International Journal of Latest Trends in Engineering and Technology
Vol.(9)Issue(4), pp.184-191
DOI: http://dx.doi.org/10.21172/1.94.28
e-ISSN:2278-621X

FSVMRT: MATERIALIZED VIEW SELECTION ALGORITHM IN DATA WAREHOUSE USING 'R'

Dr. B. Ashadevi[1] & V. Sindhuja[2]

Abstract - A data warehouse processes a given set of queries by utilizing the multiple materialized views. One of the critical decisions involved in the process of designing a data warehouse for optimal efficiency, is the materialized views selection. Due to space and maintenance cost constraint, the materialization of all views is unfeasible. The total cost associated with the materialized views to be minimized by selecting a suitable set of views in data warehousing . In this paper, we present a framework for selection of views to materialize based on the K- means clustering method using 'R' to overcome the problem resulting from conventional view selection algorithms. In the presented algorithm, Framework for Selection of Views to Materialize using Reduced Table (FSVMRT), we first generate reduced tables in the data warehouse using K – means clustering by R tool, and then we consider the combination of reduced tables as materialized views instead of a combination of the original base relations. This algorithm optimizes storage space and query processing cost of the views. Thus, an efficient data warehousing system is the outcome.
Keywords: Data Warehouse, Data Warehousing, Views, Materialization, View Selection, Materialized View Selection, Clustering

1. INTRODUCTION

The accumulation of data has led to the recent availability of outsized archives of data in industry and organization. The decision making process is faced by critical problems due to the employment of these bulk data. These problems can be managed by the developing new data models and decision support systems. Warehousing is an emerging technique that retrieves the data from distributed autonomous probably heterogeneous information sources and integrates the retrieved data [1]. On-Line Analytical Processing and Decision Support Systems utilize the large volume of extracted and summarized data stored in an information base referred as a data warehouse [2]. The data warehousing technologies is the basis for the effective embarking of many industries, for instance, manufacturing financial services, transportation, telecommunications, utilities and healthcare. In order to collect data from many data sources, a data warehouse uses an update-driven approach that communicates through networks both locally and internationally. A solid platform of consolidated historical data is provided for analysis by the data warehouse system and it also distributes such analysis to local and remote users [3]. In order to provide effective solution for the queries posted to the data warehouse, the intermediate results obtained in the query processing are stored in the data warehouse. This can avert the access of the original data sources by the users [4].

There are several research works related to view selection algorithms viz., [5], [6], [7], [8], [9], [10] and [11]. Ashadevi, B and Balasubramanian, R [5] proposed framework for selecting views to materialize which takes in to account all the cost metrics associated with the materialized views selection.[6] The piece of work also addressed the preservation of existing materialized view. The framework optimizes the maintenance, storage and query processing cost as it selects the most cost effective views to materialize. [7] the most cost effective views have been selected for materialization by the framework and the maintenance, storage and query processing cost of the views have been optimized. [8] Proposed the Algorithm for Selection of Views to Materialize using Reduced Table (ASVMRT) finds high density clusters from the dimensions of the given tables, then, produces the reduced tables using the found clusters. Next, the Multidimensional View Processing Plan (MVPP) is produced using the reduced tables, and finally, materialized views are selected from the MVPP in accordance with cost estimation. The aggregate functions are discussed in [9]. A heuristic-based greedy method that uses AND, OR, and AND-OR graphs is proposed in [10]. However, evaluation of this approach is omitted. An algorithm called HAMVD is proposed in [11].

Therefore, In this paper, our proposed algorithm that improves the speed and space problem than existing algorithms. In FSVMRT, which uses the K – means clustering technique using 'R' to select materialized views for rapid query response in a data warehouse, once clusters are found on the basis of the attributes of relation dimensions, a reduced table is then generated as the produced clusters are referenced. The generated reduced tables are the relations used for producing an MVPP in the FSVMRT (Framework for Selection of Views to Materialize using Reduced Table). After we produce an MVPP using the

[1] Assistant Professor, Department of Computer Science, M.V.Muthiah Govt. Arts College for Women, Dindigul- 624001, Tamil Nadu, India
[2] Research Scholar, Department of Computer Science, M.V.Muthiah Govt. Arts College for Women, Dindigul- 624001, Tamil Nadu, India

generated reduced tables, we then process and select the views effectively in the produced MVPP using the FSVMRT. For the justification of the proposed algorithm, the 'pubs' database used for educational purposes. We shows the experimental results in which both time and space costs were approximately better than conventional algorithms.
The rest of the paper is organized as follows; Section 2 describes a data warehouse and related works on View Selection Problem. FSVMRT is proposed in section 3, and section 4 compares FSVMRT and conventional algorithms through experimentation. Finally, we conclude and suggest future works in section 5.

