International Journal of Engineering Research & Management Technology ISSN: 2348-4039

November-2015 Volume 2, Issue-6

Email: editor@ijermt.org www.ijermt

Data Mining Technique to Reduce Redundancies and Fraud Detection in Credit Card Billing System

Sovers Singh Bisht[1], Mayank Maheshwari [2]

IIMT College Of Engineering[1-2]

Greater Noida (U.P), India

ABSTRACT:

Credit card transactions are growing every day in number by taking a larger share in Indian financial sector. Acquiring a large market share in the payment system and leading to a higher rate of stolen account numbers and subsequent losses by banks. The increased rate of transactions gives rise to many fraudulent charges in form of redundancies and application fraud. The aim in data mining is to reduce redundancies in transactional database or customer database useful in credit card collections. This approach is to solve the query of frequently occurring item sets and generating strong association rules by compressing the data at a larger extent. This paper presents to find the fraud in credit card mechanism and examine the result based on the approach. This reduces the cost associated with higher interest rates and its charges.

Keywords: Credit card, Fraud detection, Association rule mining algorithm, Huffman algorithm and frequent item sets mining**.**

1. INTRODUCTION:

With the increased dissemination of bar code scanning technologies it was possible to accumulate vast amounts of Market-basket dataset containing millions of transactions. The information collected over these transactions and collected over a data warehouse, gave rise to many forms of patterns and there data. These patterns associated with the payment industry and financial sector gave rise to credit card payment mechanism which might be online or through an EDC machine. The collection industries painful and annoying mechanism to extract debt gave rise to many questions pertaining to payment already done, fraudulent charges or other application frauds and charges which lead to harassment of consumer as well as its business. Thus the need arose to study the patterns of consumer consumption that could help in improving the marketing infrastructure and related disciplines like targeted marketing. Association Rule Mining is a data mining technique which is well suited for mining Market-basket dataset. The technique for association rule mining is divided in two parts, first it mines necessary frequent itemsets from large datasets, and second one is generating fast association rules from previously mined datasets. Large scale data mining techniques can be improved on the state of the art in commercial practice. Techniques which are scalable to analyze massive amount of transactional data can efficiently detect fraud in timely manner which is an important problem especially for e-commerce and collection industry. The techniques devised have to deal with data which is highly skewed and managing data becomes a cumbersome problem. Therefore our approach deals with the frequent item sets occurring in a credit card that might by fraudulent occurring with charges and generating highly compressed data which is susceptible to mining and extraction in favor of consumer

1.1 PROBLEM STATEMENT:

Given a set of credit card transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction. The following is an illustrative example of market basket dataset of a credit card taken as a sample.

Table 1: A data base with 5 transactions and item sets is given below

TID	Transactions
1	books, stationary
2	Books,bags,grocery,utensils
3	Stationary,bags,grocery,coke
4	books,stationary,bags,grocery
5	Books,stationary,bags,coke

Market based transaction setfor the above dataset the following rule holds well.{books}-> {stationary}
The rule can be read as, "customers who buy books also tend to buy stationary."

1.2 FORMAL DEFINITION:

Let I = {i_1, i_2, ..., i_m} be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that T subset I. The problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support (called minsup) and minimum confidence (called minconf) respectively.

1.3 KEY CONCEPTS:

1. ITEMSET:

A collection of one or more items in a market based credit card transactions.
Consider a set of literals I = {i_1, i_2... i_m} then I is called item set pertaining to database generated.

2. ASSOCIATION RULE:

An association rule is an implication of the form X->Y where X and Y are the itemsets.

3. SUPPORT:

Support measures fraction of transactions that contain both X and Y Given a rule X->The support gives the approval of consumer pertaining to the number of minimum transactions he might have done to generate the database.
Support = (X union Y)/ N
Where N is the total number of transactions

4. CONFIDENCE

Confidence measures how often item in Y appear in transactions that contain X. Given the rule X->Y:
Confidence = X union Y / X

5. LARGE ITEMSET:

Itemsets with minimum support (minsup) and minimum confidence (minconf) are called as large itemsets, while others that do not cross minimum support and minimum confidence values are known as small itemsets.

