an integral part in the working of businesses and organisations. An immense amount of data is generated with the use of software. These large datasets need to be analysed for useful information that would benefit organisations, businesses and individuals by supporting decision making and providing valuable knowledge. Data mining is an approach that aids in fulfilling this requirement. Data mining is the process of applying mathematical, statistical and machine learning techniques on large quantities of data (such as a data warehouse) with the intention of uncovering hidden patterns, often previously unknown. Data mining involves three general approaches to extracting useful information from large data sets, namely, classification, clustering and association rule mining. This paper elaborates upon the use of association rule mining in extracting patterns that occur frequently within a dataset and showcases the implementation of the Apriori algorithm in mining association rules from a dataset containing sales transactions of a retail store.**

Keywords – data mining, association rules, apriori algorithm

I. INTRODUCTION

Retailing consists of the sale of goods or merchandise from a fixed location, such as a department store or boutique, in small or individual lots for direct consumption by the purchaser. Purchasers may be individuals or businesses. A "retailer" buys goods or products in large quantities from manufacturers or importers, either directly or through a wholesaler, and then sells smaller quantities to the end-user. Retailers are at the end of the supply chain. Retailers collect terabytes of data every day – such as transactional data, customer demographics and product sales based on parameters such as seasons and festivals. This data alone cannot enable good decision making for a retailer. It is necessary to discover and understand the underlying patterns involved in the organisation's operations from these data. Hence, there is a need present for accurate, timely information to react to changing market conditions, identify new customer segments, improve inventory management, and optimise overall store performance

The typical business decisions that the management of a retail store has to make include what to put on sale, how to design coupons and how to place merchandise on shelves in order to maximise the profit. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions [2]. Extraction of frequent item sets is essential towards mining interesting patterns from datasets. A typical usage scenario for searching frequent patterns is the so called "market basket analysis" that involves analysing the transactional data of a supermarket or retail store in order to determine which products are purchased together and how often and also examine customer purchase preferences. The Apriori algorithm introduced by Agrawal et al. [1] in 1994 is an efficient technique to generate all significant association rules between items in a database.

Section 2 of this paper describes the technique of association rule mining. Section 3 describes the working of the Apriori algorithm for generating significant association rules. Section 4 details our use of the Weka data mining tool for generating association rules from a sample dataset and our implementation of the Apriori algorithm to generate association rules from the sample dataset.

II. ASSOCIATION RULE MINING

Association rules are one of many data mining techniques that describe events that tend to occur together. The concept of association rules can be understood as follows: Let I = {i_1, i_2… i_n} be a set of literals, called items. Let T denote a transaction with a set of items such that T $\subseteq$ I. Given a database D of transactions (over I), an association rule is an implication of the form X$\rightarrow$Y, where X $\subset$ I, Y $\subset$ I and X $\cap$ Y = ϕ. Two important concepts when dealing with association rules are *rule confidence* and *rule support.* The rule X$\rightarrow$Y holds in the transaction set D with confidence *c* if c% of transactions in X also contain Y i.e. Confidence (X$\rightarrow$Y) = (no. of tuples containing both X and Y) / (no. of tuples containing A) = P (X|Y) = P (X $\cup$ Y) / P (X) The rule X$\rightarrow$Y has support *s* in the transaction database D if s% of transactions in D contain X $\cup$ Y i.e. Support (X$\rightarrow$Y) = (no. of tuples containing both X and Y) / (total number of tuples) = P (X $\cup$ Y).

Association rule mining enables the finding of rules of the form X $\rightarrow$ Y and X & Y $\rightarrow$ Z with minimum confidence and support. The challenge is the selection of algorithms that can be applied to mine association rules from a particular dataset. The formidable problem that awaits any such algorithm is the problem of dimensionality. The number of possible association rules grows exponentially with the number of attributes. If there are *k* attributes (considering only binary attributes such as *buy Chips = Yes*), there are on the order of $k.2^{k-1}$ possible association rules. For example, suppose that a small store has only 100 different items, and a customer could either buy or not buy any combination of those 100 items. Then there are 100 x 2^{99} possible association rules that await a search algorithm. The Apriori algorithm for mining association rules, however, takes advantage of structure within the rules themselves to reduce the search problem to a more manageable size.

