Association Rules for Predicting Customer Lifetime Value in Retail Banking Context Based on RDB-MINER Algorithm

Zhou Xin
Department of Economic Engineering
Kyushu University
Fukuoka, Japan

Abstract—**Data mining methodology has a tremendous contribution for extracting the hidden knowledge and patterns from the existing databases. Traditionally, researchers use basket data to mine association rules of which the basic task is to find the frequent items. For relational databases whose data format is relational data other than basket data, RDB-MINER algorithm was proposed. In this paper, we introduce an improved RDB-MINER algorithm and apply it to mine association rules in retail banking relational databases. When we assess the customer lifetime value, RFM model is adopted. Moreover, we propose a method to find the association rules between customers' attributes and their lifetime value, these patterns are significant for predicting their future value.**

Keywords- Association Rules; Customer Lifetime Value; Relational Database; RDB-MINER Algorithm; RFM Model; Retail Banking.

I. INTRODUCTION

Over the past two decades, there have been numerous researches on mining frequent itemsets from precise data such as databases of market basket transactions. The discovery of association rules is based on frequent itemset mining. Association rules mining[1] is a popular and well researched method for discovering interesting relations between variables in large databases. We can use it to discover regularities between products in large scale transaction data recorded by POS system in supermarkets. For example, the rule {onions, potatoes} $\Rightarrow$ {beef} found in the sales data means if a customer buys onions and potatoes together, he is likely to also buy beef. Such rules can be used as the basis for decisions about marketing activities, e.g., promotional pricing, product placement or new product innovation. Except for the above application in marketing, association rules are employed today in many areas including bioinformatics, finance and economics data analysis, web information mining, etc. Apriori algorithm is the first and best-known algorithm to mine association rules, it uses a breadth-first search strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support.

II. BACKGROUND

A. Definition

Let D = $\{t_1,t_2,\ldots,t_n\}$ be a set of transactions in databases. Let I = $\{ i_1,i_2,\ldots,i_m\}$ be a set of m attributes called items. Each transaction in D has a unique ID, and contains a subset of items in I. An association rule is defined as $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \varnothing$.

B. Concepts

To select useful rules from all the possible rules, support and confidence are used as the two minimum thresholds.

The support sup(X) of an itemset X is the proportion of transactions in the data set which contains X. For example, the itemset {onions, potatoes} has a support of 0.2 if the occurrence frequency is two out of ten in all transactions.

The confidence is defined as $con(X \Rightarrow Y) = sup(X \cup Y)/sup(X)$. For example, if the support for occurrences of transactions where {onions, potatoes} and {beef} both appear is 0.1, the rule {onions, potatoes} $\Rightarrow$ { beef} has a confidence of 0.1/0.2= 0.5.

C. Basket data

Basket data is represented as a set of records where each record contains a unique ID and a set of bought items. Most of market data are represented in basket data format. Table1 shows an example of basket data.

Table 1. Basket data format

Transaction_ID	Bought_Items
T101	{onions, apples, bread}
T102	{egg, potatoes, rice, pork}
T103	{ball pen, notebook, ink}
T105	{onions, potatoes, beef, condiments}

Most of association rules mining algorithms are specialized in market basket data.

D. Relational data

In some specific project environment, we can only obtain the original data which are in relational format. The normalization process does not allow the multi-value attributes to exist in relational databases[2]. Table 2 shows an example of relational data, it is another representation of instance in table 1.

Table 2. Relational data format

Transaction_ID	Onions	Apples	Bread	Egg	Potato	…
T101	1	1	1	0	0	…
T102	0	0	0	1	1	…
T103	0	0	0	0	0	…
T105	1	0	0	0	1	…

Relational data can be converted into basket data, and vice versa. However, it is not easy to convert data format. With the increase in the size of database, it is becoming very difficult to handle large amount of data for computation. So a requirement arises for algorithms which can directly mine association rules in relational datasets without converting relational data format to basket data format when applying such existing algorithms for basket data. RDB-MINER[3][4] algorithm was proposed by Abdallah to solve this problem.

The remainder of this paper is organized as follows: In the next section, we propose an improved RDB-MINER algorithm. We apply it to mine association rules in retail banking context in Section 3. Section 4 is conclusion.

E. Related work

The first algorithm for mining association rules, Apriori algorithm, was proposed in 1994[5]. Since then, numerous related algorithms have been introduced [6-15] which aimed at improving the performance as compared with the Apriori algorithm. Some well known algorithms are Eclat and FP-Growth, but they only do half the job, since they are algorithms for mining frequent itemsets. Another step needs to be done after to generate rules from frequent itemsets found in a database. However, these algorithms are only specialized in mining basket data whose data representation does not conform to the relational data model. Relational database does not allow multi-valued attributes to exist.

