I.J. Information Technology and Computer Science, **2017, 6, 43-49**
Published Online June 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2017.06.06

Application of Materialized View in Incremental Data Mining Operation

Debabrata Datta
St. Xavier's College (Autonomous), Kolkata, India
E-mail: debabrata.datta@sxccal.edu

Kashi Nath Dey
University of Calcutta, Kolkata, India
E-mail: kndey55@gmail.com

***Abstract*—**Materialized view is a database object used to store the results of a query set. It is used to avoid the costly processing time that is required to execute complex queries involving aggregation and join operations. Materialized view may be associated with the operations of a data warehouse. Data mining is a technique to extract knowledge from a data warehouse and the incremental data mining is another process that periodically updates the knowledge that has been already identified by a data mining process. This happens when a new set of data gets added with the existing set. This paper proposes a method to apply the materialized view in incremental data mining.

***Index Terms*—**Apriori algorithm, confidence value, data warehousing, incremental data mining, materialized view, support value.

I. Introduction

A view is a database object which is used to store the results of a query and can be thought of as a logical table often used for query processing. A materialized view is a relatively newer concept and is another database object that, unlike a normal view, physically stores the results of a query set. It can also be defined as cache which is mainly used for faster query processing. Materialized views help in eliminating the overhead involving expensive joins and aggregations that need to be performed for a large class of queries and offer significant improvements in query processing time, particularly for aggregation queries over tables of considerable size [1].

Materialized views essentially materialize the results produced by the query in order to reduce access times [2]. Use of materialized view is mainly associated with data warehouse containing a huge amount of historical data. From this large data set, many data may not be actually useful for the future use. Here is the exact role of data mining which has become an important branch of database applications and is extensively used in the fields of decision support system, market analysis or any type of data processing. It is a method of finding out knowledge from data set through some mathematical modelling.

Incremental data mining is a form of data mining in which the existing mined data set is updated with the newly mined data with the process being repeated iteratively. The proposed method as discussed in this paper is related to the use of materialized view in implementing incremental data mining operations. The method deals with a database transaction set which consists of a set of attributes. The main algorithm on which the present method has been set up is Apriori algorithm as described in [3] and using this algorithm, the frequencies of occurrences of these attributes are obtained and the attributes which are above a particular threshold value of the frequency level are selected for materialization. Later with a new set of database transactions with the same set of attributes, the frequencies are calculated and if any change is observed with the frequencies, the materialized view gets updated. The next section describes the various research work done so far on the field of incremental data mining. Section 3 describes the methodologies behind the present research work as described in this paper.

II. Related Work

In incremental data mining process, the already mined data sets need to be modified depending on the users' ever changing requirements. So, a challenge in this field is to find out the relevance of the outputs of new user queries with the existing set of data. To do this, the association between the past data and the present data is to be identified. There have been very few researches done on this field so far. The research work as described in [4] gave a method to discover the association rule among the sets of data present in a large database. Later the work was further expanded to design another algorithm called Apriori algorithm [3].

The present research work as described in this paper also depends on this algorithm. The idea of incremental data mining was at first discussed in [5] where a novel method called Fast Update Algorithm or FUP was discussed. This algorithm mainly discussed about the use of the information already available from the existing mined data in the new information processing or mining of an updated as well as an expanded database. The

research that was discussed in [5] was further expanded in [6], where an algorithm was proposed to minimize the re-computation operation to find out the updated mined data from a transaction database when some new data sets are added or some existing data sets are removed. An approach of implementing an incremental data mining operation on the sequential patterns of data was discussed in [7]. The use of materialized views in data mining operations was at first discussed in [8] where the authors had proposed a method to use materialized view suitably in data mining applications in the same way that was used in transactional databases.

An approach was proposed in [2] as well to relate materialized views with data mining functionalities. In [2], the uses of a new form of databases object called Materialized Data Mining View or MDMV was discussed. This object could be used in the repetitive data mining queries by periodically refreshing the content of the view [2]. The next section states the Apriori algorithm which is the backbone of the proposed method. The subsequent section describes proposed research work in details. The section 5 analyzes the results obtained after applying the algorithm as described in section 4. An approach was proposed in [2] as well to relate materialized views with data mining functionalities. In [2], the uses of a new form of databases object called Materialized Data Mining View or MDMV was discussed. This object could be used in the repetitive data mining queries by periodically refreshing the content of the view [2]. The next section states the Apriori algorithm which is the backbone of the proposed method. The subsequent section describes proposed research work in details. The section 5 analyzes the results obtained after applying the algorithm as described in section 4.

