I J C T A, 9(20), 2016, pp. 283-293
© International Science Press

Frequent Query Mining using Association rule mining technique

Gaurav Dubey*, Ankur Tripathi and Archana Singh*****

ABSTRACT

Frequent Item set is regular used word in the rule mining — technique now a day for query process mining. To answering analytical queries we are storing data in data warehouse. These logical queries are very complex, difficult and analytical in behavior and, while process for a Big Data Warehouse, it is a time consuming process. At the same time, result of these queries is higher. We can reduce that higher time by frequent query mining through rule mining technique over a big data ware house. This frequent query mining technique will be improved response time of complex queries. Frequent query mining technique is very useful in E-commerce market for online line shopping. That kind of information is very useful for customer who will do online shopping and business decision makers to take decision. Eventually this form of information helps in different way – To increase sales of Business, To improve profit graph, prediction etc.For this, we need appropriate data to answer upcoming queries. In this paper, we have proposed approaches that identify such appropriate information, from the earlier queries based method to find out frequent query mining. That proposed technique initially extracts information based on Item set which we need to fetch through online shopping website.

Keywords: Data Mining, Association Rule, Data Warehouse, Decision Making, Queries, Frequent Item Set.

1. INTRODUCTION

The internet has permitted collection of a lot of information gathered from over the globe. All Companies are ambitious to work this information for their advantage keeping in mind the end goal to stay competitive in the business sector. This needs to be getting information in a way that would encourage basic leadership.On previous methodology, information has retrieved in light of the customers questions however on last methodology, information has gathered and demands have modeled based on customers beside the kept data information which warehoused prior. Warehoused big data depends in last methodology [13] wherever mined data against dissimilar sources of data information. Data warehouse is a relational database that stores historic data using for query and analysis rather than for transaction processing. So as usual Data ware house contains historical data which is derivative of transactional and it support data for decision making [12]. These queries, which we will use for decision support technologies are gathering, analyzing and taking decisions over data. So as a result when apply these queries against a big data warehouse, it will take much time for performance. Hence that approach is very expensive for frequent queries and performance time is also high. Business groups are targeting to attain at quicker resolutions, this performance time request needs to be optimized.

Therefore, frequent itemsets views constructed using rule mining technique are efficient to answering upcoming queries. In this paper we have proposed approaches that identify such appropriate information, from the earlier queries based method to find out frequent query mining. That frequent itemsets have created by using frequently used query, having frequently accessed information in the past which is competent to answer for upcoming question posted by customers within very less time.

* Amity School of Engineering and Technology Amity University Uttar Pradesh, *Email: gdubey@amity.edu*

** Amity School of Engineering and Technology Amity University Uttar Pradesh, *Email: ankur.trip@gmail.com*

*** Amity School of Engineering and Technology Amity University Uttar Pradesh, *Email: asingh27@amity.edu*

The paper is organized as follows: The approach for selecting frequent queries using association rule mining technique is given in section 2 and an example based on it is given in section 3. Section 4 is the conclusion.

2. RELATED WORK

Numerous practical solutions are present for grouping data used on functional analytic queries; however concern of execution over frequent queries still should be tended [18]. Frequent item set mining plays key role in numerous data mining fields as association rules [1, 2, 4] warehousing [10], correlations, clustering of high-dimensional biological data, and classification [9]. The logical inquiries when handled against less scale of data, as a result it takes less search space and less time for searching the customer(user)queries. Frequent itemsets view is developed among objective for enhancing analytical queries execution [20, 10]. This requires these contain data that is important for noting future questions for customer. The future queries of customer can't be subjectively recognized, as per future queries information in frequent itemsets views that is not equipped for noting future inquiries and subsequently bring about a pointless space[31, 32] overhead. Choosing such future queries information, from the big Warehoused data has mentioned in collection view sets [6]. Collection view sets by means of choosing proper views for Frequent itemsets to increase performance time of queries where adjusting [7, 8] asset requirements similar to data loading ,memory space, storage and so forth [8, 6]. So far, there are numerous methods were mentioned to as frequent itemsets views,accumulate information which have higher probability to respond most upcoming query in better query response and performance time[14, 15]. As far as anyone is concerned, the only one language that allows to combine mining queries in Mine Rule [2]. Therefore, that is convenient to classify specific subject area queries from all the historical data.So that methodology distinguishes topic based area for inquiries by combining earlier present query with the Nearest Neighbor Clustering technique [13]. That may be extensive quantities of Item set queries in every topic based Classification area [19]. The majority of similar Item Sets contains same type of data information while Other Item sets containing different type of data information. Those queries which are retrieving similar type of information are analytical and the information more prone to be gotten to in future. Therefore choosing such queries are helpful, as the information retrieved by them will probably be gotten to by most future queries

3. METHODOLOGY

This methodology intends toward choose query, since along through all the inquiry which have performed over big warehoused data into earlier period, having connected with essential data to responding upcoming data warehouse queries. The methodology chooses similar inquiry for combining customer related query within history data base. This is proceed from fetching inquiry to facilitate frequent itemsets regularly for fetching customer information on demand basis to each item set which user want through their query. This frequent itemsets contain data that is important for noting future questions for customer. We have discussed 2 point here, which would be explained below.