2. DATA WAREHOUSE AND RELATED WORKS ON MATERIALIZED VIEW SELECTIONS

In this section, we provide a brief introduction of the data warehouse and related works on selection of materialized views.

2.1. Data Warehouse

A data warehouse is a subject-oriented, integrated, time-varying, non-volatile collection of data [12]. The Data Warehouse is the heart of the architected environment, and is the foundation of all decision support system (DSS) processing. On-Line Analytical Processing (OLAP) and Decision Support Systems utilize the large volume of extracted and summarized data stored in an information base referred as a data warehouse [13].

2.2. Related Works

Gupta, H et al. [2005] proposed a theoretical framework for the general problem of selection of views in a data warehouse. They have presented competitive polynomial-time heuristics for a selection of views to optimize total query response time under a storage space constraints, for some important special cases of the problem that occur in practice. They have also presented provably competitive heuristics.In [16], the researchers proposed a framework, which is based on specification of multiple view processing plan (MVPP), to present the problem formally and they proposed a heuristic algorithm based on individual optimum query plans. But they did not use any resource constraint.Liabio, W et al. [1997] explained an A* search to pick the best set of views when only the maintenance cost is to be minimized. The problem of materialized view selection under a disk space constraint S explained in Gang Gou et al. [2006]. However, the proposed A* algorithm can find the optimal solution very efficiently when S is small, and observed that A* algorithm might coverage slowly when S is large. To avoid this problem, developed a new competitive A* algorithm in order to improve the quality of solution.
There are many other approaches on selection of common views to be materialized.[18] proposed a simple and fast heuristic algorithm called pick by size (PBS) to solve the materialized view selection problem and explored its performance. They pointed out that PBS runs several orders of magnitude faster than the heuristic algorithm proposed by Harinarayan, V et al. [1996] and is fast enough to make the exploration of the time-space trade-off feasible during system configuration. Furthermore, they have examined the view selection problem when subsets of aggregates can be computed using chunks. Shukla, A et al. [1998] and showed that the benefit of the views selected by PBS can be greater than the ones selected without chunk based precomputation.Barlalis, E et al. [1997] explained the number of representative queries is extremely small with respect to the total number of elements of the complete data cube. Using such indications (inputs), they have explained the technique to select views and an algorithm to perform selection that will reduce the solution space by considering only the relevant elements of the multidimensional lattice.
In order to improve the efficiency of problem, Lee. M and Hammer. J. [2001] assume that the set of materialized views and then ask the question: How do we to rewrite the given OLAP query to make the best use of existing materialized views? They have developed algorithms for the rewrite as well as identifying the materialized views that will best answer the query.
Gray, J et al. [1997] proposed the data cube as a relational aggregation operator generalizing group-by, crosstabs, and sub-totals. Dynamic view selection problems are an important for supporting fast online queries on such databases. In order to solve view selection problems, one needs the sizes of the various views which are obtained from running group-by queries. Time required for running such queries can be reduced by an order of magnitude by running parallel group-by queries. An interesting variant has the objective of minimizing the maximum weighted number of rows to be retrieved in responding to any query from the set of queries [22].

3. FSVMRT (FRAMEWORK FOR SELECTION OF VIEWS TO MATERIALIZE USING REDUCED TABLES)

In a different manner of conventional algorithms, we present an algorithm for selecting views to materialize using K - means clustering method.

3.1. Fsvmrt

In general, FSVMRT algorithm has the following steps:
Step 1: Find high-density clusters based on K – means clustering using R
Step 2: Generate reduced tables based on the result of step 1.
Step 3: Establish MVPP using reduced tables.
Step 4: Select materialized views based on the query response time and view maintenance cost.

