3rd International Conference on Advanced Research in Biology, Engineering, Science and Technology (ICARBEST'16)
Organized by
International Journal of Advanced Research in Biology, Engineering, Science and Technology (IJARBEST)
19th March 2016

Quine-McCluskey Methodology for Predicting Customer Buying Nature

[1]Kumaran.R, [2] Siva Murugan. C,
[3]Puvaneshwaran.R
Department of Computer Science and Engineering ,
VV College Of Engineering, Tisaiyanvillai.
[1]rkumaran210@gmail.com,[2]sivamurugancsm@gmail.com,
[3] puvaneshwaran2010@gmail.com

[4]Immanuel Gem Issac. A. J
Asst.Prof., Department of Computer Science & engg,
VV College Of Engineering, Tisaiyanvillai.
immanuelgem@vvcoe.org

Abstract

Finding frequent item sets is one of the most important fields of data mining and also finding the rules associated with the item set is another important ground in association rule mining. The process of finding new techniques to reduce candidate generation in order to generate frequent item sets efficiently is a challenging task and generating rules in order to know the purchase behavior of the customer to improve the business. Rule-based systems generate many of the redundant rules; such rules are expensive especially in online systems. Currently, there are many of the available rule minimization techniques; however, they still suffer from high complexity and lack of efficiency. In this paper, we introduce a novel method (QMR) based on Quine-McCluskey (Q-M) algorithm and Apriori Algorithm. The novelty of our algorithm is in the adaptation of Q-M that is used in reducing Boolean expressions to the rule minimization. Our minimization method is very simple and supports many items (variables) and the apriori algorithm is selecting the most frequent item sets efficiently.

1. INTRODUCTION

1.1 Introduction

Computerized data gathering tools and grown-up database technology lead to remarkable amount of data store in databases, data warehouses and other information repositories. The entire answer for knowledge innovation can be given by data mining technology. A large number of current systems are based on association rules. Such rules are usually generated from different set of items. This set of association rules in rule based systems can grow in unexpected way. In fact, the larger the frequency of the item set, the more the association rules. Certainly, large number of rules is greatly affecting the performance and efficiency of rule based systems.

1.2 Association Rules

The main idea behind the association rules is building a relation between specific values of categorical variables in large data sets. Association rules process is a common task in many data mining processes as well as in the text mining algorithms. Association techniques are powerful tools in many applications including network filtering and Iris applications. The main advantage of these rules is that they allow discovering hidden patterns in large data sets such as "customers whom order laptops often order external hard disk or a flash memory." However, the problem with association rules are the large number of rules that need to be visited every time the application needs to take a decision. For instance, a network firewall or a spam filter has to go through all of the rules every time a packet/message arrives. The association rules simply consist of a set of discrete attributes $A_i = \{a_1, a_2, \ldots, a_m\}$. Assume $D = \{T_1, T_2, \ldots, T_N\}$ is the number of transactions over the relation schema A_i. In addition, assume that there is an atomic proposition in the form of $(value_1 \leq attribute \leq$

$value_2$) as well as attribute = value for ordered and unordered attributes giving that $value_1$, $value_2$, and value belong to D. Moreover, an item set is a conjunction of atomic conditions or items where the number of items in an item set is called the length of the item set. Therefore, the rule is represented as an extended association rule in the form of $X \rightarrow Y$, where both X and Y are item sets.

2. PROPOSED SYSTEM

2.1 Proposed Model

In order to reduce the time and space in generating the frequent item sets, and also to predict the customer purchase behavior we introduce a new algorithm called Customer Purchase Behavior (CPB) based on Quine-McCluskey and Apriori. The algorithm contains two phases – Generating frequent item sets and forecasting the customer behavior on purchase of frequent item sets.

2.1.1 Quine-McCluskey for Rule Mining (QMR)

Association rules are extracted from the system database by counting all of the possible combination of attributes. With large number of attributes, the computational time of the system increases exponentially. For instance, given m number of records and n attributes, the enumeration to the attributes requires ($m \times 2n$) steps in which with large n, the computation is unfeasible.

In this part of the paper, we introduce QMR as a rule mining algorithm. The algorithm works in two steps; the data is encoded in the first step using binary number system and the Quine-McCluskey is utilized in the second step.

Step 1: Subset Generation

A subset is a part of a set. A is a subset of B if and only if every element of A is a element of B. This can be denoted as A ⊂ B.

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

A Set of n- elements has 2n subsets (contains an empty set). This can be given by using binomial sum. When n=1, 2, 3 ... the total number of subsets are 2, 4, 8 ... When n=1 (a set contain only one element), the possible subsets are {ø}, {1} i.e. 21 in our case we are not going to consider any null values. Therefore the total number of subsets used here are $2^n - 1$ in number.