6. K ITEMSET:

An itemset with k number of items is referred to as k-itemset.

7. CANDIDATE ITEMSETS:

A set of item sets which are generated from a seed of itemsets which were found to be large in the previous pass. Large itemsets for the next iteration are selected from the candidate itemsets if the support of the candidate itemsets is equal to or larger than minsup and minconf

8. ASSOCIATION RULE MINING TASK:

Given a set of transactions T, the goal of association rule mining is to find all rules having
Support ≥ minsup threshold
Confidence ≥ minconf threshold

1.4 PROPOSED ALGORITHMS:

APRIORI ALGORITHM:

INPUT:
The market base transaction dataset.

PROCEDURE:
1. The first pass of the algorithm counts item occurrences to determine large 1-itemsets.
2. This process is repeat until no new large 1-itemsets are identified.
3. (k+1) length candidate itemsets are generated from length k large itemsets.
4. Candidate itemsets containing subsets of length k that are not large are pruned.
5. Support of each candidate itemset is counted by scanning the database.
6. Eliminate candidate itemsets that are small.

1.5 OUTPUT:
Itemsets that are "large" and qualify the min support and Min confidence thresholds are redundant and generate difficulty in generating the candidate keys. But whenever we in have lager data sets like where frequency of itemsets is huge enough for example credit card collections and transactional data we find difficulty in generating the candidate key of frequent itemsets thus in this paper we have tried to solve this problem by using an extended version of Huffman algorithm.

2. APPLYING THE PROPOSED PROCESS:
Whenever data is retrieved from a transactional database the frequency of occurrence of symbols is much higher as expected outcomes. Whenever a data is mined it takes space and time which lead to fast growing tremendous amount of data in database. Our approach is generally based upon the textual data where data is compressed and minimization of redundant data is done in sample data. We know that entropy of any given data is the amount of information content which the data is about to deliver thus we know that the difference in the entropy and average size of code length serves as an indicator for redundancy in the codes. Higher the difference, higher the redundancy Our overall objective is representing the frequent itemsets in transactional database with its respective probability of occurrence of elements and reducing the size of itemsets to be mined so that we can generate minimum redundancy in large database and get an optimum solution.

The FP tree based mining algorithm is also an alternative to apriori which is based on the divide and conquer approach. First it compresses the database representing frequent item into a frequent pattern tree, or FP-Tree, which retains the itemsets association information. It then divides the compressed database into a set of conditional databases; each associated with one frequent item and mines each database separately.Thus our approach is the construction of efficient binary tree of frequent itemsets within a large database and with respect to probability of occurrence of each itemset we can generate an effective binary tree with reduced redundancy and size which can provide better mining opportunities.

2.1 PROCESS OF APPLICATION ON ATRANSACTION DATABASE:

Suppose we have a transactional database where frequency of occurrence of itemsets is given by the following table (conceptually we will take a small set of data items with higher frequency).

Table 2: A data base with 3 transactions and 3 itemsets is given below as an example with a higher support county as well as lower support count.

TID	ITEMSETS	FREQUENCY	PROBABILITY
1	Books	85	.85
2	Bags	15	.10
3	Stationary	05	.05

The Huffman code on data mining transitional database does not yield good performance when the probabilities are skewed. The solution to this is done through the extended Huffman algorithm by grouping more than one symbol in one group and the Huffman code is obtained for the whole group. The major advantage of extended Huffman algorithm is that it works best for the skewed database and transactional database with respect to credit card collections or market basket analysis has one of the most skewed probability distribution.

Suppose N itemsets are present in the database transaction stream and suppose we frame a group of M symbols where Huffman code for every group will be generated. In this way we can form total number of groups that would be N^M.For the example taken above if we form the group of two itemsets then the total number of symbols would be 3^2=9.Thus if we form the group of 3 itemsets we get a much compresses data in highly reduced form but we leave our assumption up to group of two itemsets.

Now, by multiplying the probability of each symbol we will find the probability of occurrence of group.