FSVMRT (τ, n, T, Q, SC)

```
 {
/* n: number of queries or tables */
/* T: set of target tables */
/* Q: set with n queries */
/* SC: user's input space constraint */
C=∅ ; /* set of clusters */
RT=∅ ; /* set of reduced tables */
VP=∅ ; /* set of views used in query processing plan */
MV=∅ ; /* set of views to be materialized */

for (i=0; i<n; i++)
{
C = C U find_cluster using K – means ;
 }
for (i=0; i<n; i++)
{
RT = RT ∪ generate_reduced_table(Ci, Ti, RTi);
}
make_mvpp(n, Q, RT);
select_view(VP);
........
........
........
}
```

3.2. Fsvmrt Example

In this section, we expose each step of the FSVMRT through an example. We chose the SQL Server 9.0 'authors' table of the pubs database, which is broadly used for educational purposes. Fig. 1 and 2 show the pubs database schema and authors table consisting of pubs, respectively.Once the dimension is selected using K – means clustering using R we can produce the reduced table.

Fig. 3 shows the reduced table of authors relation from 'pubs' database. The same method results in reduced tables for all relations in a data warehouse. The comparison of ASVMRT [8] and FSVMRT shown in Table 1.

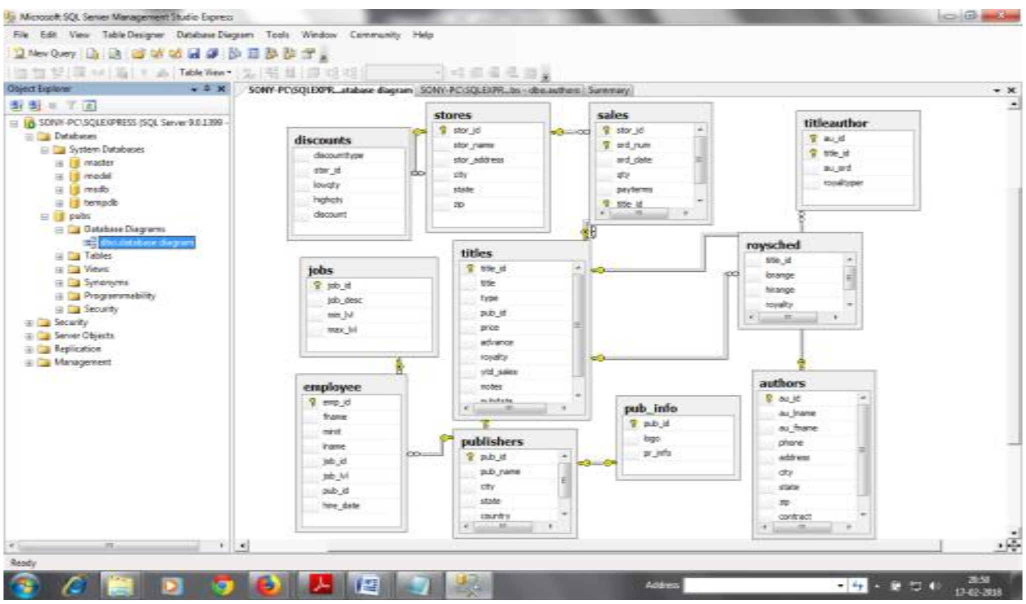

Fig.1 pubs database schema

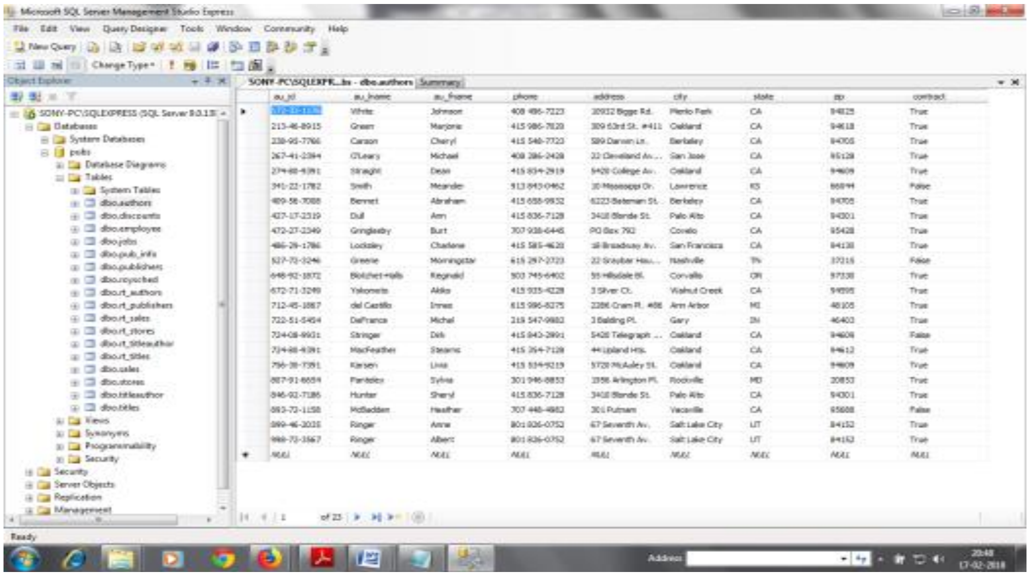

Fig. 2 authors table

