PATTERN DISCOVERY FOR MULTIPLE DATA SOURCES BASED ON ITEM RANK

Arti Deshpande[1], Anjali Mahajan[2] and A Thomas[1]

[1]Department of CSE, G. H. Raisoni College of Engineering, Nagpur, Maharashtra, India
[2]Department of IT, Government Polytechnic, Nagpur, Maharashtra, India

ABSTRACT

Retail company's data may be geographically spread in different locations due to huge amount of data and rapid growth in transactions. But for decision making, knowledge workers need integrated data of all sites. Therefore the main challenge is to get generalized patterns or knowledge from the transactional data which is spread at various locations. Transporting data from those locations to server site increases the cost of transportation of data and at the same time finding patterns from huge data on the server increases the time and space complexity. Thus multi-database mining plays a vital role to extract knowledge from different data sources. Thus the technique proposed finds the patterns on various sites and instead of transporting the data, only the patterns from various locations get transported to the server to find final deliverable pattern. The technique uses the ranking algorithm to rank the items based on their profit, date of expiry and stock available at each location. Then association rule mining (ARM) is used to extract patterns based on ranking of items. Finally all the patterns discovered from various locations are merged using pattern merger algorithm. Proposed algorithm is implemented and experimental results are taken for both classical association rule mining on integrated data and for datasets at various sources. Finally all patterns are combined to discover actionable patterns using pattern merger algorithm given in section V.

KEYWORDS

Association Rule Mining, Pattern Discovery, Data Mining, Ranking

1. INTRODUCTION

The conventional methods for mining numerous databases is to integrate the databases, then apply association mining to discover patterns. It is frequently difficult to coordinate heterogeneous databases or to move one entire database to another site, reason for both efficiency and protection concerns. Along these lines in multi-database mining we require efficient approaches that can deliver great mining results with low cost of database correspondence. Association Rule Mining discovers application in business sector to get frequently purchased items, unseen patterns, associated events etc. The business sector experts would be keen on recognizing habitually bought things, so that profitable deals can be given to the customers.

Retailers and supermarket maintains large databases which contain valuable information related to customers and items purchased by them. Customer buying patterns can be discovered from the large volumes of data which is scattered at various locations. Transactional data of each outlet of supermarket is local to the location. So every outlet maintains the local server for day to day transactions. If the supermarket has outlet at 100 locations, all 100 local servers are maintained at each site. But when organization decides any policy or campaign or promotions for any product, it is same for all 100 locations. Demographic data and behavioral patterns vary from location to location. So transactional data of all locations play vital role to discover final patterns. When

DOI: 10.5121/ijdkp.2017.7104

organization decides any offer on particular product, it basically checks for most frequently purchased or profitable or about to expire date product. For that, all transactional data at various locations get considered. If such data is transported to main server of organization and then association rule mining is applied to discover frequent patterns, then the transportation cost and space required get increased. Proposed technique suggest that instead of transporting data to the server, patterns can be discovered at local site and then those patterns are transported to the main server to discover final patters. Ranking algorithm is also used to give rank to each items available in transaction list.

2. LITERATURE REVIEW

ARM was first introduced by Agrawal et al. 1993[1]. ARM aims to extract interesting patterns from large dataset. Based on minimum support and confidence, frequent items are discovered and association rules are generated. Market basket analysis is a modelling technique which is also called as affinity analysis, it helps identifying which items are likely to be purchased together. The market-basket problem assumes we have some large number of items, e.g., “bread”, “milk.”, etc. Customers buy the subset of items as per their need and marketer gets the information that “which things customers have taken together”. So the marketers can use this information to sale the items together or give some promotional campaign on those items.

For example: If someone buys a packet of milk also tends to buy bread at the same time and it is represented as Milk=>Bread. Market basket analysis algorithms are straightforward; difficulties arise mainly in dealing with large amounts of transactional data, where after applying algorithm it may give rise to large number of rules which may be trivial in nature. The problem of large volume of trivial results can be overcome with the help of differential market basket analysis which enables in finding interesting results and eliminates the large volume. Using differential analysis it is possible to compare results between various stores, between customers in various demographic groups. Support, confidence and lift are strategically measures that control the process of association rule.

