Greedy Views Selection Using Size and Query Frequency

T.V. Vijay Kumar and Mohammad Haider

School of Computer and Systems Sciences,
Jawaharlal Nehru University,
New Delhi-110067, India

Abstract. Greedy view selection, in each iteration, selects the most beneficial view for materialization. Algorithm HRUA, the most fundamental greedy based algorithm, uses the size of the views to select the top-k beneficial views from a multidimensional lattice. HRUA does not take into account the query frequency of each view and as a consequence it may select views which may not be beneficial in respect of answering future queries. As a result, the selected views may not contain relevant and required information for answering queries leading to an unnecessary space overhead. This problem is addressed by the algorithm proposed in this paper, which considers both the size and the query frequency of each view to select the top-k views. The views so selected are profitable with respect to size and are capable of answering large number of queries. Further, experiments show that the views selected using the proposed algorithm, in comparison to those selected using HRUA, are able to answer comparatively greater number of queries at the cost of a slight drop in the total cost of evaluating all the views. This in turn aids in reducing the query response time and facilitates decision making.

Keywords: Materialized Views, View Selection, Greedy Algorithm.

1 Introduction

Historical data has been used by industries to evolve business strategies in order to be competitive in the market. Data warehouse stores such historical data on which analytical queries are posed for strategic decision making [7]. The size of the data warehouse, which continuously grows with time, and the nature of analytical queries, which are long and complex, leads to high query response time. This query response time needs to be reduced in order to make decision making more efficient. One way to address this problem is by answering queries using materialized views, which are precomputed and summarized information stored in a data warehouse[9]. Their aim is to reduce the response time for analytical queries.

The number of possible views is exponential in the number of dimensions and therefore all cannot be materialized due to limitation in storage space available for view materialization [6]. Thus, there is a need to select a subset of views from among all possible views that improves the query response time. Selecting an optimal subset of such views is shown to be an NP-Complete problem [6]. Further, materialized views cannot be arbitrarily selected as they are required to contain information that is useful for answering future queries resulting in reduced response time. This problem

S. Unnikrishnan, S. Surve, and D. Bhoir (Eds.): ICAC3 2011, CCIS 125, pp. 11–17, 2011.

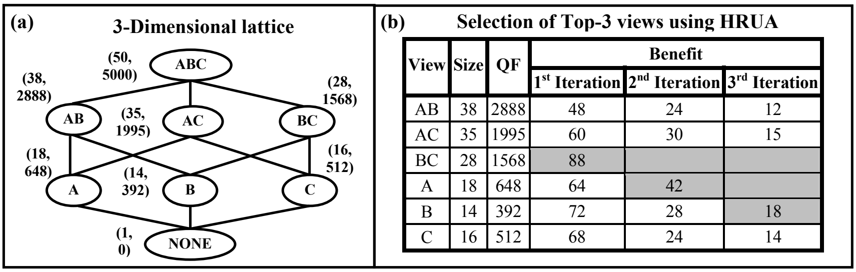
is referred to as view selection problem in literature [4]. Several view selection algorithms have been proposed in literature, most of which are greedy based [1, 2, 3, 5, 6, 8, 10, 11, 13, 14]. The greedy based view selection, in each iteration, selects the most beneficial view for materialization. Most of the greedy algorithms are focused around the algorithm in [6], which hereafter in this paper will be referred to as HRUA. HRUA selects top-k beneficial views from a multidimensional lattice. It is based on a linear cost model, where the cost is in terms of the size of the view. This cost is used to compute the benefit of each view as given below:

$$\text{BenefitV} = \sum\{(\text{Size}(\text{SMA}(W)) - \text{Size}(V)) \mid V \text{ is an ancestor of view } W \text{ in the lattice and } (\text{Size}(\text{SMA}(W)) - \text{Size}(V)) > 0\}$$

where Size(V) = Size of view V

Size(SMA(V)) = Size of Smallest Materialized Ancestor of view V.

Though HRUA uses size of the view to compute its benefit, it does not take into account the query frequency of each view, which specifies the number of queries that can be answered by a view. As a consequence, HRUA may select views that may not be beneficial in respect of answering future queries. This in turn would result in the selected views using space without having relevant and required information for answering queries. As an example, consider a three dimensional lattice shown in Fig. 1(a). The size of the view in million (M) rows, and the query frequency (QF) of each view, is given alongside the view. Selection of Top-3 views using HRUA is shown in Fig. 1(b).