III. A Short Description of the Apriori Algorithm

Since the proposed method as described in this research paper is based on Apriori algorithm, this section briefly describes the same. Apriori algorithm is an algorithm to find out the frequent item sets from a large set of data after analyzing a set of transactions on that set of data. This algorithm was proposed in a research paper as described in [3]. This algorithm is based on the identification process of the relations among items that are involved in large data sets. This is done using a rule known as the association rule. This rule may briefly be stated as follows:

Let $I = \{I_1, I_2, I_3, \ldots, I_n\}$ be a set of n number of items and let $T = \{T_1, T_2, T_3, \ldots, T_k\}$ be a set of k number of transactions where each $T_i \subseteq I$. With respect to the above defined sets, an association rule is said to be an expression of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$ with the condition that $A \cap B = \Phi$ or the null set. Now, this rule is further explained with the help of two parameters, viz., the support value and the confidence value. The support value is defined like support $(A \Rightarrow B) = P(A \cup B)$, where $P(\ )$ is a function to calculates the probability. Hence, the parameter support value identifies the percentage of transactions that contain both the attributes A and B. On the other hand, the confidence value is defined like confidence $(A \Rightarrow B) = P(B|A)$. Hence, this parameter identifies the percentage of transactions containing A that also contain B. With the help of these two parameters, Apriori algorithm says that if an item set does not support the minimum threshold value, already defined with respect to a particular set of transactions, then that item set should not be treated as a frequent item set. Otherwise, that item set should be considered to a frequent one. This observation continues with all the non-empty subsets of the initially taken item set. In other words, all the non-empty subsets of a frequent item set should also be considered as frequent ones. The execution of the Apriori algorithm consists of two steps, viz., the join step and the prune step. In the join step, a set of 'k' number of items is generated after joining a set of $(k - 1)$ number of items with itself. In the next step, i.e., in the prune step, the items which are not frequent in the set of $(k - 1)$ number of items are removed from the generated set containing 'k' number of items. The second step reduces the number of items by considering the frequency of the participating items. The detailed description of the algorithm has been given in [3].

IV. Description of the Proposed Work

In any transaction processing, the daily activities performed by the different users on a particular database are monitored. Considering this aspect, the present work as described here has two main components as mentioned below:

Initial Transaction processing component: This contains the frequently occurring sets of attributes which may be identified to form the materialized views.

Revised Transaction processing component: This contains the attributes which may be added to the exiting materialized views. This revision is done using an iterative approach.

The Initial Transaction processing component is obtained by applying the algorithm MVG_AA [9]. This algorithm uses the Apriori algorithm [3] which has been briefly discussed in the previous section. The way MVG_AA utilizes the Apriori algorithm is briefly presented below.

The Apriori algorithm consists of two basic steps, viz., join step and the prune step and since the basic idea of the Apriori algorithm is to select a confidence value and a support value, the same has been done here as well. The support value is the minimum number of times that an attribute has to occur in the set of transactions to be considered as a frequently occurring attribute and the confidence value is used to determine if an attribute should be added to the existing set of materialized views consisting of the already identified frequent attribute sets. If the percentage of transactions containing the attributes in the materialized view that also contains the concerned attribute is greater than or equal to the confidence value,

then this new attribute has been added to the materialized view. This updated materialized view is then used in the subsequent iterations.

To implement the proposed method, the input data set from a transaction processing environment is taken as a text file containing a set of 'n' number of transactions T, where $T = \{T_1, T_2, T_3, \ldots, T_n\}$, where T_i contains a set of attributes participating in the i^{th} transaction. In the proposed method, each participating attribute is identified by a positive number. So, if the number of participating attributes is stored in a variable called 'm' then the first attribute is identified by the number 1, the second one is identified by the number 2 and so on and so forth with the last attribute is identified by the number m. Moreover, since each transaction contains a set of attributes, they are mentioned in the transaction set. For example, if the i^{th} transaction contains the first attribute, the fourth attribute, the fifth attribute and the eighth attribute (with the assumption that the value of the previously considered variable 'm' >=8) then $T_i = \{1, 4, 5, 8\}$.

