Volume : 3 | Issue : 2 | February 2014 • ISSN No 2277 - 8179

Research Paper

Data Mining: Performance Tuning Of Temporal Data Mining Based On Frequent Inter-Transaction Itemsets Discovery

Computer Science

KEYWORDS : Data Mining, Inter –transaction, Association Rule.

Hitesh R Raval Research Scholar, Mewar University Chittorgarh, Rajasthan.

Dr.Vikram Kaushik Director, Manish Institute of Computer Studies,Visnagar.

ABSTRACT

A good number of the studies carried out on mining association rules so far are on mining intra-transaction associations. Traditional association rules are mainly concerned about intra-transaction rules. Mining association rules from transactions occurred at different time series is a difficult task because of high computational complexity, very large database size, multi dimensional attributes. The reason behind all this is the number of potential association rules becomes very large after the boundary of transactions is broken which pose more challenges on efficient processing. Traditional techniques, such as fundamental and technical analysis can provide investors with some tools for managing their stocks and predicting their prices. However, these techniques cannot discover all the possible relations between stocks and thus there is a need for a different approach that will provide a deeper kind of analysis. We proposed framework called FrequentITARM on real datasets of historical end of day price and traded volume. Our approach is efficient preprocessing, pruning techniques to efficiently discover the rule. Proposed work also provides better in-depth study of inter-transaction stock price movement of companies to financial research community, money managers, fund managers, investors etc.

1.Introduction

Database is a technology for data loading, storing, manipulating, querying, sharing and controlling. Leading-edge technology areas, such as data warehousing, web-based applications, object oriented databases, distributed databases, and front end tools are being increasingly utilized by organizations to manage large volume of generated data and information. It is quite useful to understand the trend, behavior of data and information provided by data mining tools & techniques for the Decision Support System.

Data mining is the process of discovering meaningful new correlations, patterns and trends are extract through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques. Data mining is concerned with analyzing large volumes of (often unstructured) data to automatically discover interesting regularities or relationships which in turn lead to better understanding of the underlying processes.

In general, data mining tasks can be classified into two categories:

Predictive mining: In prediction, we are interested in building a model that will predict unknown or future values of attributes of interest, based on known values of some attributes in the database. Classification, Regression and Deviation detection are predictive mining techniques.

Descriptive mining: In knowledge discovery process applications, the description of the data in human-understandable terms is equally if not more important than prediction. Clustering, Association and Sequential mining are some of the descriptive mining techniques.

1.1 Association rule mining

Association rule mining is a technique for discovering unsuspected data dependencies and is one of the best known data mining techniques. It aims to extract interesting correlations, frequent patterns, associations or causal structures among sets of items in the transaction databases or other data repositories.

Definition: Association Rule Mining (ARM) is to find interesting and useful patterns in a transaction database. The database contains transactions which consist of a set of items and a transaction identifier (e.g., a market basket). Association rules are implications of the form X => Y where X and Y are two disjoint subsets of all available items. X is called the antecedent or LHS (left hand side) and Y is called the consequent or RHS (right hand side).

Definition: Support of an association rule is defined as the percentage/fraction of records that contain to the total number of records in the database.

$$\text{Support}(XY) = \frac{\text{Support count of } XY}{\text{Total Number of Transaction in D}}$$

Definition: Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain to the total number of records that contain X, where if the percentage exceeds the threshold of confidence an interesting association rule can be generated.

$$\text{Confidence}(X|Y) = \frac{\text{Support}(XY)}{\text{Support}(X)}$$

2. Problem Statement

Inter-transaction association rule mining is an addition of the intra-transaction association rule mining problem and includes the discovery of relationships, or itemsets, spanning transactions in one or more random dimensions with defined inter-transaction interval. Comparing to classical association rules, difficulty is more because the search space is much bigger as the number of possible rules increases dramatically with both the number of transactions and number of dimensions. The dimensional attribute may be, for example, temporal in the prediction of stock price movement or spatial in a GIS application. Intra-transaction association rule mining is to find frequently appearing patterns for the items time series itself.