View	Size	QF	Benefit		
			1st Iteration	2nd Iteration	3rd Iteration
AB	38	2888	48	24	12
AC	35	1995	60	30	15
BC	28	1568	88		
A	18	648	64	42	
B	14	392	72	28	18
C	16	512	68	24	14

Fig. 1. Selection of Top-3 views using HRUA

HRUA assumes the root view to be materialized as queries on it are unlikely to be answered by any other views in the lattice. HRUA selects A, AB and C as the Top-3 views. These selected views result in a Total View Evaluation Cost (TVEC) of 252. If the query frequency of each view is considered, the Total Queries Answered (TQA) by the selected views is 3120, from among 8003 queries. This TQA value needs to be improved upon so that greater number of queries can be answered. The algorithm presented in this paper attempts to improve this TQA value by considering both the size, and the query frequency of each view to select the Top-k profitable views for materialization. The proposed algorithm aims to select views that are profitable with respect to size and also provide answers to large number of queries.

The paper is organized as follows: The proposed algorithm is given in section 2 followed by examples based on it in section 3. The experimental results are given in section 4. Section 5 is the conclusion.

2 Proposed Algorithm

As mentioned above, HRUA selects views that are beneficial with respect to size but may be unable to answer large number of queries. As a consequence, the query response time may be high. This problem can be addressed if selected views take into account not only their size but also their ability to answer queries, i.e. query frequency. The proposed algorithm aims to select such views by considering query frequency, along with the size, of the view to select the most profitable views for materialization. The proposed algorithm assumes that past queries provide useful indicators of queries likely to be posed in future and thus use them to determine the query frequency of each view. The proposed algorithm, as given in Fig. 2, takes the lattice of views along with the size and query frequency of each view as input and produces the Top-K views as output.

```
Input:   lattice of views L along with size and query frequency of each view
Output: Top-k views
Method:
  Let
    $V_R$ be the root view in the lattice, S(V) be the size of view V,  QF(V) be the query frequency of V in the lattice,
    SMA(V) be the smallest materialized ancestor of V, D(V) be the set of all descendent views of V, MV be the set
    of materialized views, P (V) = Profit of view V, $P_M$ = Maximum Profit, $V_P$ = View with maximum profit
    FOR V $\in$ L
            SMA(V) = RootView
    END FOR
    REPEAT
            $P_M = 0$
            FOR each view V$\in$ (L – $V_R$ $\cup$ MV)
                        $V_P = V$
                        $P(V) = 0$
                        FOR  each view W $\in$ D(V) and  (S(SMA(W)) – S(V)) > 0
                                    $P(V) = P(V) + \left|\frac{QF(SMA(W))}{S(SMA(W))} - \frac{QF(V)}{S(V)}\right|$
                        END FOR
                        IF  $P_M < P(V)$
                                    $P_M = P(V)$
                                    $V_P = V$
                        END IF
            END FOR
            MV = MV $\cup$ {$V_P$}
            FOR W $\in$ D($V_P$)
                        IF S(SMA(W)) > S($V_P$)
                        SMA(W) = $V_P$
                        END IF
            END FOR
    Until |MV| < k
    Return MV
```

Fig. 2. Proposed Algorithm

The proposed algorithm, in each iteration, computes the profit of each view P(V) as given below:

$$P(V) = \sum\left\{\left|\frac{QF(SMA(W))}{S(SMA(W))} - \frac{QF(V)}{S(V)}\right| \middle| V \text{ is an ancestor of view } W \text{ in the lattice and } (S(SMA(W)) - S(V)) > 0\right\}$$

The profit of a view V is computed as the product of the number of dependents of V and the query frequency per unit size difference of V with its smallest materialized ancestor. The profit of each, as yet unselected view, in each iteration and select the most profitable view from amongst them for materialization. In this way, the proposed algorithm continues to select top profitable view until K views are selected.

Examples illustrating selection of views using the Proposed Algorithm (PA) are given next.

3 Examples

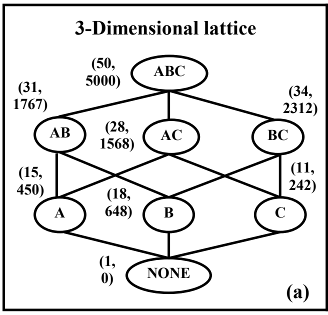
Let us consider selection of the Top-3 views from the multidimensional lattice in Fig. 1(a) using the proposed algorithm. The selection of Top-3 views is given in Fig. 3.