The proposed algorithm to implement incremental data mining operation has been named as Materialized View in Incremental Data Mining or MV_IDM. This method depends on the output generated by the algorithm called Materialized View Generation using Apriori Algorithm or MVG_AA which has been described in [9]. Since MVG_AA has been used to develop the present algorithm, the pseudo-code for MVG_AA is also given below:

```
Algorithm MVG_AA ( )
{
        Input: T = A set of 'n' number of database
        transactions and ATR = A set of attributes on
        which different transactions are to be executed
        Output: M = A set of attributes to be
        materialized
        Let T = {T1, T2, T3, …, Tn}
        Initialize M by Φ, i.e., null set
        for i = 1 to n
        do
                Let A = A set of attributes involved
                in ith transaction, denoted by Ti
                R = Apriori (A) /* R is a set which
                stores the output generated by the
                Apriori algorithm and Apriori ( ) is a
                method to invoke Apriori algorithm and
                its takes A as its parameter */
                M = M ⋃ R
        done
        S = ATR – R /* S is the set of attributes not
        selected by Apriori algorithm for
        materialization */
        R´ = Check_Confidence (S) /*
        Check_Confidence ( ) is a method which looks
        for the confidence values of the attributes in S
        and returns the attributes which satisfy the
        minimum confidence threshold value and this is
        stored in another set R´*/
        M = M ⋃ R´
        return M
}
```

With the help of the algorithm as depicted above, the algorithm for the present research work has been developed and the pseudo-code of the proposed algorithm is given below:

```
Algorithm MV_IDM ( )
{
        Input: no_iter = The number of iterations
        required to find out the incremental result on an
        already mined data set, DT = A set consisting of
        'no_iter' numbers of different sets of database
        transactions at different time frame on the same
        database and ATR = A set of attributes on which
        different transactions are to be executed.
        Output: R = The set of materialized views to be
        considered for incremental data mining
        operation and R is initialized by ϕ, i.e., null set.
        Let DT = {ST1, ST2, ST3, …, STno_iter}
        for i = 1 to no_iter
        do
                Mi = MVG_AA (STi, ATR) /* It
                returns the materialized view
                containing the most frequent itemsets
                for a single transaction set STI and all
                the participating attributes are there in
                the set ATR */
                R = R ⋃ Mi
        done
        return R
}
```

As seen from the above-mentioned pseudo-code, MD_IDM iteratively invokes MVG_AA. The number of such iterations depends on the number of times the transactions are to be considered to generate the finally mined attribute set. In MV_IDM, the set DT contains the sets of transactions. So, each element of DT is itself a set of transactions. For example, if ST_i, which is the i^{th} set of transactions to be passed as a parameter to MVG_AA, contains the first transaction, the third transaction and the seventh one (with the assumption that the value of the variable 'n' >=7, as this variable stores the number of transactions in MVG_AA) then $ST_i = \{T_1, T_3, T_7\}$. So, the i^{th} call to MVG_AA from MV_IDM would take the transactions T_1, T_3 and T_7 only to identify the attributes to be considered for storing in the materialized view for that particular iteration. Similarly, in other iterations also, other transactions are considered and accordingly new materialized view is returned.

The algorithm MV_IDM has been executed on a few randomly generated test cases as described in the next section. Each test case has been assumed to be a database transaction. The final outcome of the algorithm has been chosen to be the concluding data set to be mined.

V. Results and Analysis

The algorithm MV_IDM, as described in the previous section, has been implemented on a system having Windows Operating System with Intel Core 2 Duo 2.0 GHz CPU (x86) processor and 2 GB of primary memory. For the testing purpose of the algorithm, four sets of different transactions have been chosen, with each of the transactions having twenty different attributes of a randomly generated database. For the implementation purpose, each attribute has been assigned with a number starting from 1. So the twenty different attributes have been numbered from 1 to 20. Moreover, in each set, eighteen different transactions have been chosen. To justify the applicability of the proposed algorithm, the participating attributes in each of the transactions have been randomly chosen. The first four tables, i.e., table 1 to table 4 store the participating attributes in each transaction under each set. So, the value of the 'no_iter' variable, as specified by the algorithm MV_IDM, is 4 here for the above-mentioned test case. In each of the iterations, a new set of 18 different transactions have been added to the existing list of transactions which is initially set to empty.