Step 2: Bit Vectors

The Given transaction database is converted in to bit vectors. Consider when the database contains only 2 items; there is possible transaction types can be represented in 22 = 4 combinations.

The combinations are 00,01,10,11 Here the first position represent the item1 transaction, second position represents the item2 transaction. 00 says that no items are transacted. 01 represents item1 is not transacted and item2 is transacted.

Here 00 is not considered anyway for the transaction database. Therefore, the total number of transaction for n-items can be represented as $2^n - 1$.

3. RULE BASED SYSTEM

3.1 Predicting Rules for Sample Dataset

Suppose we want to decide how to control the AC in the classroom. Attributes are {Temperature, Humidity}. They can have the following values:

Temperature = {high, Medium, Low} Humidity = {high, Medium, Low} S is a collection of 9 examples with 5 High and 4 Medium then

Entropy(S) = -(5/9) Log2(5/9) - (4/9) Log2 (4/9) = 0.9901

Gain(S,Temperature)= Entropy(S) – (3/9)Entropy(S_{High}) - (3/9)Entropy(S_{Meduim}) - (3/9)Entropy(S_{Low}))

Entropy(S_{High}) = -(2/3) Log2 (2/3) - (1/3) Log2 (1/3) = 0.9183

Entropy(S_{Meduim}) = -(3/3) Log2 (3/3) = 0

Entropy(S_{Low}) = -(3/3) Log2 (3/3) = 0

Gain(S,Temperature) = 0.9183-(3/9)* 0.9183–0–0 = 0.6122

Gain(S, Humidity) = Entropy(S) – (3/9)Entropy(S_{High}) - (3/9)Entropy(S_{Meduim}) - (3/9)Entropy(S_{Low})

Entropy(S_{High}) = -(2/3) Log2 (2/3) - (1/3) Log2 (1/3) = 0.9183

Entropy(S_{Meduim}) = -(2/3) Log2 (2/3) - (1/3) Log2 (1/3) = 0.9183

Entropy(S_{Low})) = -(2/3) Log2 (2/3) - (1/3) Log2 (1/3) = 0.9183

Gain(S,Humidity)=0.9901-(3/9)*0.9183-(3/9)*0.9183-(3/9)*0.9183=0.0718

Table 1. Sample Dataset

Temperature	Humidity	AC Control
High	High	High
Medium	High	Medium
Low	High	High
High	Medium	Medium
Medium	Medium	Medium
Low	Medium	High
High	Low	High
Medium	Low	Medium
Low	Low	High

Temperature attribute has the highest gain, therefore it is used as the decision attribute in the root node. Since Temperature has three possible values, the root node has three branches (High, Medium and Low). The next attribute is the remaining attribute Humidity.

Temperature	Humidity	AC Control
High	High	High
High	Medium	Medium
High	Low	High

Entropy (S_{High})= = -(2/3) Log2 (2/3) - (1/3) Log2 (1/3) = 0.9183

Temperature	Humidity	AC Control
Medium	High	Medium
Medium	Medium	Medium
Medium	Low	Medium

Entropy(S_{Meduim}) = 0 ("perfectly classified")

Temperature	Humidity	AC Control
Low	High	High
Low	Medium	High
Low	Low	High

Entropy(SLow) = 0 ("perfectly classified") Therefore, rules controls the AC could be expressed as follows:

1. if (Temperature, Low) then (AC Control, High).
2. if (Temperature, Medium) then (AC Control, Medium).
3. if (Temperature, High) and (Humidity, Low) then (AC Control, High).
4. if (Temperature, High) and (Humidity, High) then (AC Control, High).
5. if (Temperature, High) and (Humidity, Medium) then (AC Control, Medium).

Rule Generation

As shown in Table 1, the temperature attribute is defined with three values (High, Medium and Low). Accordingly, this attribute can be encoded in a 2-bit form (T_1T_0). Similarly, the humidity can be also encoded in a 2-bit form (H_1H_0). The AC control can be represented by one bit (1 for High and 0 for Medium).

minterms	T_1	T_0	H_1	H_0	AC Control
m5	0	1	0	1	1
m6	0	1	1	0	1
m7	0	1	1	1	1
m9	1	0	0	1	0
m10	1	0	1	0	0
m11	1	0	1	1	0
m13	1	1	0	1	1
m14	1	1	1	0	0
m15	1	1	1	1	1

The steps required to **apply the (Q-M) method to minimize** High(T1,T0, H1, H0) is illustrated as follows.

Step 1: Translate to minterms if not given in SOP form.

Step 2: Translate to minterms in binary notation if not given in binary minterms form.