3.1. Topic based Classification

Methodology which we are considering about the data information that maximum queries used lying on big warehoused data proceeds toward topic based classification, moreover some query beyond the topic area. Those retrieved information from historical database for specific subject area needs to be joining together.This most similar query searching (neighborhood behavior) is processed using the DICE coefficient [7]. As indicated by DICE coefficient, similarity coefficients will be measure between a pair of identical objects.Here we can apply DICE coefficient on two item set I_i and I_j. So that Dice Coefficient measure [7] are specified as

$$SIM(I_i,\ \ I_j) = \frac{2\,|\,R(I_i)\ \ \cap\ \ R(I_j)\,|}{|\,R(I_i)\,| + |\,R(I_j)\,|}$$

In above mathematical equation $R(I_i)$ & $R(I_j)$ are the relationship between Item I_i to I_j.

By applying above mathematical equation, we can compute the similarity among the earlier present queries. Through these similarity queries, we would be able to construct similarity matrix. The Nearest Neighbor clustering method is based on the similar clusters for grouping immediate queries into similarity matrix. So that the similarity between group of queries will be known. Respectively specific subject area foreach cluster of queries will be identified. So the algorithm Subject Area Identification should be noted that the similarity matrix depend on Nearest Neighbor clustering [13]. So for classify Item sets, based on topic area we have proposed an algorithm which are depicted in Figure 1.

This algorithm introduces queries Item count QIC & CounterCluster CoC.Now set the value as 1 for both QIC and CoC.Hence allocate first Item query Q_{QIC} and earlier present Item queries IQ_p, into cluster C_{CoC}.After that we can consider next query Item in IQ_p and its nearest neighbor (NN), that should be in relations which have maximum similarity in similarity matrix and is recognized Item set query which are allocated for clusters. So If minimum similarity threshold £ is less than or equal to Similar Item set in that case next query Item sets are allocated just before equivalent clusters. Else Item set query is allocated to new clusters. That process step carries on for all Item set.

```
ALGORITHM Topic Based Classification
Given Input: $IQ_p$ : Already Presented Item query.
£ : Threshold Defined for Minimum Item query relationship.
SIMILARITY MATRIX : Relationship Between Item Matrix set.
Output: Item Set Queries on Cluster $IC_Q$
Procedure:
1: To assign the value of queries Item count QIC = 1 & Counter Cluster CoC=1.
2: Now allocate Item query $Q_{QIC}$ & present Item queries $IQ_p$, into cluster $C_{CoC}$
3: Increase QIC to 1.
4: To get nearest neighbor (NN) for $Q_{QIC}$ between present Item queries IQp, into allocated cluster $C_{CoC}$.
5: To use SIMILARITY MATRIX, Suppose MAXSIM represents relationship among $Q_{QIC}$ & NN Item query
   in CoC. Supposing K1 is close Clustor.
6: IF MAXSIM >=£, allocate $Q_{QIC} = C_{K1}$ or else increase Counter Cluster CoC to 1 & allocate $Q_{QIC}$ to CoC.
7: But If each queries have taken so END else GOTO Third STEP.
```

Fig ure 1: Topic Based Classification Algorithm on Nearest Neighbor technique

That may be extensive quantities of Item set queries in every topic based Classification area. The majority of similar Item Sets contains same type of data information while Other Item sets containing different type of data information. Those queries which are retrieving similar type of information are analytical and the information more prone to be gotten to in future.

3.2. Frequent Query Selection

As specified above Frequent Query Mining technique could be reduced the query fetching time and it would be able to give answers to most future queries. This requires the Frequent Query Mining contains the applicable and required information.That information can’t be randomly recognized, as the result only frequent query mining view which contains that is not equipped for giving responses to future inquiries. The methodology distinguishes such important and required data by fetching query that approach often access data. So frequent queries give information about the data that have high probability of noting future questions and as a result we can use rule mining technique for constructing the frequent query selection on behalf of consequent topic based classification.

DIC (Dynamic Itemset Counting)[4] usage relationship based on association rule mining technique. The algorithm (Frequent queries selection) happening on DIC [4] which are depicted in Figure 2. That algorithm select query on topic based classification area and taking Threshold Defined for Minimum Item query relationship as INPUT and generates all frequent item sets as OUTPUT for the equivalent area based subject.