R1: If the price of X and Y go up, 60% of time the price of Z goes up (Same day).

R2: If the price of X and Y go up, 60% of time the price of Z goes up the next day.

R3: If the price of X goes up with more than 5% and traded volume goes down with less than 50%, 60% of time the price of Z goes up less than 5% and traded volume goes up between 0 to 50% the next day.

While rule R1 reflects some relationship among the prices with fuzzy temporal window, its role in price prediction is limited. Rule R2 provides better prediction compare to R1 as it provides more meaningful temporal information but still interestingness of the rule is lacking. It is obvious that investors/traders may be more interested in the R3 kind of rules. R3 provides interestingness and accuracy along with temporal information which build

Volume : 3 | Issue : 2 | February 2014 • ISSN No 2277 - 8179

the confidence in decision making. R3 kind of rule is an important form of association rules which is useful but could not be discovered with existing association rule mining framework. This kind of rule associates different itemsets among different transactions, along the axis of day. The contextual information here is day, which is the dimensional attribute. The major advantage of inter-transaction association rules is that besides description they can also facilitate prediction, providing the user with explicit dimensional in the case of prediction, temporal information.

3 RELATED WORK

Several algorithms for mining associations have been proposed in the literature [1], [2], [3], [4], [5], [6].

The association mining task, first introduced in [1], can be stated as follows: Let I be a set of items and D a database of transactions, where each transaction has a unique identifier (tid) and contains a set of items. A set of items is also called an itemset. An itemset with k items is called a k-itemset. The support of an itemset X, denoted σX is the number of transactions in which it occurs as a subset. A k length subset of an itemset is called a k-subset. An itemset is maximal if it is not a subset of any other itemset. An itemset is frequent if its support is more than a user-specified minimum support (min_sup) value.

The Apriori algorithm [2] is the best known previous algorithm and it uses an efficient candidate generation procedure, such that only the frequent itemsets at a level are used to construct candidates at the next level. However, it requires multiple database scans, as many as the longest frequent itemset. The DHP algorithm [3] tries to reduce the number of candidates by collecting approximate counts in the previous level. Like Apriori it requires as many database passes as the longest itemset. The Partition algorithm [4] minimizes I/O by scanning the database only twice. It partitions the database into small chunks which can be handled in memory. In the first pass, it generates the set of all potentially or locally frequent itemsets and, in the second pass, it counts their global support. Partition may enumerate too many false positives in the first pass, i.e., itemsets locally frequent in some partition but not globally frequent. If this local frequent set does not fit in memory, then additional database scans will be required. The DLG [5] algorithm uses a bit-vector peritem, noting the tids where the item occurred. It generates frequent itemsets via logical AND operations on the bitvectors. However, DLG assumes that the bit vectors fit in memory, and thus, scalability could be a problem for databases with millions of transactions. The DIC algorithm [6] dynamically counts candidates of varying length as the database scan progresses, and thus, is able to reduce the number of scans over Apriori. Another way to minimize the I/O overhead is to work with only a small sample of the database. However, the number of algorithms on the inter-transaction mining problem is still few since it is a more challenging problem than intra-transaction mining. We proposed an efficient framework for mining inter-transaction association rules from historical stock market data.

4. FrequentITARM framework

Proposed framework called FrequentITARM for mining inter-transaction association rules for the dataset that contains constant number of items in each transaction. Our approach not only predicts the movement of stock price in either direction with user defined minimum support and confidence value but also predicts the probable variation in stock closing price and traded quantity based on historical data of attributes.

FrequentITARM uses *FP-tree* based algorithm because it requires only two scan of transaction database to construct a tree and uses prefix-tree structure which requires less memory.