Because the cost of changing data format usually can be expensive, especially in big database, RDB-MINER algorithm was proposed for directly mining association rules in relational database.

III. RDB-MINER ALGORITHM AND ITS IMPROVEMENT

There are some background concepts in relational database for introducing the RDB-MINER algorithm. Table 3 is a table in relational database.

Table 3. Employees

EmployeeID	LastName	FirstName	Title	HireDate	…
1	Davolio	Nancy	Sales Manager	1992-5-1	…
2	Fuller	Andrew	Vice President	1992-8-14	…
3	Leverling	Janet	Sales Manager	1992-4-1	…
4	Peacock	Margaret	Sales	1993-5-3	…

Itemset(IS) is defined as a set of items such that no two items belong to the same attribute. For example, in table 3, {Davolio, Sales Manager} is a valid IS, but {Fuller, Leverling, Vice President, 1992-8-14} is not an IS.

- An Itemset Intension(ISI) is a subset of the attributes in a relation. The itemsets that consist of the actual attribute values belonging to these attributes are instantiations of this ISI and named Itemset Extension(ISE). In table 3, {LastName, Title} is an ISI and {Davolio, Sales Manager} is an ISE of this ISI, all the ISEs of the ISI {LastName, Title} are as follows: {Davolio, Sales Manager}, {Fuller, Vice President}, {Leverling, Sales Manager}, {Peacock, Inside Sales Coordinator}.
- Suppose R is a relation with a set A of attributes. Φ(A) is a powerset of A, whose elements are all possible subsets of. Suppose R has two attributes X and Y, then Φ(A)= {{}, {X}, {Y}, {X, Y}}. An equi-cardinality subset is a subset of Φ(A) in which every ISI has the same number of elements(cardinality).

RDB-MINER algorithm is based on standard SQL, it is portable and useful for any type of relational databases. In this paper, we propose an improved RDB-MINER algorithm considering συππορτ and *confidence* into the mining process.

Here is the description of this algorithm.

An Improved RDB-MINER Algorithm

Input
R: a database relation
exclude_set: a subset of the attributes of R
min_supp: the minimum of support
min_conf: the minimum of confidence
target_attribute: the attribute in the right hand statement of association rule

```
0  Begin
1  Compute_N(N, R, exclude_set, target_attribute);

2  Compute_Powerset_exclude_target(Φ(A) , R, exclude_set,
target_attribute);
3  For c=1 to N do
4  Extract_Ec(Ec, Φ(A));
5  For ISI ∈ Ec do
6     Add_to_ISI_set(Φ(A') , R, ISI, min_supp);
7  Compute_N(N, Φ(A'));
8  For c=1 to N do
9     Compute_rule_ISI_set(Φ(A') , target_attribute,
min_supp, min_conf);
10    Add_to_rule_ISE_set(Φ(A'') ,Φ(A') , target_attribute);
11 End
```

In the description of this algorithm, exclude_set is a set of attributes(normally the primary key attributes, target_attribute and some other attributes which are useless for finding the association rules with the targrt attribute) to be excluded from the powerset. Line 1 calls the function Compute_N to compute the number of attributes of R after excluding the exclude_set and the target attribute , the function returns N. In line 2, the function returns a powerset A, of attributes of relation R after excluding the exclude_set and the target attribute . Line 4 extracts all the equi-cardinality subset from Φ(A). The for loop between the line5 and line 6 selects all the ISIs which satisfies the min_supp and add them to a new set $\Phi(A')$. Line 7 returns the number of ISIs in set $\Phi(A')$. In the loop between line 8 and line 10, line 9 selects all the ISIs from $\Phi(A')$ which simultaneously satisfy min_supp and min_conf and can be defined as an association rule with target_attribute , function Compute_rule_ISI_set returns a set $\Phi(A')$ which contains all the satisfying ISIs. In line 10, for each ISI in $\Phi(A')$, we find all the corresponding ISEs and add them to a new set $\Phi(A'')$.

Finally, $\Phi(A'')$ is the result set which contains all the ISEs satisfying the input parameters min_supp and min_conf . From these ISEs, we can summarize some interesting rules for predicting the value of target_attribute .

Here is the description of RDB-MINER algorithm[3].

Algorithm *RDB-MINER*
Input
R: a database relation
exclude_set: a subset of the attributes of R

```
0  Begin
1  Varchar SQL_str (512)
2  Compute_N (N, R, exclude_set)
3  Compute_PowerSet ( P (A) , R, exclude_set) ;
4  For c = 1 to N do
5  Extract_Ec (Ec , P (A) );
/* Ec ⊂ P (A) and each ISI ∈ Ec has a cardinality of c. */
6  For each itemset intension ISI ∈ Ec do
7  Generate_SQL (SQL_Str, ISI, Relation_Name);
8  Execute SQL_Str;
9  SQL_str = "";
10 End
11 End
12 End
```

RDB-MINER algorithm declares a variable called *SQL_str*, which is used to hold the SQL statement to be generated by the algorithm. The improved RDB-MINER algorithm does not use that variable, however, it is also based on SQL. It has more input attributes such as *min_supp* and *min_conf*, it considers support and confidence as screening parameters in the process of algorithm, whereas RDB-MINER algorithm does not consider these parameters, it computes all the ISIs, and then considers *support* and *confidence*.