View	Size	QF	Profit		
			1st Iteration	2nd Iteration	3rd Iteration
AB	38	2888	96	48	24
AC	35	1995	172	86	
BC	28	1568	176		
A	18	648	128	84	41
B	14	392	144	56	56
C	16	512	136	48	48

Fig. 3. Selection of Top-3 views using PA

PA selects AC, A and BC as the Top-3 views. These selected views have a TVEC value of 254 and a TQA value of 5115. Though the TVEC value (254) of views selected using PA is slightly inferior to the TVEC value (252) of views selected using HRUA, the views selected using PA have a significantly higher value of TQA (5115), when compared with the TQA value (3120) of views selected using HRUA. That is, the views selected using PA are able to account for a greater number of queries at the cost of a slight increase in the TVEC value.

PA may also select views that not only account for more number of queries but also may have lesser or better TVEC. As an example, consider a three dimensional lattice shown in Fig. 4(a).



Top-3 views selection using HRUA

View	Size	QF	Benefit		
			I_1	I_2	I_3
AB	31	1798	76	38	19
AC	28	1568	88		
BC	34	2312	64	32	16
A	15	450	70	26	16
B	18	648	64	42	
C	11	242	78	34	24

(b)

Top-3 views selection using PA

View	Size	QF	Profit		
			I_1	I_2	I_3
AB	31	1798	172	86	
AC	28	1568	176		
BC	34	2312	128	64	32
A	15	450	140	52	52
B	18	648	128	84	41
C	11	242	156	68	68

(c)

Fig. 4. Selection of Top-3 views using HRUA and PA

HRUA selects AC, B and C as the Top-3 views as against AC, AB and C selected by PA. The views selected using PA has TVEC of 240, which is less than TVEC of

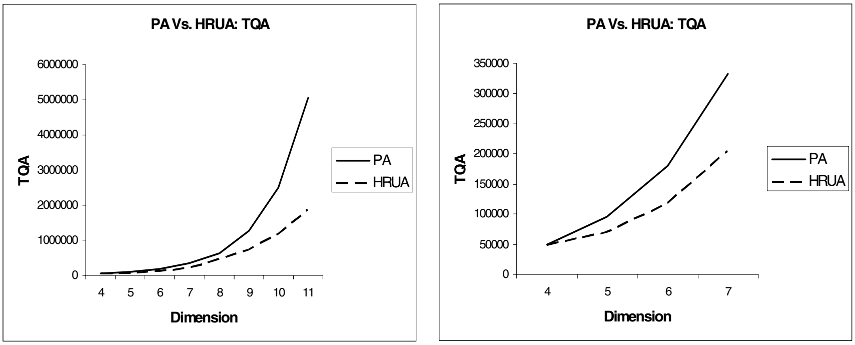
246 due to views selected using HRUA. Also, the views selected using PA has comparatively higher value of TQA of 4675 against the TQA of 2908 due to views selected using HRUA. Thus, it can be said that PA, in comparison to HRUA, is capable of selecting views that not only account for greater number of queries but also at lower total cost of evaluating all the views.

In order to compare the performance of PA with respect to HRUA, both the algorithms were implemented and run on data sets with varying dimensions. The experiment based comparisons of PA and HRUA are given next.

4 Experimental Results

The PA and HRUA algorithms were implemented using JDK 1.6 in Windows-XP environment. The two algorithms were experimentally compared on an Intel based 2 GHz PC having 1 GB RAM. The comparisons were carried out on parameters like TVEC and TQA for selecting Top-20 views for materialization. The experiments were conducted by varying the number of dimensions of the data set from 4 to 10.

First, graphs were plotted to compare PA and HRUA algorithms on TQA against the number of dimensions. The graphs are shown in Fig. 5. It is observed from the graph (Fig. 5(a)) that the increase in TQA, with respect to number of dimensions, is higher for PA vis-à-vis HRUA. This difference even exists for 4 to 7 dimensions as evident in the graph shown in Fig. 5(b).

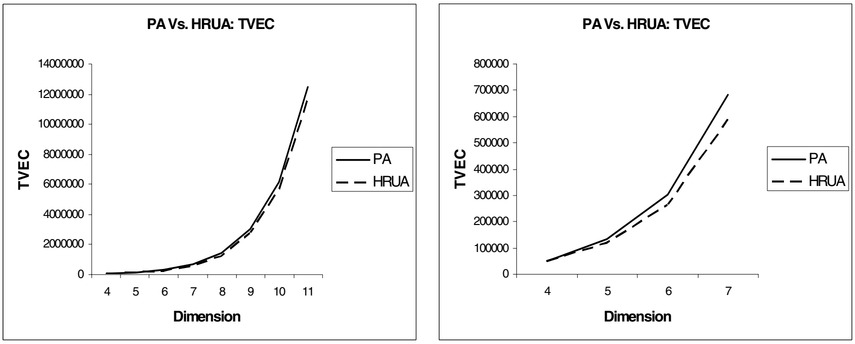


(a) (b)