That algorithm first denotes the unfilled connection Item Set Ø(null) through SOLID SQUARE (¡%) along with denote one to one connection Item sets by DASHED CIRCLES(Ì%) and other Item sets relation is untouched. In that case we can read Item set queries IM against the Item query set IQS through query transaction File(TQ). Through above rule we can increment the particular counter intended for associations Item set in TQ and denotes among DASHES. So DASHED CIRCLES(Ì%) counter used for Item relation set IR exceed with threshold Defined for Minimum Item query relationship ß, roll DASHED CIRCLES(Ì%) to DASHED SQUARE for IR. But some instant set relation ISR of relation R have the majority of its subgroup as SOLID or DASHED SQUARES, denote another connection Item sets ISR through DASHED CIRCLES(Ì%).The quantity of sweeps is increased by Item set queries IM and that method would be recurring. The procedure proceeds till there are not any more DASHED SQUARE & CIRCLE connection Item group. This should be reducing the Item set queries reaction time and direct to decision making for the customer

```
Input set:
IQS:   ITEM QUERY SET
TQ:    ITEM SET BASED ON TRANSACTION QUERIES TABLE
ß:     THREHOLD DEFINED FOR MINIMUM ITEM QUERY RELATIONSHIP
IM:    MINIMUM ITEM SET PER TRANSACTION
Output:
IFQS:  ITEMSET FREQUENT QUERY SET
Procedure:
Primarily Itemsets are identified
SOLID SQUARE ( %) ss = Defined Ø(null) Value (Complete repeated Itemset)
SOLID CIRCLE(Ï%) sc= Defined Ø(null) Value (Complete irregular Itemset)
DASHED SQUARE(°) ds = Defined Ø(null) Value (Suspicious repeated Itemset)
DASHED CIRCLE(Ë%) dc = {Defined relative Itemsets} (Suspicious irregular Itemset)
       while ( (ds ≠ 0) ¦¦ (dc ≠ 0))
start
       To Examine IM query starting through (IQS) into (TQ)
           for all QUERY INTO (TQ)
```

```
start
    for all Item sets (IR) into Dashed Square(ds) & Dashed Circle(dc)
start
    if Relation Set (IR)into Transaction (TQ)
    Increase (IR) that is (IRC) to (IRC) + 1
    for all Item relation (IR)into DASHED CIRCLE(dc)
    if (RC >= ß) then
MOVE (IR) →(dc) to (ds)
    if(instant subgroup (ISR) OF (IR) for all (ss) & (ds) therefore
MOVE ([R]+ [ISR])→ [DC]
    stop
      for all Item relation (IR) into (ds)
    if(scan IR through all query) else
GOTO (IR) →(ss)
    for all Item relation IR into dc
    if(scan IR through all query) then
GOTO (IR) →(sc)
    stop
(IM++)
    stop
      FREQUENT ITEM RELATION SET [FIRS] THAT HOLD IN (ss)
    Evaluate (IFQS) into (IQS) hold at least on (FIRS).
```

Figure 2: Frequent Query Selection based on DIC

4. EXAMPLE

Let us take earlier posted Item set query based on customer response through the data warehouse which is given in Figure 3. The schema based on earlier posted Item relations are presented in Figure 4.

Using DICE Coefficient[7] we have computed the similarity between the Item set queries in Figure 3 which was posted by customer. In Figure 5 we have constructed Similarity matrix through similarity between the Item set queries.

Now taking subject area S1 using the Frequent Query Selection based on DIC through algorithm from Figure 2.So query beside DIC relationship is depicted by Figure 6.