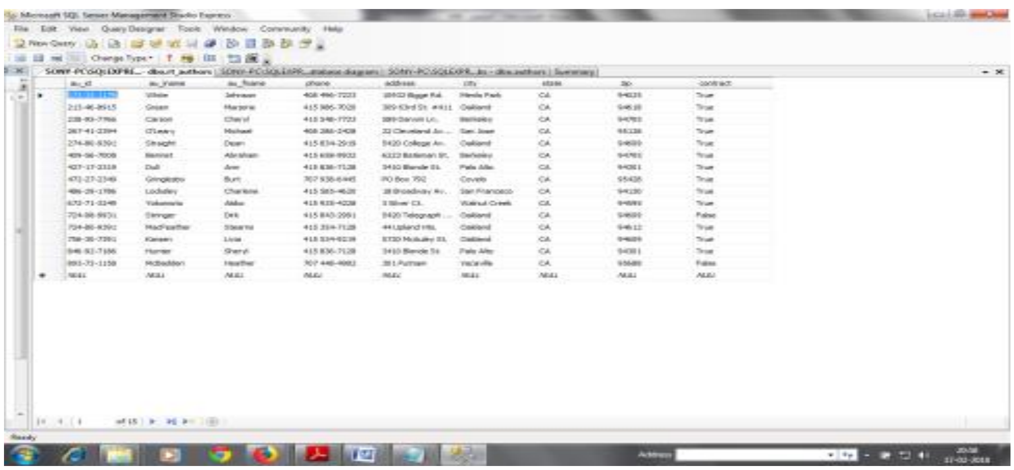

Fig. 3 Reduced table for authors table

Pubs database			
Tables	No.of Queries in Base Relation	No.of Queries in Reduced tables (FSVMRT)	No.of Queries in Reduced tables (ASVMRT)
Authors	23	15	15
Publisher	8	6	6
Sales	21	9* (indicates queries reduced)	11
Sores	6	3	3
Titleauthor	25	10	10
Titles	18	5*	18

Table 1. Comparison of FSVMRT and ASVMRT

In third step of FSVMRT, we established MVPP using the reduced tables. For an illustration of the third step of the algorithm, assume there are 4 queries.

- q1: What is the average on year-to-date sales of CA residents with a value of greater than 80 in royalty per?
- q2: What are the top 3 kinds of bestseller books from 1993 to 1995 in CA region?
- q3: Among the books with high value of royalty per, what are the titles of the books which are about economics and with price greater than $15?
- q4: What are the books printed by an American publisher which are about psychology?

In Fig. 4, □ indicates a base relation, o is for intermediary value, and • is used for a query. Once an MVPP is established as shown in Fig. 4, views to be materialized are selected considering cost. The base unit of cost estimation used in the paper is the number of tuples as adopted in [5], [6] and [7].

