An Efficient Mining Approach of Frequent Data Item Sets on Large Uncertain Databases

Isse Hassan Sheikh Nur[1], Melih Kırlıdoğ[2]

[1,2]*Dept. of Computer Engineering, Marmara University*
Göztepe 34722 Istanbul, Turkey

***Abstract* —Mining frequent items from large uncertain database is a crucial issue, according to the accuracy, performance and computational cost, where we need the frequent itemset is ascertained efficiently and accurately with low computational cost and high performance in detecting probabilistic frequent item (PFI), so all of these factors are the required or recommended in a large uncertain database to extract the frequent items efficiently and accurately. In uncertain database the support of an item occurs randomly instead of fixed variable. We will use a model based algorithm and dynamic algorithm for mining and generating candidate itemsets for frequent itemsets in large uncertain data. Our goal is a better performance based on our dataset.**

***Keywords*— Frequent itemsets, Probabilistic Frequent Item.**

I. Introduction

Data mining is the method of extracting of hidden predictive information from large databases; it is a powerful new technology with good benefit to assist companies concentrate on the foremost necessary information in their databases and warehouses. Data processing tools can also predict future pattern and behaviors, by permitting businesses to form proactive, knowledge-driven decisions. Data mining tools could answer business related queries that may take long time to solve traditionally. They search databases for hidden patterns by finding a predictive and related information that skillful or experts might miss as a result of it lies outside their anticipation. Most of the companies already collected and refined large amount of knowledge and information.

Data mining techniques will be enforced quickly on existing software package and hardware platforms to boost the value of existing resource information, and may be integrated with new merchandise and systems as they're brought on-line. The data mining techniques are the result of a long process of analysis and merchandise development. This evolution began once business information was first kept on computers, continuing with improvements in accessing data recently, generated technologies that permit users to navigate through their data in real time [1].

II. Dynamic Programing Algorithm

Dynamic programming is a design technique similar to divide and conquer. A dynamic-programming algorithm tackles each sub problem once and then saves its answer in a table, by keeping away from the work of recomputing the answer each time the sub issue is experienced.

A. Generating Candidate 2 itemset based on dynamic Programming algorithm

In this section we will illustrate an example that shows generating candidate 2-itemsets based on dynamic programing and support count of these candidate 2-itemsets will be shown as well.

Transaction ID	Item Name
105,656,689,**806,**304,**1625,**114,352 ,416,**726,**752,**1642,**1649,**1549,**1595 ,**1819,**549,**601,**769,9,**214,**272,**1200**	Trunks and Cases
115,**214,**304,680,**726,**805,**806,**982, **1549,**1566,**1625,**997,**1200,**1267,**18 19,**1851, 468,**601,**1173,**1642**	Tropical Fruits

Table1 Generating candidate 2 itemset based on DP

B. Support Calculation of Candidate 2 itemset

We extracted two candidte itemsets from our database we got 2000 data items and 9-itemsets in total, in this example we have chosen these two itemset {**Trunks and Cases**, **Tropical Fruits** }, with their transactions ids as shown in table 1. To count the support, instead of whole database for each itemset, we find longest common subsequence and length of transaction id's of these candidate 2-itemsets, by using dynamic programming approach which faster than traditional approach, an advantage of this approach is in each iteration database filtering and reduces. [2].

In the table above we have two candidates itemset namely **Trunks and Cases**, **Tropical Fruits**, the longest common sequence of these candidate 2 itemset is LCS**(Trunks and Cases**, **Tropical Fruits)** = **{(**806, 1625, 726, 1642, 1549, 1819, 601, 214, 1200**),(** 806, 1625, 726, 1642, 1549, 1819, 601, 214, 1200**)}**, now the common sequence of these two candidate itemset is **(**806, 1625, 726, 1642, 1549, 1819, 601, 214, 1200**)**, so the support of these candidate 2-itemset is 9.

C. *Generating Candidate 3 itemset based on dynamic Programming algorithm*

Now we generate the candidate 3-itemset by using the dynamic programming algorithm, generating candidate 3-itemset we first find longest common subsequence and length of transaction ids of each item of these candidate 3-itemset, the below illustrated table shows all sequence of all id transactions of candidate 3-itemset.