Moreover, in the very first iteration, the number of transactions considered is eighteen; in the second iteration, the number of transactions considered has been 36, because a new set of 18 different transactions have been added with the existing 18 transactions which have already participated in the first iteration. In the same way, the steps of the next iterations have been executed and accordingly the number of transactions has also increased.

The algorithm MV_IDM has been iteratively applied on the different transaction sets as depicted from table 1 to table 4 and the result obtained after each of the iterations is described after the results shown in table 4.

The minimum confidence value in each of the iterations has been chosen to be 0.7, which has been a standard value in different applications. On the other hand, the minimum support value has been determined to be dependent on the number of transactions considered in each of the iterations. In this case, it has been chosen to be 1/3rd of the total number of the participating transactions. This value has been chosen because it is a moderate value in the sense it is less than ½ which may become relatively high to consider participation of the attributes and also it is more than ¼ which may be relatively a small number for consideration. According to this specification, in the first iteration, since the number of participating attributes is 18, the minimum support value has been chosen to be 6 (1/3rd of 18). In the next iteration, as the number of participating attributes has been increased to 36, the minimum support value also got increased and has been set to 12 (1/3rd of 36). In this way, the minimum support values for the other iterations have also been chosen.

For the implementation purpose, the set of attributes participating a single transaction is assigned with a binary number as a whole. This idea has been taken from [9] and according to that, the participating attributes have been identified by '1's and the attributes which are not participating in a transaction have been identified by '0's and thus forming a binary pattern. For example, if in a transaction attribute numbers 6, 4 and 3 are only participating and the attributes with numbers 5, 2 and 1 are not participating then the corresponding binary pattern is formed to be 101100, which has a decimal equivalent value of 44. So, the number '44' identifies the participating attributes in this transaction.

Table 1. The first set of transactions

Transaction number	Participating attributes
1	1,3,4,5,6,7,8,10,11,12,15,16,17,20
2	2,3,4,6,7,8,10,12,13,14,15,16,17,18,19,20
3	2,9,14,16
4	1,2,3,4,5,6,7,8,9,10,11,12,14,16,17,18,19
5	1,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
6	2,4,5,6,7,8,9,10,11,12,13,14,15,18,19,20
7	1,2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19,20
8	1,2,3,8,13,15
9	2,6,16
10	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
11	2,3,5,6,7,8,11,12,13,15,18
12	1,3,10,11,15,18,19
13	13
14	2
15	3,4,5,6,10,11,12,13,14,16,17,18,20
16	9,10
17	1,2,3,4,6,7,8,9,12,13,14,15,16,17,18,19,20
18	2,3,4,7,8,10,12,16,18,20

In the first iteration, since the number of transactions was 18, the minimum support value was chosen to be 6 as already explained and after this iteration, only one frequent set was identified and that was [6, 8, 12, 14, 16, 18, 19] which had the minimum support value of 6, i.e., this set has occurred 6 times in the whole set of transactions as depicted in table 1. In accordance with the reason explained above, this set is identified by the binary number 01101010100010100000 (with the MSB corresponds to the 20th attribute and the LSB corresponds to the 1st attribute) which has the equivalent decimal number 436384.

The next step was to find out the confidence value of the other attributes on [6, 8, 12, 14, 16, 18, 19]. This calculation is based on the method as described in section 3. It was calculated that no other attributes crossed the minimum confidence value of 0.7 and hence after the first set of 18 transactions, the materialized view set has become [6, 8, 12, 14, 16, 18, 19] and the non-materialized view set has become [1, 2, 3, 4, 5, 7, 9, 10, 11, 13, 15, 17, 20]. So, it can be concluded that the seven attributes present in the materialized view have got preference over others and thus only these attributes can be mined. Now, the aim of the algorithm is to identify whether any new attribute is mined in the next iteration or not. It may also happen that after the next iteration, any existing attribute may have to be omitted from the set with the introduction of the new ones.