Table 2: Shows a group of itemsets with their respective probabilities.

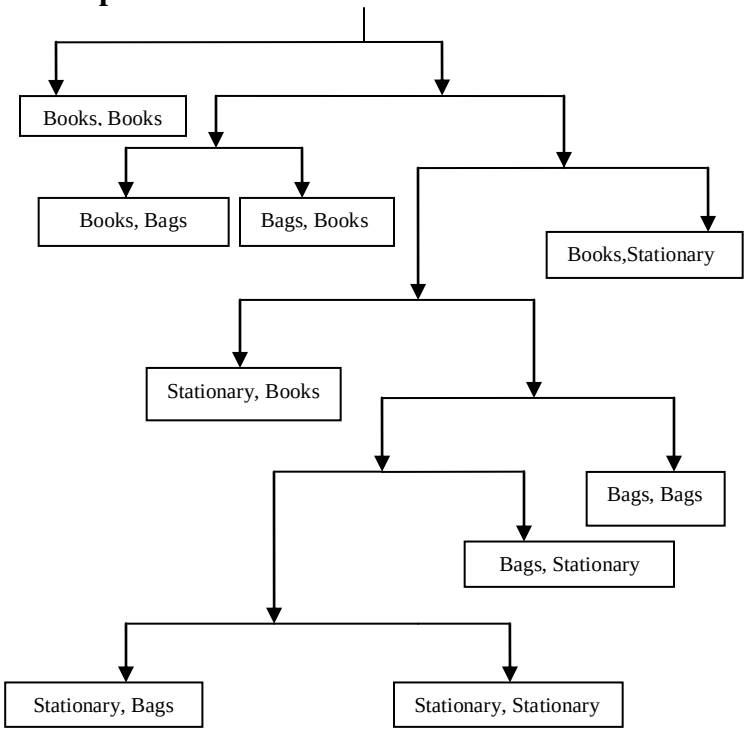
GROUP	PROBABILITY
Books, Books	.85*.85=.722
Books, Bags	.85*.10=.085
Books, Stationary	.85*.05=.042
Bags, Books	.10*.85=.085
Bags, Bags	.10*.10=.010
Bags, Stationary	.10*.05=.005
Stationary, Books	.05*.85=.042
Stationary, Bags	.05*.10=.005
Stationary, Stationary	.05*.05=.0025

Huffman Tree for the groups of data items is shown below.

Steps to construct a Huffman tree are as follows.

1. First arrange in decreasing order of probability.
2. Second construct the Huffman tree by keeping highest probability on the left side and lowest on the right.
3. Construct the Huffman tree from the list of itemsets
4. Assign the bits i.e. left child is assigned ‘0’ and the right child is assigned’1’.Traverse from the root node of the Huffman tree to the left containing a particular symbol. This traverse will generate a code for that symbol.

Figure: Representation of Extended Huffman Tree.



Thus the binary tree has many applications as in many search applications where data is constantly entering/leaving, such as the map and set objects in many language libraries. They are also used in high bandwidth router for storing router tables.

From the codes generated by the tree we can easily say that no two item sets have the same codes so that data can be saved efficiently without overlapping. This representation of grouped item sets is highly efficient in compression of large frequent item sets to a smaller data structure, divide and conquer search method and partitioning based method.

Table 3: The Huffman Codes for the groups of data items is shown below.

Group	Probability	ExtendedHuffman Code
Books, Books	. 722	0
Books, Bags	.085	100
Books, Stationary	.042	111
Bags, Books	.085	101
Bags, Bags	.010	11011
Bags, Stationary	.005	110101
Stationary, Books	.042	1110
Stationary, Bags	.005	1101000
Stationary, Stationary	.0025	1101001