Threshold Defined for Minimum Item query relationship

£= 0.5.Topic based Identification is presented in Figure 7.

```
Q1  SELECT M.Mob_Brand, M.Mob_Price, C.Cam_Brand FROM Mobile M, Laptop L, Camera C WHERE M.Mob_Brand=C.Cam_Brand AND M.Mob_Features = L.Lapi_ Features AND M.Mob_Brand = ‘Samsung’
Q2  SELECT T.Tab_Brand,TV.Tel_brand,WP.Water_Price FROM Tablets T, Telivisons TV ,Water Purifier WP WHERE T.Tab_Brand = TV.Tel_brand AND TV.Tel_brand = WP.Water_Brand AND T.Tab_Price <=’10000$’
Q3  SELECT T.Tab_Brand,TV.Tel_brand,P.Price_Low FROM Tablets T, Telivisons TV ,Price PWHERE T.Tab_Brand = TV.Tel_brand AND TV.Tel_ Features =T.Tab_Features AND P.Price_Low <=’5000$’
Q4  SELECT T.Tab_Brand,TV.Tel_brand,S.Screen_Size FROM Tablets T, Telivisons TV ,Screen SWHERE T.Tab_Brand = TV.Tel_brand AND S.Screen_Size =T.Screen_Size AND T.Tab_Brand =’Samsung’
Q5  SELECT M.Mob_Brand, M.Mob_Price, C.Cam_Brand,L.Lapi_Price FROM Mobile M, Laptop L,Camera C WHERE C.Cam_Brand = M.Mob_Brand AND M.Mob_Features = L.Lapi_ Features AND M. Mob_Brand = ‘Samsung’ AND M.Mob_Price <=’2000$’;
Q6  SELECT T.Tab_Price,TV.Tel_Price,P.Price_Low FROM Tablets T, Telivisons TV ,Price PWHERE TV.Tel_ Features =T.Tab_Features AND T.Tab_Price_Low =P.Price_Low
Q7  SELECT C.Price,SW.Brand FROM Smart_Watch SW, Speaker SP ,Camera C WHERE C.Cam_Brand =SW.SmWtch_Brand AND SP.Speaker_Brand=’Philips’ and SW.Warranty=’4-years’;
Q8  SELECT M.Mob_Brand, M.Mob_Price, T.Tab_Brand,L.Lapi_Price,T.Tab_Price FROM Mobile M, Laptop L,Tablets T WHERE T.Tab_Brand=M.Mob_Brand AND M.Mob_Features = L.Lapi_ Features ANDM. Mob_Brand = ‘Iphone’
Q9  SELECT M.Mob_Mem_Size,L.Lapi_Mem_Size,MS.Size FROM Mobile M, Laptop L,Memory_Size MS WHERE L.Lapi_Mem_Size= M.Mob_ Mem_Size AND MS.Size=’4GB’ M. Mob_Brand = ‘Samsung’ AND M.Mob_Price <=’2000$’;
Q10 SELECT T.Tab_Warranty,TV.Tel_Warranty,RF.Features FROM Tablets T, Telivisons TV , REFRIGERATORS RF WHERE T.Tab_Brand = TV.Tel_brand AND RF.REFI_Band =TV.Tel_Brand AND TV.Tel_Warranty =’3-Years’
Q11 SELECT T.Tab_Price_Low.Tel_brand,S.Screen_Size FROM Tablets T, Telivisons TV ,Screen SWHERE T.Tab_Price_Low = TV.Tel_Price_Low AND S.Screen_Size =T.Screen_Size AND T.Tab_Brand =’LG’
Q12 SELECT RF.REFI_BRAND,TV.Tel_Price,P.Price_Low FROM REFRIGERATORS RF, Telivisons TV ,Price P WHERE TV.Tel_ Features =RF.REFI_Features AND P.Price_Low<=’2000$’;
Q13 SELECT T.Tab_Price,T.Screen_Size_,P.Price_Low FROM Tablets T,Screen S ,Price PWHERE T.Screen_Size=’7.9INCH’ AND T.Tab_Price_Low =P.Price_Low
Q14 SELECT M.Mob_Brand,M.Mob_Price,L.Lapi_Brand FROMMobile M, Laptop L,Display_Size DS WHERE M.Mob_Brand=L.Lapi_Brand AND M.Mob_Features=L.Lapi_Features AND M.Mob_Brand=‘Iball’ AND M.Mob_Size=’8INCH’
Q15 SELECT M.Mob_Brand, M.Mob_Price, T.Tab_Brand FROM Mobile M, Tablets T,Sound SWHERE M.Mob_Brand=T.Tab_Brand AND M.Mob_Features =T.Tab_Features AND S.Qulity=’MP3’
Q16 SELECT M.Mob_Brand, M.Mob_Price, T.Tab_Brand,L.Lapi_Brand FROM Mobile M, Tablets T,Laptop LWHERE M.Mob_Brand=T.Tab_Brand AND M.Mob_Features =’8 megapixels’
Q17 SELECT SP.SPEK_Price,TV.Tel_Price,P.Price_Low FROM Speaker SP, Telivisons TV ,Price PWHERE TV.Tel_Features =’HD’ AND SP.SPEK_Price>=’1000$’;
Q18 SELECT C.Cam_Warranty,TV.Tel_Warranty,RF.Features FROM Camera C, Telivisons TV , REFRIGERATORS RF WHERE C.Cam_Brand = TV.Tel_brand AND RF.REFI_Band =TV.Tel_Brand AND RF.REFI_Warranty =’3-Years’
Q19 SELECT M.Mob_Brand, M.Mob_Price, T.Tab_Brand,L.Lapi_Brand FROM Mobile M, Tablets T,Laptop L WHERE M.Mob_Brand=T.Tab_Brand AND M.Mob_Features =L.Lapi_Features AND M.Mob_Brand=’Sony’;
Q20 SELECT M.Mob_Camera, M.Mob_Price, T.Tab_Camera,L.Lapi_Price,T.Tab_Price FROM Mobile M, Laptop L,Tablets T WHERE T.Tab_Camera=M.Mob_Camera AND M.Mob_Features = L.Lapi_ Features AND M. Mob_Camera = ’16 megapixals’
```

Figure 3: Previously Posed Queries on a Data Warehouse

```
Mobile(Mob_Brand,Mob_Price,Mob_Features,Mob_Camera,Mob_Size,Mob_Mem_Size)
Laptops(Lapi_Brand,Lapi_Price,Lapi_Features,Lapi_Mem_Size)
Tablets(Tab_Brand,Tab_Price,Tab_Features,Tab_Camera,Screen_Size,Tab_Warrenty)
Camera(Cam_Brand,Price,Features,Cam_Warrenty)
Telivisons(Tel_Brand,Tel_Price,Tel_Features,Tel_Warranty)
Refrigerators(REFI_Brand,Features,REFI_Warranty)
Smart Watch(Brand,Warranty)
Speakers(Speaker_Brand,SPEK_Price)
Water Purifier (Water_Brand,Water Price)
```