III. THE APRIORI ALGORITHM

The Apriori algorithm was proposed by R. Agrawal and R. Srikant in 1994 [1] for mining frequent item sets to obtain strong Boolean association rules. A frequent item set is a set of transactions that occurs with a minimum specified support. A strong rule is one that satisfies both minimum support and minimum confidence. Apriori algorithm uses an iterative level-wise search, where k-itemsets (an itemset that contains *k* items) are used to explore *k+1* itemsets, to mine frequent itemsets from transactional database for Boolean association rules.

The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent item set properties. The basic methodology involved is to first find the set of frequent 1-itemsets (k=1). This set is denoted L1. L1 is then used to find the set of frequent 2-itemsets, L2, which is in turn used to find L3, and so on, until no more frequent k-itemsets can be found.

Algorithm Apriori: Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- *D*, a database of transactions;
- *min sup*, the minimum support count threshold.

Output:

L, frequent itemsets in *D*.

Method:

L_1 = find_frequent 1 - itemsets(D)
for (k = 2;$L_{k-1} \neq \phi$;k++)
    C_k = apriori_gen(L_{k-1})
for each transaction $t \in D$ // scan *D* for counts
    C_t = subset(C_k, t) // get the subsets of *t* that are candidates
for each candidate $c \in C_t$
    c.count++
L_k = {$c \in Ck | c.count >= min\_sup$}
return $L = \cup_k L_k$

procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
    for each itemset $l1 \in L_{k-1}$
      for each itemset $l2 \in L_{k-1}$
        if (l_1[1] = l_2 [1]) $\wedge$ (l_1 [2] = l_2 [2]) $\wedge$ ... $\wedge$ (l_1 [k-2] = l_2 [k-2]) $\wedge$ (l_1 [k-1] < l_2 [k-1]) then
          c = l_1 join l_2 // join step: generate candidates
        if has_infrequent subset(c, L_{k-1}) then
          delete c // prune step: remove unfruitful candidate
        else add c to C_k
    return C_k

procedure has_infrequent subset(*c*: candidate *k*-itemset)
L_{k-1}: frequent (*k*-1)-itemsets) // use prior knowledge
for each (*k*-1)-subset *s* of *c*
if $s \notin L_{k-1}$ then
return TRUE;
return FALSE;

Each iteration involves two steps – 1) Generate large k-itemsets and 2) Determine the support of each itemset using the transaction database. Infrequent itemsets are then pruned and strong rules are generated from the frequent itemsets. The algorithm is based on the property that any subset of a large itemset must be large. The working of the algorithm is explained through the following simple example –

Consider a retail store offering the following items for sale: {noodles, chips, biscuits, juice, cheese, ketchup, bread}. Let *I* denote this set of items. Customers coming to the store, pick up a basket, and purchase various combinations of these items (subsets of *I*). Here, the quantity purchased is purchased is not considered; only whether or not a particular item is purchased. Suppose Table 1 lists the transactions made at the store.

TABLE 1. TRANSACTIONS MADE AT THE RETAIL STORE

Transaction	Items Purchased
1	Bread, cheese, juice
2	Noodles, ketchup, juice
3	Juice, biscuits, chips, ketchup
4	Cheese, juice, biscuits, chips
5	Chips, noodles, bread
6	Ketchup, noodles, chips, biscuits
7	Biscuits, juice
8	Bread, biscuits, cheese
9	Ketchup, noodles, chips
10	Chips, juice
11	Cheese, bread, chips, ketchup
12	Noodles, chips, ketchup
13	Ketchup, juice, noodles, chips
14	Juice, cheese, bread, chips, biscuits

The two principal methods of representing this type of market basket data is using either the transactional data format or the tabular data format. The *transactional data format* requires only two fields, an *ID* field and a *content* field, with each record representing a single item only. For example, the data in Table 1 could be represented using transactional data format as shown in Table 2.

TABLE 2. TRANSACTIONAL DATA FORMAT FOR THE RETAIL STORE DATA

Transaction ID	Items
1	Biscuits
1	Cheese
1	Juice
2	Noodles
2	Ketchup
2	Juice
.	.
.	.