The improved RDB-MINER algorithm has better performance than the RDB-MINER algorithm. It filters ISIs through min_supp in advance and then find the ISIs which simultaneously satisfy min_supp and min_conf, however, RDB-MINER firstly finds all the ISIs and then filter ISIs through min_supp and min_conf. For example, assume a relation has some attributes and one target attribute, we get the N which is the number of useful attributes for leading to the target attribute. The number of ISIs is 2^N-1. We give a table to compare the two algorithms.

Table 4. Comparison of RDB-MINER algorithm and improved RDB-MINER algorithm

	Time performance	Space performance
RDB-MINER algorithm	compare 2^N-1 times in memory	2^N-1 unit spaces
Improved RDB-MINER algorithm	compare from cardinality 1 to cardinality N, for ISIs in lower cardinality which don't satisfy min_supp, there is no need to compare ISIs which contain the same factors in higher cardinality	no bigger than 2^N-1 unit spaces

From the table, we know that when applying improved RDB-

MINER algorithm, the time of comparisons in memory shall be less than the former one. So it has better performance on time efficiency and space efficiency.

IV. APPLICATION

In this section, we show an application in which the improved RDB-MINER algorithm can be applied.

A. RFM model in retail banking context

Since the increased importance is placed on customer equity in today's business environment, many companies pay more attention to the notion of customer lifetime value and their future profitability to increase market share. The RFM (Recency, Frequency and Monetary) model[16] is used to assess the customer lifetime value. Recency is the last purchase date in a particular period, Frequency is the number of purchases in a particular period, Monetary is the value of purchases in a particular period. In retail banking context[17], suppose that the time window period is 3 months, and the definitions of RFM are described as follows:

- Recency is the interval between the date of last purchase and the first day of last 3 months.
- Frequency is the number of days which occur at least one transaction during last 3 months.
- Monetary is daily average amount of money in all the customer's deposits during last 3 months.

Based on Delphi Experts Grading Method, the relative weights of the RFM variables W_R , W_F and W_M can be obtained. $W_R + W_F + W_M = 1$. Use the normalization method of statistics, we can obtain the normalized R, F , M called NR , NF , NM respectively. Then, we can use formula (1) to compute the customer lifetime value(CLV).

$$CLV = NR * W_R + NF * W_F + NM * W_M \quad (1)$$

Because { NR , NF , NM } $\subset$ [0,1] and $W_R + W_F + W_M = 1$, CLV$\subset$ [0,1]. We can divide the range [0,1] into 10 sections with a sub range of 0.1, and assess the rank of CLV(CLVR) using one of a series of consecutive numbers increased by one 1,2,3,…,10 which represents the CLVR from low level to high level. The results of calculated CLV for different customers or different segments of customers can be used to improve marketing and strategies in the retail banking.

B. Association Rule Mining

Suppose a relation customers in retail bank is shown in table 5, the relative values of W_R , W_F , W_M are 0.08, 0.32 and 0.6 respectively. CLV and CLVR are generated from formula (1). Recency, frequency and monetary are normalized value.

Then, we can use SQL to implement the improved RDB-MINER algorithm to find association rules.

The inputs are:

relation: customers ;
exclude_set : a set of attributes {CID, name, age, sex, city, recency, frequency, monetary, CLV};
min_supp : 0.2;
min_conf : 0.4;
target_attribute : CLVR.

From the function Compute_N , we get N=3. Then we obtain the powerset A of attributes, Φ(A)= {job, education, institution}. In a loop, we get all the ISIs: for cardinality 1, they are {{job},{education},{institution}}; for cardinality 2, they are {{job, education},{job, institution}, {education, institution}}; for cardinality 3, it is only one item, {job, education, institution}. For each ISI, using fuction Add_to_ISI_set , we find the ISEs which satisfies the min_supp. Then, in a loop for itemsets in the ISI set, we find all the ISEs which simultaneously satisfy min_supp and min_conf. Finally, we can summarize all the association rules.