As shown in the table 2 we have three candidates itemset which are extracted from our data, the candidate items are {Used Clothing**,** Paper Containers**,** Wheat**},** the longest common sequence of these candidate itemset is LCS**(**Used Clothing, Paper Containers ,Wheat**)** = **{(**1288, 1633, 1227, 1288, 1633, 1227, 1288, 1633, 1288**)}**, now the common sequence of these three candidate itemset is **(**1288, 1633, 1227**)**, so the support of these candidate 3-itemset is 3.

Transaction ID	Item Name
1092,1419,**1633,**198,1589,**1227,**1809,943,**1288,**1353,1862,605	UsedClothing,Paper Containers
371,**1633,**1862,**1288,**1092,**1227,**605,943,198,1589,1353,1419	PaperContainers,Used Clothing
1633,1227,457,1057,1192,1188,1467,1581,1661,**1288**	Used Clothing, Wheat
	PaperContainers,Wheat

Table 2 Generating candidate 3- itemset based on DP

III. Mining PFI based on Apriori algorithm

The most common method for discovering frequent items is the Apriori algorithm.

Apriori Algorithm

Let D be the market-basket database, where each row contains T Transactions. Transactions tagged with unique identifier T_{id}. Now let I be the item set $\{I_1, I_2, I_3 \ldots I_n\}$. If an item set contain k-item then it called K-itemset, and if all subset of K-itemset satisfies the minimum support count then it's called L_k frequent itemset or large itemset. This algorithm need to perform two basic steps which are (1) Join, self-join with previous frequent L_{k-1} itemset and create new candidate C_{k+1} itemset. (2) Prune, filter from the current candidate itemset whose subset is not frequent in previous step. Below step explain the working of *Apriori* algorithm [3].

1. Assume that minimum support count and minimum confidence are given as min-sup and min-conf respectively.
2. Scan the entire database and find out candidate 1-itemset C_1 along with occurrence count. That is number of times each item appeared in database.
3. From C_1 eliminate those items which count is not satisfying min_sup threshold. Remaining 1-items in C_1 which called L_1.
4. $L_1 \infty L_1$, and create new C_2, again scan the database and calculate number of times candidate 2-itemset appeared in database.
5. Apply the pruning in C_2 and we get the L_2.
6. As this way iteratively step 2 to 5 is carried out until the C_K is null [3].

The following figure shows us the frequent itemsets that has been extracted from our database during candidate generation process.

FREQUENT ITEMSET

Item1	Item2	Item3	Support	TID
Raw Sugar	Special Pharmac...	Paper Containers	3	325,1818,1782,
Raw Sugar	Special Pharmac...	Ethylene Polymers	3	1013,1635,650,
Raw Sugar	Special Pharmac...	Cellulose Fibers ...	3	325,1270,1013,
Raw Sugar	Special Pharmac...	Buses	3	650,1782,1818,
Raw Sugar	Special Pharmac...	Laboratory Glass...	3	974,1007,963
Raw Sugar	Onions	Iron Cloth	3	1899,1551,1276
Raw Sugar	Onions	Cars	3	974,1551,1276,
Raw Sugar	Paper Containers	Wheat	3	717,1818,1832,
Raw Sugar	Paper Containers	Iron Chains	3	1033,1818,67,
Raw Sugar	Paper Containers	Knit Womens Suits	3	1160,1353,1033,
Raw Sugar	Paper Containers	Heavy Pure Wov...	3	1353,67,325
Raw Sugar	Paper Containers	Combustion Engi	3	1160,1499,1261

Fig. 1 Generated Frequent Itemsets

Frequent Items Based on Dynamic algorithm

A dynamic programming algorithm is used to extract frequent itemsets in order to reduce the execution of the dataset and to extract the PFI. Fig. 2 shows us the extracted frequent itemse.