High(T_1, T_0, H_1, H_0) = $\sum(0101, 0110, 0111, 1101, 1111)$

D(T_1, T_0, H_1, H_0) =

$\sum(0000, 0001, 0010, 0011, 0100, 1000, 1100)$.

Step 3: Form table with six attributes, the first three columns initially include number of ones, minterm's symbol and minterm's binary form, respectively. The rest of columns are reserved for (M-Q) operations that require (n-1) cubes where n is the number of variables.

Step 4: Compare minterms in adjacent block looking for situation in which terms only differ in one column. Place terms in next column with missing literal replaced by (-).

Step 5: Compare minterms in adjacent block at 1-Cube column looking for situation in which terms only differ in one column. Place terms in 2-Cube column with missing literal replaced by (-).

Step 6: Form Prime Implicant Table with prime implicants listed in a column and the minterms of High(T1,T0, H1, H0) listed across a row.

Step 7: Search a first column with only one cross mark. The column of minterm (0110) has one cross mark at the prime implicant (0---). Mark the prime implicant's row with (*) and the minterm's column with check mark.

Step 8: Repeat step 7 for the rest of minterms.

Step 9: Translate to literal notation

High(T_1, T_0, H_1, H_0) = T1 + T0 H0

When the steps are repeated for Med(T1,T0, H1, H0), the result will be:

Med(T_1, T_0, H_1, H_0) = □T0 + T1□H0

As shown above, the given rules can be reduced to four rules as follows:

1. if (Temperature, Low) then (AC Control, High).
2. if (Temperature, High) and (Humidity, Low) then (AC Control, High).
3. if (Temperature, High) and (Humidity, High) then (AC Control, High).
4. if (Temperature, Medium) then (AC Control, Medium)
5. if (Temperature, High) and (Humidity, Medium) then (AC Control, Medium).

4. PERFORMANCE ANALYSIS

4.1 Predicting Customer Purchase Behavior

The conditional probability plays a major role in the generation of rules in transaction databases normally. When an item set satisfying both minimum support threshold and minimum confidence threshold will be considered as the strong item set and the association rules may be generated from the item set. Here the conditional probability plays a major role. When a customer purchases items A and B, the conditional probability of A given B is defined as joint probability of A and B, and the probability of B:

$$P(A/B) = \frac{P(A \cap B)}{P(B)}$$

The equation saying that the space for judging the frequent item set or the strong item set is reduced because of B.

By applying the Quine- McCluskey method, the rule generation for predicting the customer purchase behavior is done. For that, Group the itemsets according to the number of ones. This is called as primary clustering. Each itemset in one cluster is compared with every other cluster. This technique is called as matching process.

Any two itemset with one itemset differ from only one item can be combined. The unmatched variable can be replaced as - (hyphen). This shows that the item may or may not be purchased. The itemsets in one cluster are combined with the next following cluster only. Any two itemsets differing by more than one bit cannot match. If any two itemsets are the same in every position except a position, a

tick mark is placed to the right of both the itemsets to show that they are transacted under combination.

This procedure is repeated until the itemsets in one cluster cannot make a match with any other itemsets. The unchecked itemsets in the tables are used for the analysis of the purchase behaviour of customers. A 1(one) under the item shows that the item is purchased and 0(zero) under the item shows that the item is not purchased. The – (Hyphen) shows that the item may or may not be purchased. For example if a transaction represents 0 – 1 1 for the itemset {A, B, C, D}, it shows that When item A is not purchased then items C and D are purchased together. Here item B may or may not be purchased.

From the above analysis, the following rules can be framed.

Rule 1: Buys (X,"C") → Buys (X,"D") [Purchase Nature: 50%]

Rule 2: Buys (X,"C") → Buys (X,"D") Buys(X, "A, E")^ ¬Buys (X,"A, E") [Purchase Nature: 50%].

5. EXPERIMENTAL RESULTS

5.1 Experimental Results

In order to test the proposed algorithm, the data mining tool, Weka is utilized. The transaction dataset of supermarket 18 transactions is taken with 5-items for testing.

This dataset is given as the input for the Apriori algorithm, CBVAR, ICBV and our proposed method for the frequent item sets generation. The Apriori algorithm and CBVAR are not removing the redundant transactions thus leads to processing of lengthier database. The ICBV is removing the duplicates but it is taking more tables to generate the frequent item sets.