Support: The support of the rule, that is, the relative frequency of transactions that contain $x \wedge y$
[2] $\text{support}(x \rightarrow y) = \text{support}(x+y)$
Confidence: The confidence of the rule defines how many number of times transactions contains x also contains y.
$\text{confidence}(x \rightarrow y) = \text{support}(x+y) / \text{support}(x)$
Lift: The lift value of the rule is the additional interestingness measures on the rules. These measures can then be used to either rank the rules by importance (or present a sorted list to the user) or as an additional pruning criterion.
$\text{lift}(x \rightarrow y) = \text{confidence}(x \rightarrow y) / \text{support}(y)$

Jia Hu et al. [3] presented a conceptual model with dynamic multi-level workflows corresponding to mining-grid centric multi-layer grid architecture which govern enterprise application integration. The technique helps to mine multiple data sources for customer segmentation based on their online behaviour. One to one marketing and personalization is possible due to the model given.

Mayssam Sayyadian [4] described the prediction model assuming that all the needed data to build an accurate prediction model resides in a single database. Tuples from various autonomous and heterogeneous databases are combined to collect all information which is required to build suitable classification model. HeteroClass framework is proposed for effective classification based on heterogeneous databases. They have also suggested the method to combine schema matching and structure discovery techniques.

The Author [5] proposed a specialised technique to improve mining multiple databases using local pattern analysis. A new generalized technique is given for mining multiple large databases which improves the quality of synthesized global patterns considerably. In [7] the method proposed synthesize high frequency rules from different data sources by calculating weights of different sources based on their transaction population. These weights can be allocated based on turnover or quantity of items sold or profit gain. They have also introduced global confidence to get final global patterns.

Multi-Database Mining is given by Shichao Zhang et. al. [8].The author have mainly focused on mono and multi database mining. They introduced various types of patterns in multi-database like (1) local patterns (2) high-vote patterns (3) exceptional patterns (4) suggested patterns. They suggested the classification of multi database and the problems related to make clusters of those.

3. Proposed Architecture

The transactional data is spread over the chain of retail store at various locations where the store is located. The transactional database consists of two sections; first section includes the “Sales” binary data per transaction depending on the items purchased by the customers. The second section contains the “Stock” data per item which includes the total stock available and the number of days left for the item to expire at each site. The frequent itemset generation algorithm Apriori [6] is applied to generate local patterns at each site. Instead of transporting the local transactional data, only the patterns generated at local site are transported to server site. To get top n items, Simple Additive Weighting (SAW) [9] method is applied on server site based on the integrated patterns from various locations. Finally collaborative Association Rule Mining (CARM) is used to generate deliverable patterns based on top n selected items. The proposed architecture diagram is given in fig 1.

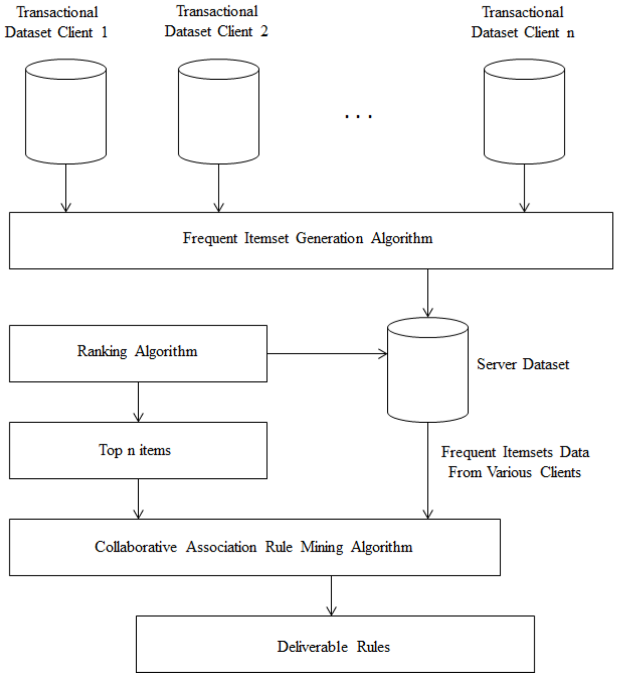
Transactional Dataset Client 1
Transactional Dataset Client 2
Transactional Dataset Client n
. . .
Frequent Itemset Generation Algorithm
Ranking Algorithm
Server Dataset
Top n items
Frequent Itemsets Data From Various Clients
Collaborative Association Rule Mining Algorithm
Deliverable Rules

Figure1. Architecture of proposed system

3.1 Frequent Itemset Generation

For each local site same minimum support is set and Apriori [1] algorithm is applied on transactional data. Two support values are considered to get frequent item set. Idea is to find the patterns with maximum purchased item with minimum purchased items. So two support values are set as (1) Lower Support (2) Upper Support. Thus only the items less than or equal to lower support and the items more than or equal to upper support are considered to generate frequent item sets. So minimum sold and maximum sold items are combined together. Instead of transporting transactional data, frequent item set along with stock and expiry days of each item is transported to server site. On server site ranking method SAW [9] is applied to get the rank for each item. SAW uses the different criteria like total number of stock available, Number of days to expire and Profit for each item to give the rank to each item. So only the top n items are selected based on the rank of items. Finally frequent itemsets from each site are selected which are having only the items from those top n items.