Figure 4: Relations Accessed by Previously Posed Queries Q1 . . Q20

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
Q1	1	0	0	0	1	0	0.333	0.666	0.666	0	0	0	0	0.666	0.333	0.333	0	0.333	0.666	0.666
Q2	0	1	0.666	0.666	0	0.666	0	0.333	0	0.666	0.666	0.333	0.333	0	0.333	0.333	0.333	0.333	0.333	0.333
Q3	0	0.666	1	0.666	0	1	0	0.333	0	0.666	0.666	0.333	0.666	0	0.333	0.333	0.666	0.333	0.333	0.333
Q4	0	0.666	0.666	1	0	1	0	0.333	0	0.666	1	0.333	0.666	0	0.333	0.333	0.333	0.333	0.333	0.333
Q5	1	0	0	0	1	0	0.333	0.666	0.666	0	0	0	0	0.666	0.333	0.666	0	0.333	0.666	0.666
Q6	0	0.666	1	1	0	1	0	0.333	0	0.666	0.666	0.666	0.666	0	0.333	0.333	0.666	0.333	0.333	0.333
Q7	0.333	0	0	0	0.333	0	1	0	0	0	0	0	0	0	0	0	0.333	0.333	0	0
Q8	0.666	0.333	0.333	0.333	0.666	0.333	0	1	0.666	0.333	0.333	0	0.333	0.666	0.666	1	0	0	1	1
Q9	0.666	0	0	0	0.666	0	0	0.666	1	0	0	0	0	0.666	0.333	0.666	0	0	0.666	0.666
Q10	0	0.666	0.666	0.666	0	0.666	0	0.333	0	1	0.666	0.666	0.333	0	0	0.333	0.333	0.666	0.333	0.333
Q11	0	0.666	0.666	1	0	0.666	0	0.333	0	0.666	1	0.333	0.666	0	0.333	0.333	0.333	0.333	0.333	0.333
Q12	0	0.333	0.333	0.333	0	0.666	0	0	0	0.666	0.333	1	0.333	0	0	0	0.666	0.666	0	0
Q13	0	0.333	0.666	0.666	0	0.666	0	0.333	0	0.333	0.666	0.333	1	0	0.333	0.333	0.333	0	0.333	0.333
Q14	0.666	0	0	0	0.666	0	0	0.666	0.666	0	0	0	0	1	0.333	0.666	0	0	0.666	0.666
Q15	0.333	0.333	0.333	0.333	0.333	0.333	0	0.666	0.333	0	0.333	0	0.333	0.333	1	0.666	0	0	0.666	0.666
Q16	0.333	0.333	0.333	0.333	0.666	0.333	0	1	0.666	0.333	0.333	0	0.333	0.666	0.666	1	0	0	1	1
Q17	0	0.333	0.666	0.333	0	0.666	0.333	0	0	0.333	0.333	0.666	0.333	0	0	0	1	0.333	0	0
Q18	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0	0	0.666	0.333	0.666	0	0	0	0	0.333	1	0	0
Q19	0.666	0.333	0.333	0.333	0.666	0.333	0	1	0.666	0.333	0.333	0	0.333	0.666	0.666	1	0	0	1	1
Q20	0.666	0.333	0.333	0.333	0.666	0.333	0	0	0.666	0.333	0.333	0	0.333	0.666	0.666	1	0	0	1	1

Figure 5: Similarity Matrix showing similarity between earlier posed queries Q1 to Q20

Q1	Mobile	Laptop	Camera
Q5	Mobile	Laptop	Camera
Q8	Mobile	Laptop	Tablets
Q9	Mobile	Laptop	Memory_Size
Q15	Mobile	Tablets	Sound
Q16	Mobile	Tablets	Laptop
Q19	Mobile	Tablets	Laptop
Q20	Mobile	Laptop	Tablets

Figure 6: Query beside DIC relationship

Taking at first QIC=1 & CoC=1

Allot Q_{QIC} that is Q_1 to cluster CCoC that is C_1

So C_1 =[Q_1]

Increase QIC to QIC++ that is QIC=2

FIND MAXSIM QQIC that is Q_2 = 0 among NEAREST NEIGHBOR QUERY Q_1

While MAXSIM < £, Increase CoC to CoC++ that is CoC =2 & CCoC that is C_2=[Q_2]

Increase QIC to QIC++ that is QIC=3

FIND MAXSIM QQIC that is $Q_{3=}$0.666 among NEAREST NEIGHBOR QUERY Q_2 into C_2

While MAXSIM >£, MOVE $Q_3 \rightarrow C_2$

C_2=[Q_2, Q_3]

Increase QIC to QIC++ that is QIC=4

FIND MAXSIM QQIC that is Q_4 is 0.666 among NEAREST NEIGHBOR QUERY Q_2 & Q_3

While MAXSIM >£, MOVE $Q_3 \rightarrow C_2$

C_2=[Q_2, Q_3, Q_4]