For mining inter-transaction association rules, we adopted the following steps:

1) Preprocess the datasets and generate the mega-transactionsfor sliding window w;
2) Frequent itemsets mining on the mega-transactions using *FP-tree* an intra-transaction frequent itemsets mining algorithm;
3) Prune all itemsets which dissatisfy the definition of inter-transaction frequent itemset. 4) Generate inter-transaction association rules from the frequent itemsets.

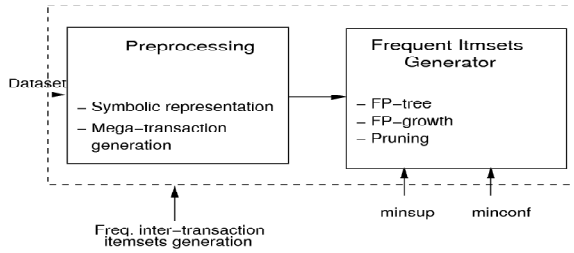

Fig. 1 Preprocessing

Table 1 and Table 2 show example datasets of 10 trading days on closing price and traded volume respectively, where C_x denotes company x and D_y denotes day y.

4.1 Symbolic representation

Time series data are difficult to manipulate, but when they can be treated as symbols (item units) instead of data points, interesting patterns can be discovered and it becomes an easier task to mine.

Change in closing price (C_p) and traded volume (C_v) in inter-day have been discredited and represented as single digit. For example, change in closing price of company 1 (C_1) from day 1 to 2 from table 1 is 305.25 to 302.30. So C_1 has percentage change in closing price C_p is between 0 to -2.5 ranges whose corresponding single digit representation is 3 and same way change in traded volume can be calculated. We have grouped change in closing price (C_p) and traded volume (C_V) and combine them as described below.

Day	*C1*	*C2*	*C3*	*C4*	*C5*
D1	305.25	148.35	158.90	114.05	313.95
D2	302.30	144.90	153.05	116.25	320.15
D3	295.90	140.50	149.95	117.55	328.55
D4	294.05	141.10	149.60	115.65	332.45
D5	291.65	141.95	150.70	117.05	330.75
D6	294.65	140.95	151.90	116.25	332.10
D7	296.40	139.05	152.35	112.95	330.60
D8	289.50	132.60	155.25	116.05	339.25
D9	299.35	135.70	155.40	120.85	344.00
D10	297.75	138.40	155.45	124.85	345.40

Table 1: Example dataset - closing price

Table 1: Example dataset - closing price

$C_p > 2.5\%$ $than$ 1, $0 \le C_p \le 2.5\%$ $than$ 2,
$0 > C_p \ge -2.5\%$ than 3, $C_p < -2.5\%$ than 4,
$C_V > 50\%$ than 1, $0 \le C_V \le 50\%$ than 2,
$0 > C_V \ge -50\%$ than 3, $C_V < -50\%$ than 4.

4.2 Sliding window

An inter-transaction association rule that spans across w intervals is found if an association exists between items that are w intervals apart. Many resources may be required to discover

Table 2: Example dataset - traded volume *(values are in thousands)*

4.2.1 Definition: Sliding window (W) in a transaction database D is a block of w continuous intervals along domain $Dom(D)$, starting from interval d_o such that D contains a transaction at intervald_o. Each interval d_i in W is called a sub window of W denoted as W[j] where j = $d_o -$, j is the sub window number of $d_j$within W.

4.2.3 Mega-transaction generation
Mega-transaction or Extended Transaction [17][16] in a sliding window W is just the set of items in W, each appended with the corresponding sub window number of the interval that contains the item.

4.2.4 Definition: Mega-transaction (M) Let W be sliding window with w intervals and u be the number of literals in ∑. A mega-transaction M contained within W to be

M = {$e_i(j)|e_i \in$ W[j], $1 \le i \le u$, $0 \le j \le w - 1$}.