Table **5.** Customers

CID	name	age	sex	job	education	institution	city	recency	frequency	Monetary	CLV	CLVR
1	Li Pin	30	male	manager	Phd	Company	CityA	0.1	0.011	0.4	0.252	3
2	Li Li	22	female	student	master	College	CityB	0.1	0.022	0.001	0.016	1
3	Zhu Qi	45	male	sales	bachelor	Insurance	CityC	0.5	0.011	0.6	0.404	5
…	…	…	…	…	…	…	…	…	…	…	…	…

V. CONCLUSION

This paper provides with several contributions. Firstly, we have proposed an improved RDB-MINER algorithm which considers *min_supp* and *min_conf* in the process of finding ISIs and corresponding ISEs, and give its advantages over RDB-MINER algorithm. Secondly, we adopt this algorithm to find the association rules in customer value management. The obtained application results can help us to classify the customers, predict specific customer's future value and launch some significant commercial activities to promote their lifetime value. The difference between existing RDB-MINER algorithm and the improved one is that the former considers *support* and *confidence* after getting all the ISIs, the latter considers *support* and *confidence* in the process of algorithm, the latter has better performance in time complexity, when the volume of data is very big, the performance becomes important, so choosing a suitable algorithm is very crucial.

REFERENCES

[1] R. Agrawal and R. Srikant: Fast Algorithms for Mining Association Rules. Proceedings of the International Conf. on Very Large Databases (VLDB'94), pp. 487--499. Santiago, Chile , 1994

[2] R. Elmasri, and S. Navathe: Fundamentals of Database Systems, Fifth Edition, Addison-Wesley , 2007

[3] Abdallah Alashqur: RDB-MINER: A SQL-Based Algorithm for Mining True Relational Databases. Journal of Software, vol. 5, no. 9, pp. 998—1005. 2010 Academy Publisher , 2010

[4] Abdallah Alashqur: Using a Lattice Intension Structure to Facilitate User-Guided Association Rule Mining. Computer and Information Science, vol. 5, no. 2, pp. 11--21. Canadian Center of Science and Education, Toronto , 2012

[5] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the International Conf. on Very Large Databases (VLDB'94), 1994, pages 487-499, Santiago, Chile.

[6] J. Wang, J. Han, Y. Lu, and P. Tzvetkov. "TFP: An effcient algorithm for mining top-k frequent closed itemsets." IEEE Trans. Knowledge and Data Engineering, 2005, 17:652-664.

[7] D. Xin, J. Han, X. Yan, and H. Cheng. "Mining compressed frequent-pattern sets." In *Proc. 2005 Int.Conf.* Very Large Data Bases (VLDB'05), pages 709-720, Trondheim, Norway, Aug. 2005.

[8] B. Rozenberg, E. Gudes, "Association rules mining in vertically partitioned databases," Data and Knowledge Engineering 59(2) 2006. Pages: 378–396.

[9] Mohammad-Hossein Nadimi-Shahraki, Norwati Mustapha, Md Nasir Sulaiman, Ali B. Mamat, "A New Algorithm for Discovery Maximal Frequent Itemsets,"International Conference on Data Mining (DMIN) 2008:309-312

[10] B. Lucchese, S. Orlando, R. Perego, "Fast and memory efficient mining of frequent closed itemsets," IEEE Transactions on Knowledge and Data Engineering 18 (1), 2006, 21–36.

[11] H. Moonestinghe, S. Fodeh, P.N. Tan, "Frequent closed itemset mining using prefix graphs with an efficient flowbased pruning strategy," Proceedings of the 6th International Conference on Data Mining, Hong Kong,2006, pp. 426–435.

[12] Lisheng Ma, Yi Qi, "An Efficient Algorithm for Frequent Closed Itemsets Mining," International Conference on Computer Science and Software Engineering (CSSE)2008:259-262

[13] G. Liu, H. Lu, Y. Xu, and J. X. Yu, "Ascending Frequency Ordered Prefix-tree: Efficient Mining of Frequent Patterns," Proc. 2003 Int. Conf. on Database Systems for Advanced Applications (DASFAA03), Kyoto, Japan, March, 2003.

[14] J.Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation" In *Proc. 2000 ACMSIGMOD* Int. Conf. Management of Data (SIGMOD'00), pages 1-12, Dallas, TX, May 2000.

[15] K. Gouda, and M. Zaki, "GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets," Data Mining and Knowledge Discovery: An International Journal, 2005, 11(3): 223-242.

[16] Journal, 2005, 11(3): 223-242.Mahboubeh Khajvand, Kiyana Zolfaghar, Sarah Ashoori, Somayeh Alizadeh: Estimating customer lifetime value based on RFM analysis of customer purchase behavior: Case study. Procedia CS (PROCEDIA), vol. 3, pp. 57--63. Elsevier Ltd, 2011 .

[17] Mahboubeh Khajvand, Mohammad Jafar Tarokh: Estimating customer future value of different customer segments based on adapted RFM model in retail banking context. Procedia CS (PROCEDIA), vol. 3, pp. 1327--1332. Elsevier Ltd, 2011.