Increase QIC to QIC++ that is QIC=5

FIND MAXSIM QQIC that is Q_5 is 0.666 among NEAREST NEIGHBOR QUERY Q_1

While MAXSIM >£, MOVE Q_5 to C_1

C_1={Q_1, Q_5}

Increase QIC to QIC++ that is QIC=6

FIND MAXSIM QQIC that is Q_6 is 0.666 among NEAREST NEIGHBOR QUERY Q_2, Q_3 &Q_4

While MAXSIM >£, MOVE Q_6 to C_2

C_2={Q_2, Q_3, Q_4, Q_6}

In above steps we are conceded out to classify cluster queries. C_1, C_2, C_3, C_4 & C_5 cluster querie characterize five topic based classification areas S_1, S_2, S_3, S_4 and S_5 correspondingly which are given below:

S_1=[Q1, Q5, Q8, Q9, Q15, Q16, Q19, Q20]

S_2= [Q2, Q3, Q4, Q6, Q7, Q10, Q12, Q13, Q18]

S_3=[Q_{11}], S_4=[Q_{14}], S_5=[Q17]

Figure 7: Topic Based Classification via earlier posted queries Q1 to Q20

The frequent Item set into S1 for THREHOLD DEFINED FOR MINIMUM ITEM QUERY RELATIONSHIP ß = 0.5 and IM = 4 is identified as given in Fig. 8.

From Fig. 6, the frequent Item relations set in S1 are {Mobile, Laptop}. Therefore, the fetched frequent Item queries are Q1, Q5, Q8, Q9 & Q20 where those hold {Mobile, Laptop} as part of FROM part. On same way the frequent Item relations set into S2 are {Tablets, Telivisons} and therefore the fetched frequent Item are Q2, Q3, Q4, Q6 & Q10. So in that way we can construct frequent Item queries over different topic based area identification using rule mining technique.

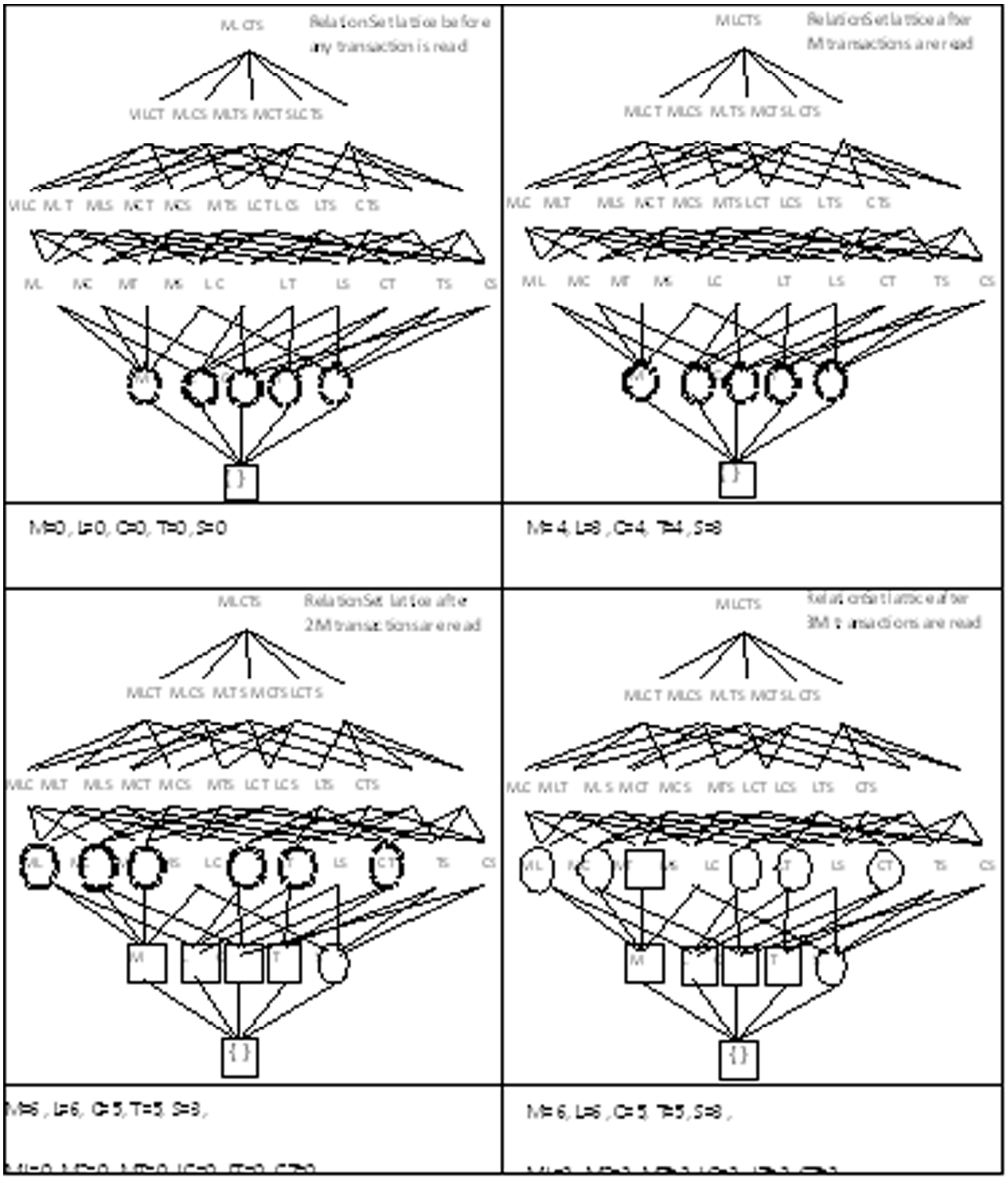
MLCTS
RelationSet lattice before any transaction is read
MLCT MLCS MLTS MCTS LCTS
MLC MLT MLS MCT MCS MTS LCT LCS LTS CTS
ML MC MT MS LC LT LS CT TS CS
M
{ }
M=0, L=0, C=0, T=0, S=0
MLCTS
RelationSet lattice after M transactions are read
MLCT MLCS MLTS MCTS LCTS
MLC MLT MLS MCT MCS MTS LCT LCS LTS CTS
ML MC MT MS LC LT LS CT TS CS
M
{ }
M=4, L=8, C=4, T=4, S=8
MLCTS
RelationSet lattice after 2M transactions are read
MLCT MLCS MLTS MCTS LCTS
MLC MLT MLS MCT MCS MTS LCT LCS LTS CTS
ML MC MT MS LC LT LS CT TS CS
M
T
{ }
M=6, L=6, C=5, T=5, S=8,
ML=0, MC=0, MT=0, LC=0, LT=0, CT=0
MLCTS
RelationSet lattice after 3M transactions are read
MLCT MLCS MLTS MCTS LCTS
MLC MLT MLS MCT MCS MTS LCT LCS LTS CTS
ML MC MT MS LC LT LS CT TS CS
M
T
{ }
M=6, L=6, C=5, T=5, S=8,