3a

C1	C2	C3	C4	C5
3	3	4	2	2
3	4	3	2	1
3	2	3	3	2
3	2	2	2	3
2	3	2	3	2
2	3	2	4	3
3	4	2	1	1
1	2	2	1	2
3	2	2	1	2

3b

C1	C2	C3	C4	C5
4	4	1	1	1
2	3	2	2	2
2	2	4	4	1
2	2	2	2	2
4	4	2	3	2
2	2	3	4	2
2	1	2	1	1
2	3	3	2	4
4	4	3	3	2

Table 3: Symbolic representation of change in closing price and traded volume

11: 00	12: 01	13: 02	14: 03
21: 04	22: 05	23: 06	24: 07
31: 08	32: 09	33: 10	34: 11
41: 12	42: 13	43: 14	44: 15

Table 4 Mapping table

To distinguish the items in a mega-transaction from the items in a traditional transaction given in table 1 and Table 2, we called the items in a mega-transaction, **extended-items**. We denote the set of all possible extended items as I'. Given ∑ = {$e1, e2, \ldots, eu$ } be a set of u distinct items and W be a sliding window with w intervals, we have: I' = $e_1(0)$,. . . , $e_1(w-1)$, $e_2(0)$,. . . ,$e_2(w-1)$,. . . , $e_w(0)$,....,$e_w(w-1)$. Table 5 shows 1-dimensional transaction dataset T with its dimensional attributes, trading day. In dataset, 9 such transactions are shown. Assuming that the length of sliding window w is 3, we will have 7 sliding windows. Table 6 shows generated extended-items.

Day	C_1	C_2	C_3	C_4	C_5
D_2^1	11	11	12	04	04
D_3^2	09	14	09	05	01
D_4^3	09	05	11	11	04
D_5^4	09	05	05	05	09
D_6^5	07	11	05	10	05
D_7^6	05	09	06	15	09
D_8^7	09	12	05	00	00
D_9^8	01	06	06	01	07
D_{10}^9	11	07	06	02	05

Table 5. Modified symbols

1111 1211 1312 1404 1504 ***2109*** *2214 2309 2405 2502* ***3109*** *3205 3311 3411 3504*

1109 *1214 1309 1405 1502* ***2109*** *2205 2311 2411 2504* ***3109*** *3205* ***3305*** *3405 3509*

1109 *1205 1311 1411 1504* ***2109*** *2205 2305 2405 2509 3107 3211* ***3305*** *3410 3505*

1109 *1205* ***1305*** *1405 1509 2107 2211 2305 2410 2505 3105 3209* ***3306*** *3415 3509*

1107 1211 ***1305*** *1410 1505 3105 2209 2306 2415 2509* ***3109*** *3212* ***3305*** *3400 3500*

1105 1209 1306 1415 1509 ***2109*** *2212 2305 2400 2500 3101 3206* ***3306*** *3401 3507*

1109 *1212* ***1305*** *1400 1500 2101 2206 2306 2401 2507 3111 3207* ***3306*** *3402 3505*

Table 6: Mega-transactions retrieved from using a sliding window (w = 3)

4.3 Tree construction
Here use an existing data structure called *FP-tree*. *FP-tree* is an extended prefix-tree structure storing crucial, quantitative information about frequent patterns. To ensure that the tree structure is compact and informative, only frequent length-1 items will have nodes in the tree, and the tree nodes are arranged in such a way that more frequently occurring nodes will have better chances of node sharing than less frequently occurring ones [16].