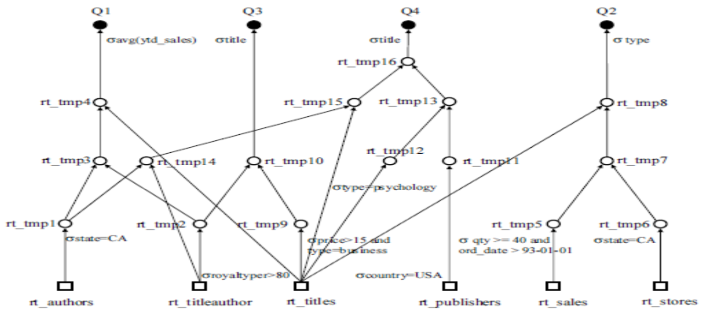

Fig. 4 MVPP for 4 queries-reduced tables

If we select the rt_authors relation as the materialized view, the total cost Ct is calculated by addition of view processing time-cost Ca(42)(the number of tuples of rt_authors(15) , rt_temp1(15), rt_temp3(6) and rt_tmp4(6), since only rt_authors, rt_temp1, rt_tmp3 and rt_tmp4 are used to process Q1) and view maintenance cost Cm(0)(rt_authors is a base relation). Total cost T is calculated by total cost of (Q1+ Q2+ Q3+ Q4). In a similar manner, we can fill in table 2. In Table 2 the first column indicates the relations used in MVPP, the second column is the query frequency (FQ), the third is the number of tuples(t#), and the fourth, fifth, and sixth are view processing time-cost(Ca), view maintenance(Cm), and total cost(Cv), respectively. The final column represents total cost (T) for all the queries.

$$Ca = \left(\sum^{n} \text{selected queries}\right) \quad \rightarrow \quad 1$$

$$Cm = \sum_{i=1}^{n} Ca * FQ \quad \rightarrow \quad 2$$

$$Cv = \sum_{i=1}^{n} Ca + Cm \quad \rightarrow \quad 3$$

$$\text{Total Cost(T)} = \sum_{i=1}^{n} Cv \quad \rightarrow \quad 4$$

3.3. Analysis and Features of FSVMRT

In the first step, the high-density cluster for target base relations is found based on the K-means clustering method using ‘R’. For each dimension of the table, the dimension with the maximum density value is selected for further process. The k-means clustering technique is not only providing potentially useful information, but also improving query processing time and saving view storage space can be achieved.

In the second step, reduced tables containing the only corresponding tuples are produced based on K-means clustering method using ‘R’. While conventional algorithms consider all the tuples of a base relation for materializing, the targets of materializing are restricted to the tuples of the reduced tables in the proposed algorithm FSVMRT. Therefore, it can achieve the goals of improvement in query response time and saving of storage for views. Note that it requires larger storage space (for intermediary reduced tables) and takes more time for clustering. However, off-line tasks of the clustering phase and production step of reduced tables do not lower the performance of a data warehouse system, since it is almost impossible to process tasks such as updating and maintaining views on-line in a data warehouse containing scores of terabytes of data.

In the third step of the algorithm, we produced an MVPP by using the reduced tables generated in the previous step.

In the fourth step, the views which can derive benefits in the case of materialized ones were selected within the bounds of the user’s input space constraint, while considering view processing time cost and view maintenance cost in the produced MVPP. The conventional algorithms consider only the cost for join operation and restrict query frequency to the query itself. We argue that these cost estimation methods leave out some important factors in cost. In the FSVMRT, cost for the select operation is supplemented to cost estimation formulation. Also, we imposed query frequency on all the tuples consisting of the query rather than the query itself because we considered the fact that the views consisting of the query can be used in another query.

4. IMPLEMENTATION RESULTS AND ANALYSES

In this section, we first expose the implementation results on the pubs database.

4.1 Experimentation and Results in Pubs Database

The framework approach with reduced tables allows for 4 times faster speed and 2 times less storage space against full materialized view. In the simple and virtual example, only 2 relations are addressed. However, there are a number of views in a data warehouse environment. Therefore, it is crucial to improve and save on both response time and storage space as close to 2 times in terms of performance of a data warehouse. The results are shown in Table 3. and Fig. 5.The results of K-means clustering using R shown in the following Tables

authors (in city)								
	CA	IN	KS	MD	MI	OR	TN	UT
1	***0***	0	0	1	0	0	0	0
2	***15***	0	0	0	0	1	0	2
3	***0***	1	0	0	1	0	1	0
4	***0***	0	1	0	0	0	0	0