FREQUENT ITEMSET BASED ON DP

Item1	Item2	Item3	Support	TID
Raw Sugar	Office Machine Parts	Onions	1	604,
Raw Sugar	Office Machine Parts	Confectionery Sugar	1	949,
Raw Sugar	Office Machine Parts	Iron Cloth	1	551,
Raw Sugar	Office Machine Parts	Excavation Machinery	2	893,949,
Raw Sugar	Office Machine Parts	Stranded Iron Wire	2	1060,32,
Raw Sugar	Office Machine Parts	Mattresses	1	604,
Raw Sugar	Office Machine Parts	Prefabricated Buildings	1	949,
Raw Sugar	Office Machine Parts	Light Mixed Woven ...	1	949,
Raw Sugar	Office Machine Parts	Tractors	2	551,834,
Raw Sugar	Office Machine Parts	Edible Preparations	1	1060,
Raw Sugar	Office Machine Parts	Glass Articles	1	1060,
Raw Sugar	Office Machine Parts	Synthetic Filament	1	1060

Fig. 2 Frequent Itemset Based on DP

Frequent Items Based Model Based Algorithm

The model based algorithm has been used to extract threshold based probabilistic frequent itemset in order to reduce the execution of the dataset and to extract the PFI. The minimal support count is in our dataset is 3; the Fig. 3, shows us the extracted itemset from our database.

FREQUENT ITEMSET BASED ON MB

Item1	Item2	Item3	Support	TID
Raw Sugar	Special Pharmac...	Paper Containers	3	325,1818,1782,
Raw Sugar	Special Pharmac...	Ethylene Polymers	3	1013,1635,650,
Raw Sugar	Special Pharmac...	Cellulose Fibers ...	3	325,1270,1013,
Raw Sugar	Special Pharmac...	Buses	3	650,1782,1818,
Raw Sugar	Special Pharmac...	Laboratory Glass...	3	974,1007,963
Raw Sugar	Onions	Iron Cloth	3	1899,1551,1276
Raw Sugar	Onions	Cars	3	974,1551,1276,
Raw Sugar	Paper Containers	Wheat	3	717,1818,1832,
Raw Sugar	Paper Containers	Iron Chains	3	1033,1818,67,
Raw Sugar	Paper Containers	Knit Womens Suits	3	1160,1353,1033,
Raw Sugar	Paper Containers	Heavy Pure Wov...	3	1353,67,325
Raw Sugar	Paper Containers	Combustion Engi...	3	1160,1499,1261

Fig. 3 Frequent Itemset Based on MB

Calculating the Recall and Precision values

Precision is the fraction of retrieved instances that are significant, while recall is the fraction of relevant instances that are retrieved. Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness or quantity. High recall means that an algorithm returned most of the relevant results. High precision means that an algorithm returned more relevant results than unessential [4].

In our data itemset we have 2000 raws of data in total and 56303 which was retrieved from candidate-3 itemset based on dynamic programing and 1652 of transactions retrieved from candidate 3-itemset in our data itemsets respectively. In our data itemset we have 2000 raws of data in total and 56303 which was retrieved from candidate-3 itemset based on dynamic programing and 1652 of transactions retrieved from candidate 3-itemset in our data itemsets respectively.

After calculating the recall and precision by using the data that has been extracted during the candidate generation, recall and precision values of 2.8 of recall and 0.1 of precision have been found respectively.

IV. Performance Evaluation

In this part of performance evaluation we compared these two graphs to evaluate the performance of each graph by comparing the result generated during the mining process. In this case the results of both the dynamic programming and model based algorithm are compared in order to evaluate the performance. In this process initially 2000 transactions are used for mining PFI by using dynamic programing algorithm and model based algorithm. The data set is obtained from the website with following address http://atlas.media.mit.edu/, the website is observatory of economic complexity it makes international trade data and economic complexity indicators available through millions of interactive visualizations. In this case we are interested in the products imported by Somalia.

The followings graph evaluates the performances of both dynamic programing algorithm and model based algorithm on runtime and number of PFI extracted with various minimal supports. In this graph DP stands for dynamic programing and MB stands for model based, from the run time graph, the total time taken by dynamic programing algorithm to compute probabilistic frequent itemset is 10,000 milli seconds, and the total time taken by model based algorithm to compute probabilistic frequent itemset is 15,000 milli seconds.