Figure 8: Classification of frequent relative Item Set in topic area S1
M: Mobile, L: Laptop, C:Camera, T: Tablets, S: Sound

5. CONCLUSIONS

In this paper, a methodology which chooses Frequent Query Mining technique for frequent Item set, through already posted queries on big warehoused data have presented. So that methodology primarily distinguishes groups of Clusters Item queries which have been already posted. Then all such type of Items clusters set which identify topic based area. This methodology chooses repeated item set over given previous queries in all topic area. That elected repeated Item set queries into all topic based identification area, classify information which have retrieved in the past and consequently select upcoming Itemset queries by customer.Frequent Query Mining technique could be reduced the query fetching time and it would be able to give answers to most future queries. Further frequent Item queries being topic specific, as they are constructed for a specific topic based area, and most future queries posed on the data warehouse are subject specific, fewer number of views are required to answer future queries. Therefore, the performance of the query time would be further decreasedand less time for searching the customer(user)queries on various decision making queries.

REFERENCES

[1] Agrawal, S., Chaudhari, S. and Narasayya, V. ‘Automated Selection of Materialized Views and Indexes in SQL databases’, In 26th International Conference on Very Large Data Bases (VLDB 2000), Cairo, Egypt, pp. 495-505, 2000.

[2] Aouiche, K. and Darmont, J. ‘Data mining-based materialized view and index selection in data warehouse’, In Journal of Intelligent Information Systems, Pages 65 – 93, 2009

[3] Cios K.J., Pedrycz W, Swiniarski RW, & Kurgan LA. Data mining: A knowledge discovery approach. New York NY: Springer, 2012

[4] Liu X., Zhai K., & Pedrycz W. An improved rules mining method. Expert Systems with Applications, 2012, 39(1):1362–1374. doi:10.1016/j. eswa.2011.08.018.

[5] Gupta H. and Mumick I. S. ‘Selection of Views to Materialize in a Data warehouse’, IEEE Transactions on Knowledge & Data Engineering, 17(1), pp. 24-43, 2005

[6] Jiawei Han, Micheline Kamber, Morgan Kaufmann “Data mining Concepts and Techniques” Publishers, 2006.

[7] Lin D.-I and Kedem Z. Pincer Search: An algorithm for discovering the maximum frequent set IEEE Transactions on Database and Knowledge Engineering, 2002, 14 (3): 553–566.

[8] Grahne G. and Zhu G. Fast Algorithms for frequent itemset mining using FP-trees, in IEEE transactions on knowledge and Data engineering, 2005,17(10):1347-1362.