In the *tabular data format*, each record represents a separate transaction, with as many 0/1 flag fields as there are items. The data from Table 1 could be represented using the tabular data format, as shown in Table 3.

TABLE 3. TABULAR DATA FORMAT FOR THE RETAIL STORE DATA

Transaction	Noodles	Chips	Bread	Juice	Cheese	Ketchup	Biscuits
1	0	0	1	1	1	0	0
2	1	0	0	1	0	1	0
3	0	1	0	1	0	1	1
4	0	1	0	1	1	0	1
5	1	1	1	0	0	0	0
6	1	1	0	0	0	1	1
7	0	0	0	1	0	0	1
8	0	0	1	0	1	0	1
9	1	1	0	0	0	1	0
10	0	1	0	1	0	0	0
11	0	1	1	0	1	1	0
12	1	1	0	0	0	1	0
13	1	1	0	1	0	1	0
14	0	1	1	1	1	0	1

Applying Apriori on the above transactions and specifying the minimum confidence as 80%, the rules as shown in Table 4 are obtained.

TABLE 4. FINAL LIST OF ASSOCIATION RULES FOR RETAIL STORE DATA: RANKED BY SUPPORT X CONFIDENCE (MINIMUM CONFIDENCE 80%)

If *Antecedent,* Support then *Consequent*	Support	Confidence	Support X Confidence
If buy ketchup, then buy chips	6/14 = 42.9%	6/7 = 85.7%	0.3677
If buy noodles, then buy chips	5/14 = 35.7%	5/6 = 83.3%	0.2974
If buy noodles, then buy ketchup	5/14 = 35.7%	5/6 = 83.3%	0.2974
If buy bread, then buy cheese	4/14 = 28.6%	4/5 = 80%	0.2288
If buy cheese, then buy bread	4/14 = 28.6%	4/5 = 80%	0.2288
If buy noodles and chips, then buy ketchup	4/14 = 28.6%	4/5 = 80%	0.2288
If buy noodles and ketchup, then buy chips	4/14 = 28.6%	4/5 = 80%	0.2288

IV. IMPLEMENTATION OF APRIORI ALGORITHM

For our experimentation, we tested the working of the Apriori algorithm in Weka. Upon completion, we successfully implemented the Apriori algorithm in a Visual C#.Net application.

Weka (Waikato Environment for Knowledge Analysis) is a collection of machine learning algorithms for data mining tasks developed at the University of Waikato, New Zealand. Weka is free software available under the GNU General Public License. Weka supports several standard data mining tasks. Datasets may be given to Weka in ARFF (attribute-relation file format) or CSV (Comma Separated Values). We utilised a CSV file containing the names of the items and the transactions as specified in Table 5.

TABLE 5. . TABULAR DATA FORMAT FOR SAMPLE TRANSACTIONAL DATA

Transaction	A	B	C	D	E	F
1	1	1	0	1	0	1
2	1	1	1	1	1	0
3	1	1	1	0	1	0
4	1	1	0	1	0	0

The association rules generated by Weka are shown in Figure 1. The minimum support was specified as 40% and minimum confidence as 80%. Weka generates all possible association rules from the dataset. The number of generated rules, support and confidence may be specified by providing the corresponding parameters.