Let us evaluate the performance of the groups of symbol i.e. Huffman code
Average length of group of symbol=1.15 bits/itemset
Entropy= - $\sum_{i=1}^{N}$ Pi ∗ log(Pi)
=.1.99+.332+.216
=.747 bits/itemset
Thus the redundancy of this code is= (1.15-.747)bits/itemset
This is 54 % of the redundancy.
Let us evaluate the performance of the groups of symbol now with extended Huffman code. From above table we may compute the average number of bits that requires representing a group of two characters.
Average length of group of symbol=1.658 bits/group.
Thus if we calculate the information gain or entropy of the group of item sets as under
Entropy= - $\sum_{i=1}^{N}$ Pi ∗ log(Pi)
=1.490 bits/group
Therefore redundancy here in generating Huffman code for the group.
The redundancy=1.658-1.490=.168 bits/group
Therefore the information gain is only 11.2 % of the Entropy value it proves that if we generate the extended binary tree for the given itemsets in a transactional database we can remove the redundancy to a much higher extent .The above given example shows that we are left with only 11.2% of the redundancy. Thus if we extend the grouping of itemsets we can remove the redundancy to much higher extent as compared to typical Huffman code .

2.2 CONCLUSION OF THE APPROACH:

We have proposed an approach of frequent pattern mining of large databases using an extended Huffman algorithm where we generate probabilities of itemsets which are appearing frequently and are in huge distribution within the transactions of large itemsets. The credit transactions are based on a consumer's choice but when it comes to billing, a large amount of data set is generated and which has to be resolved by the bank.In this approach we have several advantages when we try mining large databases and also with we use algorithm like FP-growth such as:

1. The tree generated is highly compressed and the redundancies of itemsets have been highly reduced due to the grouping of itemsets, it maintains necessary semantics for frequent itemsets mining problem and is a more efficient approach to compress data than traditional compression algorithms.
2. The approach saves the time and cost of scanning highly large database in load, refresh and transform applications of scanning a large database and compression leads to avoidance of unnecessary charges on a consumer.
3. This approach applies a divide and conquers growth approach which avoids generation of large candidate keys from the transactions,by reducing the steps and size of the data sets to be searched. This prevents any financial loss of the consumer and even avoids uneven debt collection practices as there is no place for fraudulent charges.
4. This process generates binary codes which are unique and are applied to reduce redundancy to a much larger extent and repetition of data is avoided in large databases.
5. Finding probabilities of item sets makes it easier to generate the codes of the itemsets which are highly recommended for fast moving data streams such as real time traffic and network monitoring. Hence compression becomes easier and faster.
6. This approach builds a concept that itemsets which are large can be assimilated and then after reducing their count can be assimilated again to form tight bonds. Hence strong association rules can be generated by which analysis becomes easier in large databases with itemsets occurring frequently.

7. The transactions carried with the credit card should be secure one and it should be able to detect the fraudulent transactions carried out by fraudsters.

REFERENCES:

1. An approach to extract efficient frequent patterns form transactional database-Mamta Dhanda.
2. Introductions to algorithms-T.H.Cormen, C.L.Leiserson, R.L.Rivest.
3. Data Mining: Concepts and Techniques: Concepts and Techniques -Jiawei Han, Micheline Kamber, Jian Pei.
4. R.Agrawal and R.Srikant. Fast algorithms for mining association rules.In VLDB'94, pp.487 {499.
5. G.Grahne, L. Lakshmanan, and X. Wang. Efficient mining ofconstrained correlated sets. ICDE'99.
6. Mining frequent patterns without candidate generation- Jaiwei Han,Jian Pei,Yiwen Yin
7. R.Agarwal, C.Aggarwal, and V.V.V.Prasad.A tree projection algorithm for generation of frequent itemsets. In J.Parallel and Distributed Conputing, 2000.
8. Construction of FP tree using Huffman coding-Dr.S.N.Patro, Prof.Sujogya, Mishra, Mr.Pratyusabhanu Khuntia and Mr.Chidananda Bhagabati-IJCSI-2012
9. Survey on methods for credit card fraud detection-Shilpa H.Taklikar, Prof.R.P.Kulkarni, IJARCSSE-2014
10. Fraud detection of credit card payment system by genetic algorithm K.RamaKalyani,D.UmaDevi-IJSER-2012
11. An effective approach in data mining to reduce redundancies in large databases-IJETAE-Dec2012-Sovers Singh Bisht, Dr.Sanjeev Bansal.