Table 2. The second set of transactions

Transaction number	Participating attributes
1	5,8,9,10,15,17,20
2	2,20
3	1,2,3,4,5,6,8,9,10,12,13,14,16,17,18,19,20
4	6,7,12,13,14
5	2,4,7,8,10,13,14,15,20
6	1,2,3,5,6,7,8,9,10,11,12,14,15,16,17,18,19,20
7	4,16,18
8	1,4,14
9	1,2,3,5,6,9,10,11,12,13,14,16,17,18,19,20
10	1,2,4,7,8,9,12,13,14,17,20
11	1,2,3,4,7,8,9,11,13,14,15,17,18,19
12	1,2,3,5,7,8,9,11,12,13,14,15,17,19,20
13	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
14	1,2,4,12,15,16
15	1,4,6,7,8,10,11,12,13,14,15,16,17,18,19,20
16	1,2,3,4,6,7,8,11,12,13,14,15,16,17,18,20
17	1,5,6,9,10,12,14,15,16,18,20
18	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

Similarly, in the second step of the iteration, as shown in table 2, a new set of 18 transactions was considered along with the already considered set and hence in this iteration, the total number of participating transactions was 36 and the minimum support values has been increased to 12 ($1/3^{rd}$ of 36).

Table 3. The third set of transactions

Transaction Number	Participating attributes
1	6,10
2	4,5,7,15
3	1,3,4,6,8,9,11,12,14,15,16,17,18,19,20
4	2,3,5,8,9,11,18,20
5	1,2,3,4,5,7,8,9,11,12,13,14,15,17,18,19,20
6	1,2,4,7,9,10,11,13,14,15,16,17,18,19,20
7	2,3,4,7,10,11,15,16,17,18,19,20
8	1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
9	1,2,3,4,6,8,9,12,14,15,17,20
10	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
11	3,6,8,12,13,14,16,17,19
12	1,2,3,5,6,7,11,16,17,19,20
13	2,3,5,6,7,8,10,12,15,17,19,20
14	2,3,4,5,6,8,9,10,11,12,14,15,16,18,19,20
15	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19
16	1,4,6,8,11,12,14,15,20
17	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
18	8

After applying the algorithm, only the set [10, 14, 16, 18] was identified to be materialized, i.e., crossed the threshold value of the minimum support. It was also found that the above-mentioned set has got the minimum support value of 13. So, like the previous iteration, the next step was to find out the confidence value of the other attributes on [10, 14, 16, 18], i.e., on 0010101000100000000_2 or 172544_{10}.

Again, it was identified that no other attribute has confidence value above or equal to the predefined minimum confidence value of 0.7. So, after the second iteration, the materialized view set became [10, 14, 16, 18] and hence this set contained the attributes to be mined. The interesting point is that one new attribute, i.e., attribute number 10, has been added in the materialized set and the four attributes (6, 8, 12 and 19) got removed from the materialized set that was generated after the first iteration. After the third iteration also, as shown in table 3, no other attribute could cross the minimum confidence value and the materialized view was identified to be the set [10, 17, 19].

But after the fourth iteration, i.e., the final iteration, as shown in table 4, as it has happened for the test case, more sets have satisfied the minimum support value as well as the minimum confidence value. In this iteration, the total number of transactions involved was 72 and hence the minimum support value was chosen to be 24 ($1/3^{rd}$ of 72). With the result of this final iteration, the next steps are to be executed.

Table 4. The fourth set of transactions

Transaction Number	Participating attributes
1	1,5,14
2	1,3,5,10,12,14,19,20
3	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
4	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
5	1,2,3,4,5,7,8,9,11,12,13,14,15,17,18,19,20
6	5,20
7	1,3,4,6,7,8,9,10,11,12,13,14,15,16,19,20
8	4,10,11,14,16,17,18
9	7,12,14
10	1,2,6,8,10,12,13,17,19
11	5,12,15
12	1,2,3,6,11,12,13,14,15
13	3,4,14,17,18
14	1,3,7,15,19
15	2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19,20
16	2,6,9,11,12,15
17	4
18	1,2,3,4,5,6,7,8,10,11,12,13,14,15,16,17,19,20