Table 4.authors based on K-means clustering using R

publishers (in country)			
	France	Germany	*USA*
1	0	0	***2***
2	1	1	***1***
3	0	0	***2***
4	0	0	***1***

Table 5.punlishers based on K-means clustering using R

sales (in payterms)			
	Net 30	*Net 60*	ON invoice
1	0	***0***	1
2	2	***2***	2
3	0	***2***	0
4	6	***5***	1

Table 6.sales based on K-means clustering using R

stores (in state)			
	CA	OR	WA
1	***1***	1	0
2	***1***	0	0
3	***0***	0	2
4	***1***	0	0

Table 7.stores based on K-means clustering using R

titleauthor (in royaltyper)							
	25	30	40	50	60	75	100
1	0	0	0	0	0	0	10
2	2	2	0	0	0	0	0
3	0	0	0	0	2	2	0
4	0	0	3	4	0	0	0

Table 8.titleauthor based on K-means clustering using R

titles (in type)						
	business	mod_cook	popular_comp	psychology	trad_cook	UNDECLIND
1	0	0	0	1	3	0
2	1	0	0	4	0	0
3	0	2	0	0	0	1
4	3	0	3	0	0	0

Table 9.titles based on K-means clustering using R

Table 10 and 11 shown the Time taken from conventional algorithms. The cost estimation approach for 4 queries (same queries as in section 3) without reduced tables is presented in Table 10. Table 11 results from referencing the entire queries log and summarizing Table 2 and 10. For a comparison of the conventional approach with ours, we assumed that the space constraint variable SC from the user's input is not specified.

			Ca					Cm				Cx				T
Tables		FQ	No.of rows	q1	q2	q3	q4	q1	q2	q3	q4	q1	q2	q3	q4	
rt_authors		1	15	33				0				33				33
rt_publishers		1	6				17				0				17	17
rt_sales		1	9		51				0				51			51
rt_stores		1	3		6				0				6			6
rt_titleauthor		2	10	46		40		0		0		46		40		86
rt_titles		4	5	32	20	28	60	0	0	0	0	32	20	28	60	140
rt_temp1	query 1 (q1)	1	15	18				18				36		0		36
rt_temp2		2	10	26		40		52		80		78		120		198
rt_temp3		1	2	3				3				6		0		6
rt_temp4		1	1	1				1				2		0		2
rt_temp5	query 2 (q2)	1	0		0				0				0			0
rt_temp6		1	9		9				9				18			18
rt_temp7		1	3		3				3				6			6
rt_temp8		1	0		0				0				0			0
rt_temp9		1	0		0				0				0			0
rt_temp10	query 3 (q3)	1	2			2				2				4		4
rt_temp11		1	0			0				0				0		0
rt_temp12		1	0			0				0				0		0
rt_temp13		1	10			10				10				20		20
rt_temp14		1	0			0				0						0
rt_temp15	query 4 (q4)	1	5				10				10				20	20
rt_temp16		1	6				11				11				22	22
rt_temp17		1	5				5				5				10	10

Table 2. Cost computation for 4 queries with reduced tables

	before clustering (in time)			after clustering (in time)			Time		
	user	System	elapsed	user	system	elapsed	User	System	elapsed
Authors	6.05	0.98	683.69	6.06	0.98	683.9	0.01	0	0.21
Publishers	6.16	0.98	745.81	6.16	0.98	745.81	0	0	0
Sales	6.87	1.09	855.6	6.89	1.09	855.61	0.02	0	0.01
Stores	6.98	1.09	1015.3	7	1.09	1015.34	0.02	0	0.01
Titleauthor	7.03	1.09	1051.4	7.05	1.09	1051.38	0.02	0	0.02
Titles	7.16	1.1	1119.8	7.17	1.1	1119.77	0.01	0	0.01