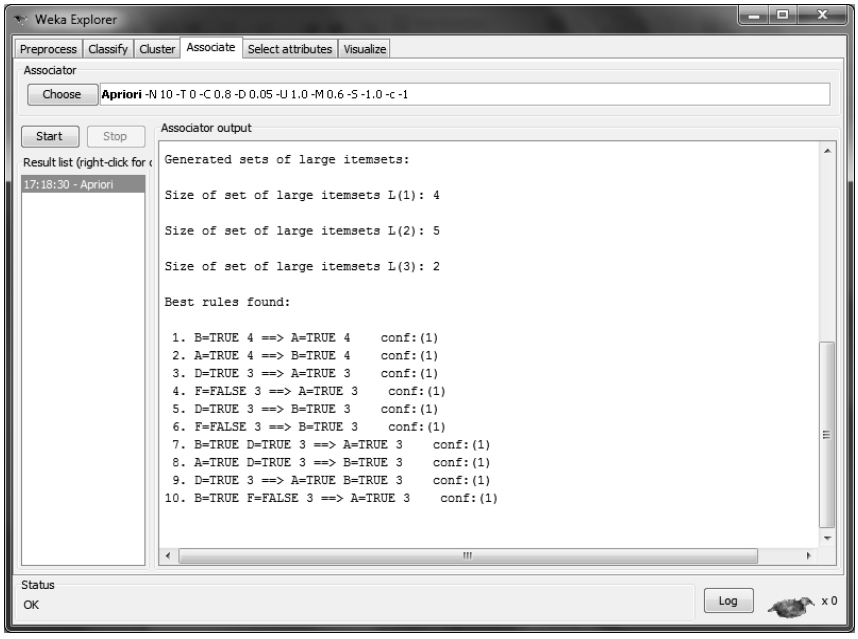

Figure 1. Association Rules generated through Weka

Subsequently, we developed an application implementing the Apriori algorithm using Visual C# for the transactional data as specified in Table 5. The items and transactions can be provided and the minimum support and confidence can also be specified.

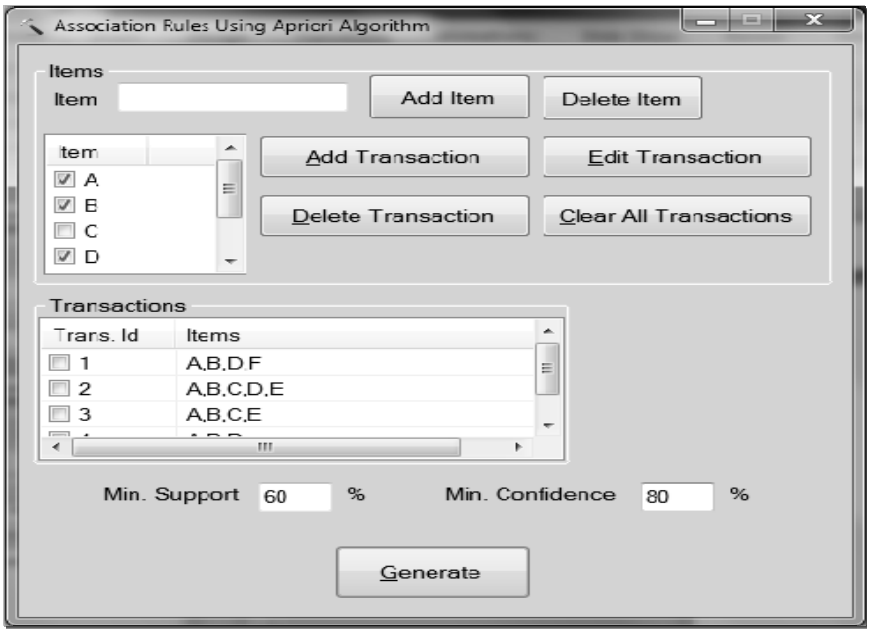

Figure 2. Specifying transactions in the application

The association rules generated by our application are consistent with the rules generated by Weka. We have restricted the rules that are generated only to items that go together. Weka, on the other hand, generates all possible rules including those for items that do not go together. The maximal association rules generated for the data in Table 5 is displayed in Figure 3.