Table 5. The sets with their respective support values after the fourth iteration

Set of attributes	Decimal Value	Support Value
[3, 12, 14, 16]	43012	25
[4, 16]	32776	28
[3, 14, 16]	40964	25
[3, 16, 19]	294916	25
[15, 16]	49152	26
[14, 16, 18]	172032	25
[14, 16, 19]	303104	25
[6, 12, 14, 16, 19]	305184	24
[8, 14, 16]	41088	24
[12, 15, 16]	51200	24
[12, 14, 16, 19]	305152	24

It was found that more than one set of attributes have crossed the threshold value of 24. The sets and their corresponding support values are given in table 5. This table also lists the decimal values of all the selected sets of attributes.

As a part of the algorithm, the next step was to find the final materialized view and to do that, it was required to check the confidence values of the attributes on each of the attribute set already chosen as displayed in table 5.

If the confidence value of any such attribute crosses the threshold value of the minimum confidence value chosen then only the attribute will be considered in the final view materialization set.

It may be observed from table 5 that in all of the transactions attribute number 16 has appeared. So, in the next phase of the implementation, it was required to check whether any other attribute or attribute set in each transaction had confidence value above the minimum confidence value.

Table 6. The confidence values of all other attributes on attribute number 16

Set of Attributes	Set of other attributes on attribute number 16	Decimal value of the set in column 2	Confidence value of the set in column 2 on attribute number 16
[3, 12, 14, 16]	[3,12,14]	10244	0.6756756756756757
[4, 16]	[4]	8	0.7567567567567568
[3, 14, 16]	[3.14]	8196	0.6756756756756757
[3, 16, 19]	[3.19]	262148	0.6756756756756757
[15, 16]	[15]	16384	0.7027027027027027
[14, 16, 18]	[14.18]	139264	0.6756756756756757
[14, 16, 19]	[14,19]	270336	0.6756756756756757
[6, 12, 14, 16, 19]	[6,12,14,19]	272416	0.6486486486486487
[8, 14, 16]	[8,14]	8320	0.6486486486486487
[12, 15, 16]	[12.15]	18432	0.6486486486486487
[12, 14, 16, 19]	[12,14,19]	272384	0.6486486486486487

The next table, i.e., table 6, depicts the confidence values obtained in all of the above-mentioned cases. It is observed from table 6 that only in the second and the fifth rows, the confidence value is above the minimum confidence value of 0.7. These two rows correspond to the attribute number 4 and the attribute number 15 respectively.

Hence, it may be concluded that along with the attribute number 16, two more attributes, i.e., attribute 4 and attribute 15 may be considered for the final set of materialized view and hence in the final set of mined data. So, the finally selected set of attributes for data mining after the above-mentioned four iterations is [4, 15, 16]. This selection of attributes for data mining is purely based on the frequency of occurrence of the attributes.

Thus, materialized view has been successfully used in implementing data mining operations, specifically when the input data sets are changing dynamically.

VI. Conclusion

The proposed method has discussed about the possible use of materialized view in incremental data mining where periodic update operations are required. Since this method is based on Apriori algorithm, only the frequencies of the attributes are considered for view materialization. The frequencies of the attributes have been calculated based on the historical transactions performed on the database. There may be other possibilities and factors for considering materialized view in incremental data mining. The factors may include the relevance of the output of a query with the existing mined data set and the space constraint of the mined data. With the introduction of these as the additional parameters, the updation of the mined data may be more specific. One major drawback of the proposed algorithm has been with regards to the number of transactions considered in each of the iterations. Here, eighteen new transactions have been considered in each of the iterations. In practice, the number of transactions may be much more than that. Further, in each of the new iterations, the participating set of transactions of the previous iteration has also been considered. So, if the 'no_iter' value if large, then in the final iteration, the number of transactions to be considered would have been huge. In this work, the Apriori algorithm has been invoked iteratively within MVG_AA ( ) algorithm. Moreover, MVG_AA ( ) has been invoked iteratively from MVG_IDM ( ). So, asymptotically, the Apriori algorithm is invoked in the polynomial time of 'no_iter'. Additionally, the complexity of the Apriori algorithm can be controlled by modifying the minimum support value.