Table 3. Time taken for K-means clustering

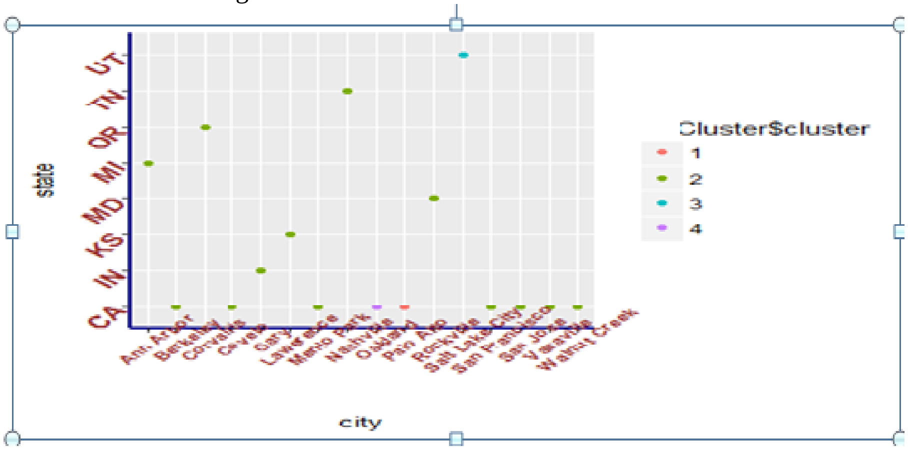

Fig 5.K-means clustering for authors table

			Ca					Cm				Cy				T
Tables		F	row	q1	q	q3	q4	q1	q	q3	q4	q1	q2	q3	q4	
Authors		1	23	45				0				45				45
Publishers		1	8				19				0				19	19
Sales		1	21		7				0				72			72
Stores		1	6		1				0				11			11
Titleauthor		2	25	84		74		0		0		84		74		158
Titles		4	18	10	8	14	112	0	0	0	0	100	80	148	11	440
temp1	query 1 (q1)	1	15	22				22				44				44
temp2		2	10	34		44		68		88		102		132		234
temp3		1	6	7				7				14				14
temp4		1	1	1				1				2				2
temp5	query 2 (q2)	1	13		1				1							0
temp6		1	18		2				2				11			11
temp7		1	3		5				5				0			0
temp8		1	2		4				4				80			80
temp9		1	2		2				2				0			0
temp10	query 3 (q3)	1	10			15				15				30		30
temp11		1	4			9				9				18		18
temp12		1	3			5				5				10		10
temp13		1	10			12				12				24		24
temp14		1	2			2				2				4		4
temp15	query 4 (q4)	1	5				10				10				20	20
temp16		1	6				11				11				22	22
temp17		1	5				5				5				10	10

Table 10. Cost computation for 4 queries without reduced tables

		Conventional algorithms	ASVMRT	FSVMRT
Partial materialization views	Materialized views	authors, publishers, sales, stores	rt_authors, rt_publishers, rt_sales, rt_stores	rt_authors, rt_publishers, rt_sales, rt_stores
	Total cost	210	147	74*
	Storage space	58	35	33*
Full materialization views	Materialized views	ALL	ALL	ALL
	Total cost	3646	2441	642*
	Storage space	220	149	116*

Table 11. Performance comparison on the pubs database

5. CONCLUSION AND FUTURE WORK

A materialized view technique FSVMRT is proposed in this paper, which uses one of the data mining techniques (K-means clustering technique). In FSVMRT, finds high density clusters from the dimensions of the given tables, and then, produces the reduced tables using the K-means clusters. Next, the MVPP is produced using the reduced tables, and finally, materialized views are selected from the MVPP in accordance with cost estimation.

A materializing views technique, FSVMRT is proposed in this paper, which adopts one of the data mining techniques (i.e., K means clustering method using R). The user is able to input a space constraint value within which materialized views are selected. These kinds of user interfaces are not found in conventional algorithms.

As shown in the experimental results, the proposed algorithm achieves almost 1.8 times better performance in terms of both query response time and storage space of materialized views. Even in the case where the value of the space constraint variable is not specified (i.e., when we assume there is no space constraint), our algorithm shows better performance in the pubs database.

Broadly, there lie two issues with the data warehouse. The first is selection of materialized views, and the other is maintenance of the views for consistency of a data warehouse. FSVMRT in this paper is in regards to the first issue. As future works, we will focus on how to update and maintain the reduced tables when there occurs any update in the source data.

6. REFERENCES

[1] Y. Zhuge, H. Garcia-Molina, J. Hammer, and J.Widom [1995], "View Maintenance in a Warehousing Environment." In Proceedings of the ACM SIGMOD Conference, San Jose, California, May.