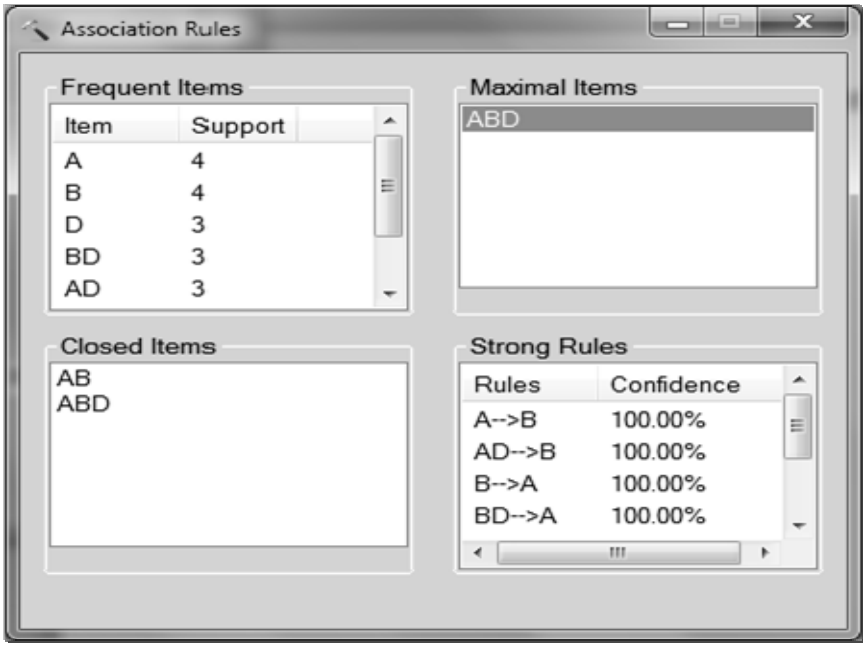

Figure 3. Association Rules generated through by the application

V. CONCLUSION AND FUTURE WORK

Use of an association rule mining driven application to manage retail businesses will provide retailers with reports regarding prediction of product sales trends and customer behaviour. This will allow retailers to make hands-on, knowledge-driven decisions. Our future work will focus on developing such an application with an aim to monitor and measure performance using key performance indicators (KPIs) based on historical information, current conditions and future goals, analyse sales trends and customer buying patterns to boost profitability and make accurate forecasts about future sales.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", Proceedings of the 20th VLDB Conference Santiago, Chile, 1994.

[2] R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", Proceedings of ACM SIGMOD Conference, Washington DC, USA, May 1993.

[3] Ana Azevedo and Manuel Filipe Santos, "A Perspective on Data Mining Integration with Business Intelligence", Information Science Reference, IGI, 2011.

[4] Conceptual Model of Business Value of Business Intelligence Systems, Ales Popovic, Tomaz Turk, Jurij Jacklic, Journal of Management, Vol. 15, 2010.

[5] Nizar Mabroukeh and C. Ezeiefe, "A Taxonomy of Sequential Pattern Mining Algorithms", ACM Computing Surveys, Vol. 43, No. 1, Article 3, Nov. 2010.

[6] M. Tiwari, "Data Mining: A Competitive tool in Retail Industries", Global Journal of Enterprise Information System, vol. 2, Issue 2, December 2010.

[7] E. W. T. Ngai, Li Xiu and D. C. K. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification", Elsevier, Expert Systems with Applications. vol. 36, p. 2592–2602, 2009.

[8] The WEKA Data Mining Software: An Update, Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, SIGKDD Explorations, Vol.1, Issue 1, 2009.

[9] K. Shyamala and S. P. Rajagopalan, "Mining Essential and Interesting Rules for Efficient Prediction", AJIT, Vol. 6, 2009.

[10] Rajesh Natarajan and B. Shekar, "Tightness: A Novel Heuristic and a Clustering Mechanism to Improve the Interpretation of Association Rules", IEEE IRI, Las Vegas, Nevada, USA, 2008

[11] Tong Gang, Cui Kai and Song Bei, "The Research & Application of Business Intelligence System in Retail Industry", Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 2008.

[12] Andreas Fink et al., "Advances in Data Analysis, Data Handling and Business Intelligence", Proceedings of the 32nd Annual Conference of the Gesellschaft für Klassifikation e.V., Joint Conference with the British Classification Society (BCS) and the Dutch/Flemish Classification Society (VOC), Helmut-Schmidt-University, Hamburg, 2008.

[13] Chunhua Ju and Dongjun Ni, "Distributed Mining Model and Algorithm of Association Rules for Chain Retail Enterprise", ISECS International Colloquium on Computing, Communication, Control, and Management, 2008..

[14] Hamid Rastegari and Mohd. Sap, "Data Mining and E-Commerce : Methods, Applications and Challenges", Journal of Information Management, Vol. 20, 2008.

[15] Hongwei Liu, Bin Su and Bixi Zhang, "The Application of Association Rules in Retail Marketing Mix", Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, 2007