3.2 Proposed Algorithm

Input : transactional Dataset D_1, D_2,…….., D_k for k sites, Lower Support, Upper Support
Algorithm:

```
1.  For each site from 1 to k
        a.  Generate frequent ItemSets Using Apriori by considering Lower Support
            and Upper Support
        b.  Transport Frequent itemets with support to Server Site
        c.  Transport Stock Available, Days to expire of each item and profit for the
            item to Server Site
End For
2.  Merge All Frequent items generated from step 1.
3.  Apply SAW on combined table of all items
        a.  Normalize the data using min-max normalization
        b.  Assign weights to Stock Available, Days to expire of each item and
            profit for the item
        c.  Assign Rank to each item
4.  Get TOP n items
5.  Filter Out frequent itemsets from each site which contains only items from n
    items
6.  Merge all filtered frequent itemsets
```

Output: Deliverable Itemsets generated from step 6

4. EXPERIMENTS AND RESULTS

For experimental purpose two sites S1 and S2 are considered. Synthetic data for 20 items are generated using SQL Data Generator [10] for both the sites. Algorithms are implemented using Java 7 and database used is MySQL .By considering Upper Support 40% and Lower Support 2%, frequent itemsets are generated using Apriori for both site S1 and S2 as shown in Table 1 and 2 respectively.

Number	Elements	Support
1	[Juice]	0.53
2	[Biscuits]	0.71
3	[Bread]	0.54
4	[Butter]	0.41
5	[Biscuits, Bread]	0.41

Table 1. Frequent Itemsets at Source 1 (S1)

Number	Elements	Support
1	[Biscuits]	0.72
2	[Bread]	0.59
3	[Milk]	0.41
4	[Oats]	0.41
5	[Biscuits, Bread]	0.48

Table 2. Frequent Itemsets at Source 2 (S2)

Stock available, Expiry in days and Profit on each item is also transported from site S1 and S2 to server site as shown in Table 3 and 4.

Item	Expiry in Days	Stock	Cost Price	Selling Price	Stock Value	Stock Sale Value	Profit per Item	Profit on Full Sale
Juice	8	25	80.0	99.0	2000.0	2475.0	19.0	475.0
Tea	15	62	42.0	58.0	2604.0	3596.0	16.0	992.0
Biscuits	6	48	20.0	32.0	960.0	1536.0	12.0	576.0
Chicken	12	50	160.0	190.0	8000.0	9500.0	30.0	1500.0
Bread	5	35	20.0	30.0	700.0	1050.0	10.0	350.0

Table 3. Data Fetched from Site S1

Item	Expiry in Days	Stock	Cost Price	Selling Price	Stock Value	Stock Sale Value	Profit per Item	Profit on Full Sale
Juice	10	50	80.0	99.0	4000.0	4950.0	19.0	950.0
Tea	12	90	42.0	58.0	3780.0	5220.0	16.0	1440.0
Biscuits	7	40	20.0	32.0	800.0	1280.0	12.0	480.0
Chicken	6	40	160.0	190.0	6400.0	7600.0	30.0	1200.0
Bread	4	50	20.0	30.0	1000.0	1500.0	10.0	500.0

Table 4. Data Fetched from Site S2

Data from table 3 and 4 are integrated and normalized before applying the ranking method SAW. Rank for each item is calculated using SAW based on Stock available, Expiry in days and Profit criteria. Top 6 items are shown below in Table 5.

Item	Value	Rank
Chicken	0.83	1
Milk	0.54	2
Jam	0.53	3
Bread	0.46	4
Juice	0.4	5
Biscuits	0.39	6

Table 5. Top 6 items with their value and Rank

Only the itemsets consist of items from table 5 are selected from table 1 and 2 which gives the frequent itemset list from source S1 and S2. Table 6 shows the final deliverable patters.