2109 3109
1109 2109 3109 3305
1109 2109 3305
1109 1305 3306
1305 3109 3305
2109 3306
1109 1305 3306

Table 7: Frequent mega-transaction itemsets in order of their occurrences

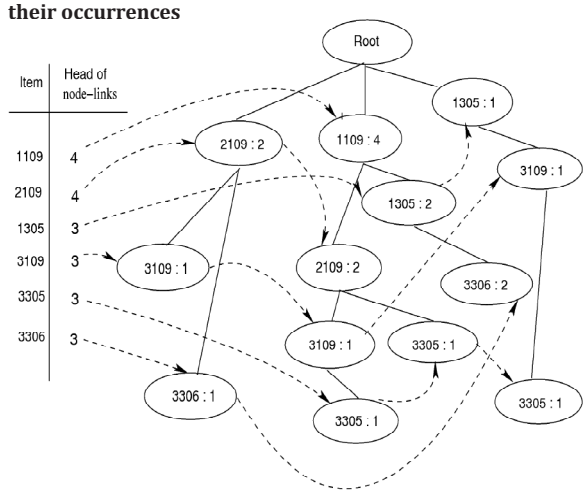

Figure 2: The FP-tree for the example dataset with w = 3 and *minsup = 3*

Figure 2 shows the *FP-tree* for the frequent mega-transaction itemsets shown in table 7. The head elements of the item lists are shown to the left of the vertical bar, the prefix tree to the right of it. The root of a tree is created and labeled with *"null"*. First digit in 4 digits number before colon represents the dimensional offset, in this case the item occurrence day, while colon delineated numbers depict the node frequency. The *FP-tree* is constructed as follows by scanning the transaction database D the second time.

Volume : 3 | Issue : 2 | February 2014 • ISSN No 2277 - 8179

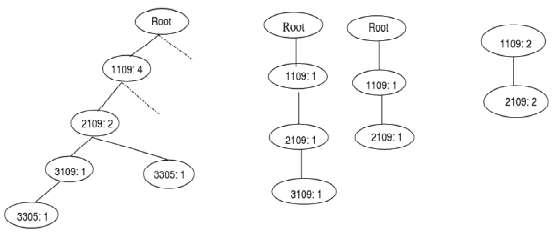

Figure 3: Conditional FP-tree for item *3305*

5 Performance Analysis

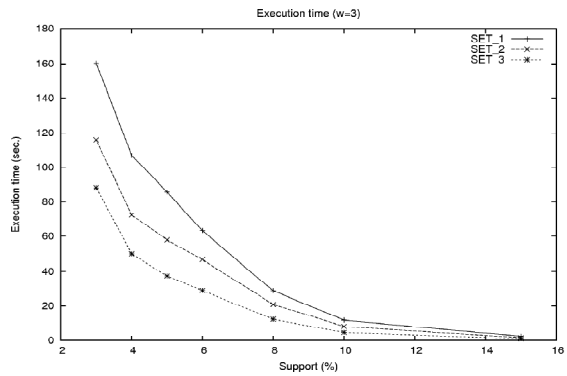

Figure 4: Execution time (w = 3)

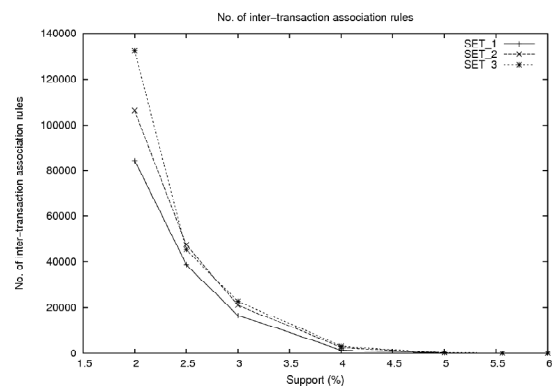

Figure 5: No. of inter-transaction rules

Figure 4 shows the execution time requires to mine inter-transactions association rules with FrequentITARM framework when support threshold is lowered from 15% to 2%. It concludes that an order of magnitude difference in the execution times exists due to large number of discovered rules at the lower support values as shown in figure 5. Figure 5 also admits the basic nature of inter-transaction rule mining i.e. number of rules is increasing exponentially once the minimum support barrier is broken