Fig. 5. TQA - PA Vs. HRUA

In order to ascertain the impact of better TQA, due to PA, on the TVEC, graphs for TQA against number of dimensions were plotted and are shown in Fig. 6. It is evident from the graph (Fig. 6(a)) that the TVEC of PA is slightly more than that of HRUA. This difference is almost negligible for dimensions 4 to 7 as shown in Fig. 6(b). This small difference shows that the PA selects views which are almost similar in quality to those selected by HRUA.

It can be reasonably inferred from the above graphs that PA trades significant improvement in TQA with a slight drop in TVEC of views selected for materialization.



(a) (b)

Fig. 6. TVEC - PA Vs. HRUA

5 Conclusion

In this paper, an algorithm is proposed that greedily selects Top-K views from a multidimensional lattice using views size and query frequency. The algorithm computes the profit of each view, which is defined as a function of the size and query frequency, and then selects from amongst them the most profitable view for materialization. Unlike HRUA, the proposed algorithm is able to select views that are not only profitable with respect to size but are also able to account for large number of queries. The selected views thereby would reduce the average query response time.

Further experiment based comparison between the proposed algorithm and HRUA on parameters TQA and TVEC showed that the proposed algorithm, in comparison to HRUA, was found to achieve significant improvement in TQA at the cost of a slight drop in the TVEC in respect of views selected for materialization. This shows that the proposed algorithm trades significant improvement in total number of queries answered with a slight drop in the quality of views selected for materialization.

References

1. Agrawal, S., Chaudhuri, S., Narasayya, V.: Automated Selection of Materialized Views and Indexes in SQL Databases. In: Proceedings of VLDB 2000, pp. 496–505. Morgan Kaufmann Publishers, San Francisco (2000)
2. Aouiche, K., Darmont, J.: Data mining-based materialized view and index selection in data warehouse. Journal of Intelligent Information Systems, 65–93 (2009)
3. Baralis, E., Paraboschi, S., Teniente, E.: Materialized View Selection in a Multidimensional Database. In: Proceedings of VLDB 1997, pp. 156–165. Morgan Kaufmann Publishers, San Francisco (1997)
4. Chirkova, R., Halevy, A., Suciu, D.: A Formal Perspective on the View Selection Problem. The VLDB Journal 11(3), 216–237 (2002)
5. Gupta, H., Mumick, I.: Selection of Views to Materialize in a Data Warehouse. IEEE Transactions on Knowledge and Data Engineering 17(1), 24–43 (2005)

6. Harinarayan, V., Rajaraman, A., Ullman, J.: Implementing Data Cubes Efficiently. In: Proceedings of SIGMOD 1996, pp. 205–216. ACM Press, New York (1996)
7. Inmon, W.H.: Building the Data Warehouse, 3rd edn. Wiley Dreamtech (2003)
8. Nadeau, T.P., Teorey, T.J.: Achieving scalability in OLAP materialized view selection. In: Proceedings of DOLAP 2002, pp. 28–34. ACM, New York (2002)
9. Roussopoulos, N.: Materialized Views and Data Warehouse. In: 4th Workshop KRDB 1997, Athens, Greece (August 1997)
10. Serna-Encinas, M.T., Hoya-Montano, J.A.: Algorithm for selection of materialized views: based on a costs model. In: Proceeding of Eighth International Conference on Current Trends in Computer Science, pp. 18–24 (2007)
11. Shah, A., Ramachandran, K., Raghavan, V.: A Hybrid Approach for Data Warehouse View Selection. Int. Journal of Data Warehousing and Mining 2(2), 1–37 (2006)
12. Teschke, M., Ulbrich, A.: Using Materialized Views to Speed Up Data Warehousing. Technical Report, IMMD 6, Universität Erlangen-Nümberg (1997)
13. Vijay Kumar, T.V., Ghoshal, A.: A Reduced Lattice Greedy Algorithm for Selecting Materialized Views. In: CCIS, vol. 31, pp. 6–18. Springer, Heidelberg
14. Vijay Kumar, T.V., Haider, M., Kumar, S.: Proposing Candidate Views for Materialization. In: CCIS, vol. 54, pp. 89–98. Springer, Heidelberg (2010)