[9] Inmon W. H. ‘Building the Data Warehouse’, 3rd Edition, Wiley Dreamtech India Pvt. Ltd, 2003

[10] Lawrence, M. ‘Multiobjective Genetic Algorithms for Materialized View Selection in OLAP Data Warehouses’, GECCO’06, July 8-12, Seattle Washington, USA, 2006

[11] Lehner, W., Ruf, T. and Teschke, M. ‘Improving Query Response Time in Scientific Databases Using Data Aggregation’, In proceedings of 7th International Conference and Workshop on Database and Expert Systems Applications, DEXA 96, Zurich, 1996

[12] Shah, B., Ramachandran, K. and Raghavan, V. ‘A Hybrid Approach for Data Warehouse View Selection’, International Journal of Data Warehousing and Mining, Vol. 2, Issue 2, pp. 1 – 37, 2006

[13] Widom, J. ‘Research Problems in Data Warehousing’, 4th International Conference on Information and Knowledge Management, Baltimore, Maryland, pp. 25-30, 1995

[14] Vijay Kumar, T.V. and Devi, K. ‘Frequent Queries Identification for Constructing Materialized Views’, In the proceedings of the International Conference on Electronics Computer Technology(ICECT-2011), April 8-10, 2011, Kanyakumari, Tamil Nadu, Published by IEEE, Volume 6, pp. 177-181, 2011

[15] Vijay Kumar, T.V., Haider, M.: Selection of Views for Materialization using Size and Query Frequency, Communications in Computer and Information Science (CCIS), Volume 147, Springer Verlag, pp. 150-155, 2011

[16] Vijay Kumar, T.V., Haider, M., Kumar, S.: A View Recommendation Greedy Algorithm for Materialized Views Selection, Communications in Computer and Information Science (CCIS), Volume 141, Springer Verlag, pp. 61-70 , 2011

[17] Vijay Kumar, T.V., Haider, M.: Greedy Views Selection using Size and Query Frequency, Communications in Computer and Information Science (CCIS), Volume 125, Springer Verlag, pp. 11-17, 2011

[18] Vijay Kumar, T.V., Goel, A. and Jain, N.: Mining Information for Constructing Materialised Views, International Journal of Information and Communication Technology, Inderscience Publishers, Volume 2, Number 4, pp. 386-405, 2010

[19] Vijay Kumar, T.V. and Jain, N.: Selection of Frequent Queries for Constructing Materialized Views in Data Warehouse, The IUP Journal of Systems Management, Vol. 8, No. 2, pp. 46-64, May 2010

[20] Vijay Kumar, T.V., Haider, M.: A Query Answering Greedy Algorithm for Selecting Materialized Views, Lecture Notes in Artificial Intelligence (LNAI), Volume 6422, Springer Verlag, pp. 153-162, 2010

[21] Vijay Kumar, T.V., Haider, M., Kumar, S.: Proposing Candidate Views for Materialization, Communications in Computer and Information Science (CCIS), Volume 54, Springer Verlag, pp. 89-98, 2010

[22] Vijay Kumar, T.V., Ghoshal, A.: A Reduced Lattice Greedy Algorithm for Selecting Materialized Views, Communications in Computer and Information Science (CCIS), Volume 31, Springer Verlag, pp. 6-18, 2009

[23] Theodoratos, D. and Xu, W. ‘Constructing Search Spaces for Materialized View Selection’, In 7th ACM Internatioanl Workshop on Data Warehousing and OLAP (DOLAP 2004), Washington, USA, 2004

[24] Shah, B., Ramachandran, K. and Raghavan, V. ‘A Hybrid Approach for Data Warehouse View Selection’, International Journal of Data Warehousing and Mining, Vol. 2, Issue 2, pp. 1 – 37, 2006

[25] Theodoratos, D. and Sellis, T. ‘Data Warehouse Configuration’. In proceeding of VLDB pp. 126-135, Athens, Greece, 1997

[26] Teschke, M. and Ulbrich, A. ‘Using Materialized Views to Speed Up Data Warehousing’, Technical Report, IMMD 6, Universität Erlangen-Nümberg, 1997

[27] Shah, B., Ramachandran, K. and Raghavan, V. ‘A Hybrid Approach for Data Warehouse View Selection’, International Journal of Data Warehousing and Mining, Vol. 2, Issue 2, pp. 1 – 37, 2006

[28] O’Neil, P. and Graefe, G. ‘Multi-Table joins through Bitmapped Join Indices’, SIGMOD Record, Vol. 24, No. 3, pp. 8-11, 1995

[29] Mohania M., Samtani S., Roddick J. and Kambayashi Y. ‘Advances and Research Directions in Data Warehousing Technology’, Australian Journal of Information Systems, 1998

[30] Lehner, W., Ruf, T. and Teschke, M. ‘Improving Query Response Time in Scientific Databases Using Data Aggregation’, In proceedings of 7th International Conference and Workshop on Database and Expert Systems Applications, DEXA 96, Zurich, 1996

[31] Jain, A.K. and Dubes, R.C. “Algorithms for Clustering Data”. Englewood Cliffs NJ: Prentice Hall, 1988

[32] Gouda, K. and Zaki,M.J. GenMax : An Algorithm for Mining Maximal Frequent Itemsets’, Mining and Knowledge Discovery, 2005, 11: 1-20