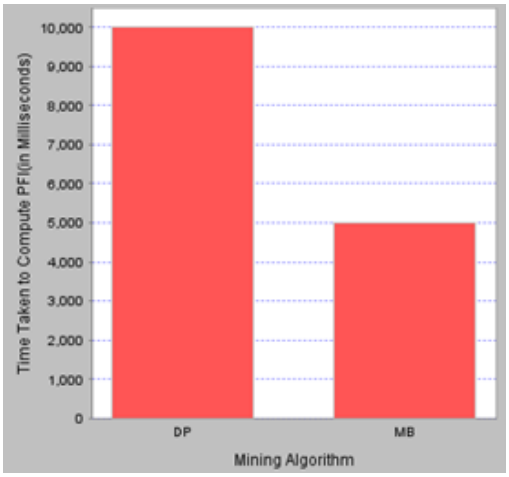

Fig. 4 Run time graph

In Fig. 4, the graph compares the time taken to generate the probabilistic frequent itemsets from large uncertain databases with various minimal supports using Dynamic Programming and Model based algorithms.

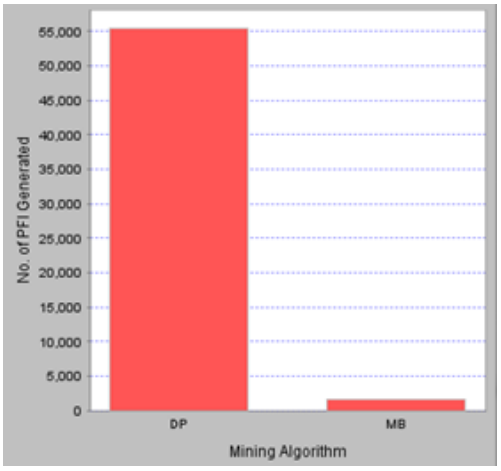

Fig. 5 FPI graph

In Fig. 5, the graph shows the number of probabilistic frequent itemset extracted with various minimal support by using both dynamic programming and model based algorithm. In the run time graph we see that the dynamic programing takes more time to extract the PFI than our proposed model based

algorithm which takes less time than the dynamic programing. Finally from performance analysis it is concluded that the proposed model based algorithm can efficiently extract the Threshold based PFI in short time.

V. CONCLUSIONS

We compared the performances of the two algorithms which are dynamic programing algorithm and model based algorithm, we found that in case of run time the dynamic programing algorithm took 10.0 seconds to compute the propabilistic frequent itemset while the model based algorithm took 5.0 seconds to compute the propabilistic frequent itemset, we see that the difference between these two algorithms when it comes to run time, the model based algorithm took less time to compute the propabilistic frequent item than the dynamic programing algorithm, so at this point the model based algorithm outperforms the dynamic programing algorithm as whole.

REFERENCES

[1] Meta Group Application Development Strategies, "Data Mining for Data Warehouses: Uncovering Hidden Patterns", July 1995.

[2] R. S. Jadon, S. Joshi, "An Implementation of Frequent Pattern Mining Algorithm using Dynamic Function", International Journal of Computer Applications, vol. 9, pp. 37–41, Nov. 2010.

[3] D. Bhalodiya, K. M. Patel, Ch. Patel, "An Efficient way to Find Frequent Pattern With Dynamic Programming Approach," Nirma University *International Conference On Engineering, NUiCONE '13*, vol. 20, pp. 569–571, Nov. 2013.

[4] M. W.David, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," Journal of Machine Learning Technologies, vol. 2, pp. 37–63, 2011.

BIOGRAPHY

Isse Hassan received the BSc degree in Computer Science and Information Technology from Islamic University of Technology and MSc in Computer Engineering from Marmara University.

Melih Kırlıdoğ received the BSc degree in Civil Engineering from Middle East Technical University and MBA and PhD in MIS from Wollongong University.