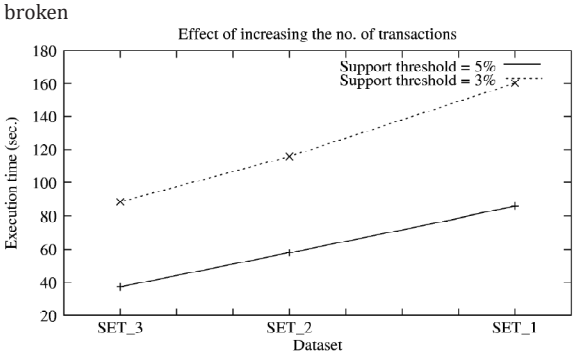

Figure 6: Effect of increasing the no. of transactions

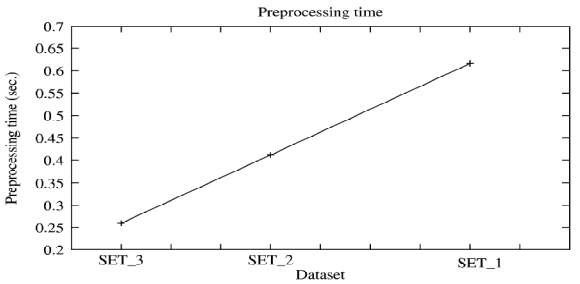

Figure 7: Preprocessing time

We investigated how FrequentITARM performs when the number of transactions in the database increases. We vary the number of transactions in dataset from 1000 to 2400 with support threshold = 3%, 5% and sliding window size w = 3. Figure 4.3 shows the effect on execution time when number of transactions in database increases. As number of transactions increases, more number of frequent itemsets generated which requires more time for execution. It is clear from figure 6 that the execution time increases with the number of transactions in database for given support value.

Preprocessing time is directly proportional to the number of transactions in database. Figure 7 depicts that the preprocessing time increases almost linearly with increase in number of transactions and has nothing to do with itemsets discovery.

Conclusion

Company stock closing price, traded volume, business sector, earning numbers, Price Earnings ratio, rumors etc. are some of the factors that influence the stock price. Obviously, all these factors cannot be easily modeled and embedded, since some of them are related with human psychology. We have taken two most influencing factors - closing price and traded volume for the work. Experimental results show the normal phenomena of National Stock Exchange (NSE), India, as company stock price is following movement of higher index based companies for reasonable higher support and confidence. Enhanced mechanism that provides better trade-off between main memory and interactive inter-transaction rule mining as current work is not suitable for interactive inter-transaction rule mining because *FP-tree* does not support such property. We have considered set boundary for symbolic representation, applying fuzzy logic can provide more accuracy with higher computation complexity. On financial research perspective, test proposed framework on dataset combining leading companies across countries where stock market movement is co-related such as United States (DOW JONES & NASDAQ), China (SSEC), Hong Kong (HANG SENG), Japan (NIKKEI), Taiwan (TAIEX) for wider acceptance.

REFERENCE

R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases,"ACM SIGMOD Conf. Management of Data, May 1993. | | [2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri Verkamo, "Fast Discovery of Association Rules," Advances in Knowledge Discovery and Data Mining, U. Fayyad and et al.,eds., pp. 307 to 328, Menlo Park, Calif.: AAAI Press, 1996. | | [3] J.S. Park, M. Chen, and P.S. Yu, "An Effective Hash Based Algorithm for Mining Association Rules," ACM SIGMOD Int'l Conf. Management of Data, May 2011. | | [4] A. Savasere, E. Omiecinski, and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases," Proc. 1st Very Large Data Bases Conf., 2013. | | [5] S.-J. Yen and A.L.P. Chen, "An Efficient Approach to Discovering Knowledge from Large Databases," Fourth Int'l Conf. Parallel and Distributed Information Systems, Dec. 2012. | | | [6] S. Brin, R. Motwani, J. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data," ACM SIGMOD Conf. Management of Data, May 20012. |