[2] S. Chaudhuri and U. Dayal [1997]. “An Overview of Data Warehousing and OLAP Technology”. SIGMOD Record, vol: 26, no: 1, pp: 65-74.

[3] J. X. Yu, X. Yao, C. Choi and G. Gou [2003]. “Materialized view selection as constrained evolutionary optimization. IEEE Transactions on Systems, Man and Cybernetics, Part C, “ vol: 33, no: 4, pp: 458–467.

[4] C. Zhang, X. Yao, and J. Yang [2001]. “ An evolutionary Approach to Materialized View Selection in a Data Warehouse Environment. IEEE Transactions on Systems, Man and Cybernetics”, vol. 31, no.3, pp. 282–293.

[5] B.Ashadevi, R.Balasubramanian [2008], "Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment", proc. of the International Journal of Computer Science and Network security Vol. 8, No. 10, pp. 236-242.

[6] Ashadevi. B., and R. Balasubramanian [2009] . “Optimized Cost effective Approach for Selection of Materialized Views in Data Warehousing”, International Journal of Computer Science and Technology, vol. 9, no. 1, pp. 21-26.

[7] Ashadevi. B., and Balasubramanian. R, Navaneetham.P [2010]. “A Framework for the View Selection Problem in Data Warehousing Environment”, International Journal on Computer Science and Engineering Vol. 02, No. 09, 2820-2826.

[8] Jin-Hyuk Yang*, and In-Jeong Chung* [2006]. “ASVMRT: Materialized View Selection Algorithm in Data Warehouse”, International Journal of Information Processing Systems, Vol.2, No.2, 67-75.

[9] Harinarayan V., Rajaraman A., and Ullman J [1996]. “Implementing data cubes efficiently”, In Proc. of the ACM SIGMOD International Conference of Management of Data, Canada

[10] Gupta H [1997]. “Selection of views to materialized in a data warehouse”, In ICDT

[11] Yang J., Karlapalem K., Li Q., “Algorithms for materialized view design in data warehousing environment”, In Proc. of VLDB’97, pp.136-145.

[12] Inmon. W.H. [1996]. Building the data warehouse. Second Edition., John Wiley and Sons, Canada, ISBN:0471-14161-5.

[13] Chaudhuri. S., and Dayal. U. [1997]. An overview of data warehousing and OLAP technology. ACM SIGMOD, pp:65-74.

[14] Gupta, H., and Mumic. I.S.[2005]. Selection of views to Materialize in a Data warehouse. IEEE Transaction on Data and Knowledge Engineering, pp:24-43.

[15] Yang, J., Karlapalem. K., and Li. Q. [1997]. A framework for designing materialized views in a data warehousing environment. Proceedings of the Seventieth IEEE International Conference on Distributed Computing systems, USA, pp:458.

[16] Labio, W., Quass. D., and Adelberg. B.[1997]. Physical database design for data warehouse., Proceedings of the thirteenth International Conference on Data Engineering, Birmingham, UK, pp:277-288.

[17] Gang Gou., YU. J.X., and Hongjun Lu. [2006]. A* search : An efficient and Flexible approach to Materialized view selection. IEEE Transactions on Man, and Cybernetics, Part C, pp:411-425.

[18] Shukla, A., Deshpande. P.M., and Naughton. J.F. [1998]. Materialized view selection for multidimensional datasets. Proceedings of the Twenty fourth International Conference on Very Large Data Bases, USA, pp:488-499.

[19] Baralis, E., Paraboschi. S., and Tenientle. E. [1997]. Materialized view selection in a multidimensional database. Proceeding of the Twenty fourth International conference on Very Large Data Bases, pp:488-499.

[20] Lee, M., and Hammer. J. [2001]. Speeding up warehouse physical design using a randomized algorithm. International Journal of Cooperative Information System, pp:327-353.

[21] Gray, J., Chaudhuri. S., Bosworth. A., Layman. D., Reichart. D., and Venkatrao. M. [1997]. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. Data Mining and Knowledge discovery, pp:29-54.

[22] Agrawal, V., Nadhakeolyar. U., Sundararaghavan. P.S., and Ahmed. M. [2004]. Optimal view materialization in a data warehouse.Proceedings of the Decision Sciences Institute, USA, pp:5931-5936).