References

[1] Debabrata Datta and Kashi Nath Dey, "A Soft Computing Approach of Materialized View Creation", *Proceedings of the 3rd International Conference on Business & Information Management*, National Institute of Technology, Durgapur, India, January, 2016.

[2] T. Morzy, M. Wojciechowski and M. Zakrzewicz, "Materialized Data Mining Views", *Proceedings of the*

Fourth European Conference on Principles of Data Mining and Knowledge Discovery, 2000, pp. 65 – 74.

[3] Agarwal R. and Srikant, R., "Fast Algorithms for Mining Association Rules", *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, pp. 487 – 499.

[4] R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases". *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1993, pp. 207 – 216.

[5] D. W.-L. Cheung, J. Han, V. Ng, and C. Y. Wong, "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique", *Proceedings of the 12th International Conference on Data Engineering*, 1996, pp. 106 – 114.

[6] Thomas S., Bodagala S., Alsabti K. and Ranka S, "An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases", *Proceedings of the International Conference on Knowledge Discovery in Databases*, 1997, pp. 263 – 266.

[7] Parthasarathy, M., J. Zaki, M. Ogihara and S. Dwarkadas, "Incremental and Interactive Sequence Mining", *Proceedings of the ACM CIKM Conference*, 1999, pp. 251 – 258.

[8] Czejdo, Bogdan, Morzy, Mikolaj, Wojciechowski, Marek and Zakrzewicz, Maciej, "Materilaized View in Data Mining", *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, 2002, pp. 827 – 831.

[9] Debabrata Datta and Kashi Nath Dey, "Materialized View Generation using Apriori Algorithm", *International Journal of Database Management Systems*, Vol - 7, No. – 6 ISSN: 0975-5705, 2015, pp. 17 – 27.

[10] Morzy, MikoÃlaj, Morzy, Tadeusz and Kr ́olikowski, Zbyszko, "Incremental Association Rule Mining Using Materialized Data Mining Views", *Proceedings of the International Conference on Advances of Information Systems*, 2004, pp. 77 – 87.

[11] Baralis E., Paraboschi S. and Teniente E, "Materialized view selection in a multidimensional database", *Proceeding of the 23rd International Conference on Very Large Data Bases*, 1997, pp. 156 – 165.

[12] Chaudhuri, S. & Dayal, U., "An Overview of Data Warehousing and OLAP Technology". *In ACM Sigmod Record*, Volume 26, Issue 1, 1997, pp. 65 – 74.

[13] Wojciechowski M. and Zakrzewicz M., "Itemset Materializing for Fast Mining of Association Rules", *Proceeding of the 2nd Conference on Advances in Databases and Information Systems*, 1998.

[14] Imielinski T. and Mannila H., "A Database Perspective on Knowledge Discovery", Communications of the ACM, Vol. 39, No. 11, 1996, pp. 58 – 64.

[15] P.P. Karde and V.M.Thakare, "Selection & Maintenance of Materialized View and It's Application for Fast Query Processing: A Survey", *International Journal of Computer Science & Engineering Survey*, Volume 1, Number 2, 2010, pp. 16 – 29.

[16] T. Nalini, A. Kumaravel and K. Rangarajan, "A comparative study analysis of materialized view for selection cost", International Journal of Computer Science & Engineering Survey, Volume 3, Number 1, 2012, pp. 13 – 22.

Authors' Profiles

Debabrata Datta is presently an Assistant Professor in the Department of Computer Science, St. Xavier's College (Autonomous), Kolkata. He has a teaching experience of more than 10 years both at the undergraduate level as well as the postgraduate level of Computer Science and Applications. His research interests include data warehousing and data mining. He has published more than fifteen research papers in different international peer-reviewed journals as well as conferences.

Kashi Nath Dey is an Associate Professor in the Department of Computer Science & Engineering, University of Calcutta, India. He has about 8 years of industrial experience in different IT companies in India. He has about 26 years of experience in teaching and research. He has more than 35 research publications in different international conferences and international journals and authored 7 books published by Pearson Education (India).

How to cite this paper: Debabrata Datta, Kashi Nath Dey,"Application of Materialized View in Incremental Data Mining Operation", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.6, pp.43-49, 2017. DOI: 10.5815/ijitcs.2017.06.06