Number	Elements	Source
1	[Juice]	S1
2	[Biscuits]	S1, S2
3	[Bread]	S1, S2
4	[Milk]	S2
5	[Biscuits, Bread]	S1, S2

Table 6. Combined Frequent Itemsets from Sources S1 and S2

So from the above results , item [Milk] is missing at Site S1 but still considered as final deliverable, similarly [Juice] is missing at Site S2.

In second scenario, all transactional data is transported to server site from site S1 and S2, association rule mining is applied on integrated data. Frequent Itemsets generated are given in Table 7.

Number	Item	Support
1	[Biscuits]	0.72
2	[Bread]	0.56
3	[Biscuits, Bread]	0.44

Table 7. Frequent ItemSets after combining Transactional Data from site S1 and S2

After comparing the results from table 6 and 7, it is observed that frequent items [Juice] and [Milk] are missing in table 7 though they are in top 6 items list. If the expiry date, stock and profit are considered, these items should move fast for selling. So retail expert can take decisions to give some discount or offer on these two items.

5. CONCLUSION

The technique proposed is reducing the cost of transportation of data from client side to server site. As the association rule mining is applied at local site, space required and time to execute algorithm is reduced drastically. This helps in finding the various patterns based on geographical location as people from different places have generally different buying pattern habits. Once all patterns get integrated from various sites, we are able to get some common patterns. So ranking method helps to sort the most important items which are supposed to move fast based on their expiry date, stock and profit etc. So filtering those items from frequent itemlist helps retailer to clear those stock. We found that integrated transactional data result missing some patterns which are crucial. This algorithm further can be extended by considering quantity purchased for each item and quantity based association rule mining can be used to get patterns. We are working on quantitative association rule mining for multiple sources.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases,” in ACM SIGMOD Record, vol. 22, no. 2. ACM, 1993, pp. 207-216.

[2] Deshpande, Arti, and Anjali Mahajan. "Domain Driven Multi-Feature Combined Mining for retail dataset." InInternational Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958.

[3] Hu, Jia, and Ning Zhong. "Organizing multiple data sources for developing intelligent e-business portals." Data Mining and Knowledge Discovery 12, no. 2-3 (2006): 127-150.

[4] Sayyadian, Mayssam. "HeteroClass: a framework for effective classification from heterogeneous databases." CS512 Project Report, University of Wisconsin, Madison (2006).

[5] Adhikari, Animesh, Pralhad Ramachandrarao, Bhanu Prasad, and Jhimli Adhikari. "Mining Multiple Large Data Sources." Int. Arab J. Inf. Technol. 7, no. 3 (2010): 241-249.

[6] Borgelt, Christian. "Efficient implementations of apriori and eclat." In FIMI’03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations. 2003.

[7] Ramkumar, Thirunavukkarasu, and Rengaramanujam Srinivasan. "Modified algorithms for synthesizing high-frequency rules from different data sources."Knowledge and information systems 17, no. 3 (2008): 313-334.

[8] Zhang, Shichao, Xindong Wu, and Chengqi Zhang. "Multi-database mining."IEEE Computational Intelligence Bulletin 2, no. 1 (2003): 5-13.

[9] Savitha, K., and C. C handrasekar. "Trusted network selection using SAW and TOPSIS algorithms for heterogeneous wireless networks." arXiv preprint arXiv:1108.0141 (2011).

[10] http://www.red-gate.com/products/sql-development/sql-data-generator/

Authors

Arti Deshpande is an Assistant Professor in the Department of Computer Engineering in Thadomal Shahani Engineering College, Mumbai, India. She received Master of computer engineering from Mumbai University, Maharashtra, India. Currently, she is a PhD student at the Department of Computer Science and Engineering in G. H. Raisoni College of Engineering , Nagpur, Maharashtra, India.

Dr. A. R. Mahajan is working as Head of Information Technology Department, GP, Nagpur. She is BE and ME in Computer Science and Engineering She has completed her PhD in Computer Science and Engineering. She has presented thirty four papers in International Journal and one paper in national Journal She has published Forty three papers in International conferences and five papers in national conferences .She has more than twenty years of experience. Her area of specialization is compiler optimization, Artificial Intelligence, Parallel algorithms. She is a member of IEEE, ISTE, and CSI.

A.Thomas received M.Tech(Computer Science & Engineeirng) in 2013. She is Head of Computer Science Department at G.H.Raisoni College of Engineering, Nagpur. She has completed M.Phil (Computer Science) in the year 2011. Her area of specialization is soft Computing and Data mining