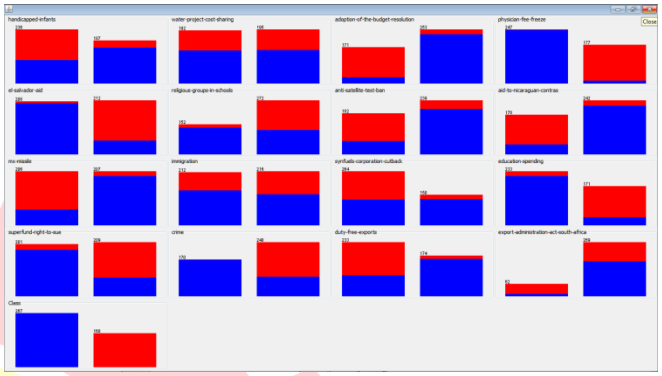

Figure 5.1 Voting Dataset for 16 Categories

Figure 5.2 Supermarket Dataset with 20 transactions

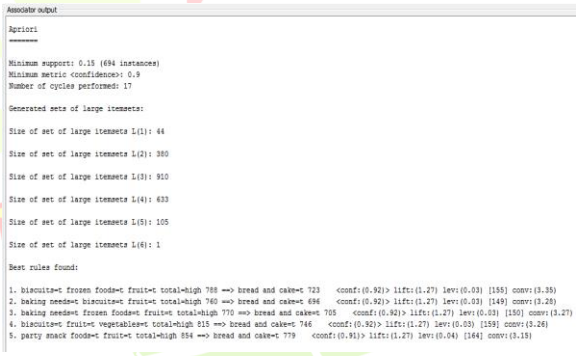

Figure 5.3 the associate rules of the supermarket dataset

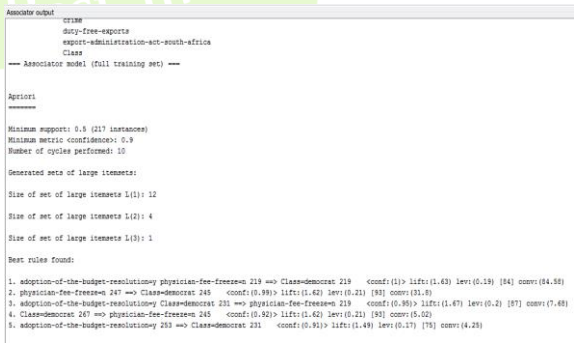

Figure 5.4 the associate rules of the voting dataset

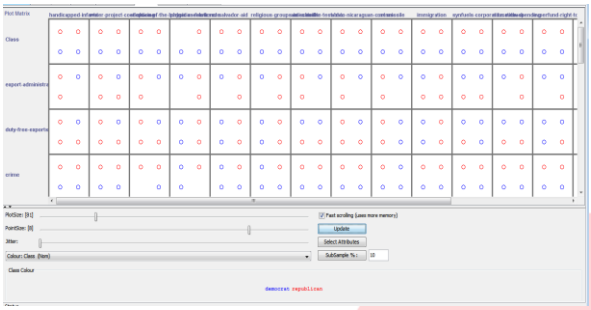

Figure 5.5 the plot matrix for the voting dataset

6. Conclusion and Future Work

The Proposed algorithm uses single scan, which reduces the database scans and hence the computation time taken is also very less for the frequent itemsets generation. It uses fewer amounts of steps to generate the frequent itemsets with duplicate transaction elimination. The strong rules for judging the customer purchase behaviour has been generated using Quine- McCluskey method which is a new concept in Association Rule Mining. Future work in this direction could be the use of minimum space and scan in the customer purchase behaviour, since an item in one cluster is compared with all the other itemsets in the next cluster leads to more time in the generation of associated rules.

References

[1] Ashok Savasere, Edward Omiecinski, and Shamkanth Navathe, "An Efficient Algorithm for Mining AssociationRules in Large Databases" In VLDB, Zurich, Switzerland, pp.432-443,1994.

[2] Banu Ozden, Sridhar Ramaswamy, Avi Siberschatz R: "Cyclic Association Rule", In Proceedings of Fourteenth International Conference on Data Engineering, pp 412-425,1998.

[3] Bifet Albert, Geoff Holmes, and Ricard Gavalda Mining Frequent Closed Graphs on Evolving Data Streams", In Proceedings of the 17 the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 591-599, 2011.

[4] Chaohui Liu, Jiancheng an, The Software Engineering School, China "Fast Mining and Updating Frequent Itemsets",ISECS International Colloquium on Computing, Communication, Control and Management, Vol.1, pp. 365-368,2008.

[5] F.Berzal, J.C.Cubero, N.Marin, J.M.Serrano, TBAR "An Efficient Method for Association Rule Mining in Relational Databases", In Elsevier, Data and Knowledge, Engineering Vol 37, pp. 47-64, 2001.

[6] Han J, Pei J, Yin Y "Mining Frequent Patterns without Candidate Generation" Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, ACM press, pp. 1-12, 2000.