Methods for Evaluating Iceberg Queries

A.Padmapriya,M.C.A.,M.Phil.,Ph.D
Department of Computer Science and Engineering
Alagappa University
Karaikudi

T.Shanmuga Priya
Research scholar
Department of Computer Science and Engineering
Alagappa University,Karaikudi

ABSTRACT
Iceberg queries are a special case of SQL queries involving GROUP BY and HAVING clauses, wherein the answer set is small relative to the database size. Iceberg queries have been recently identified as important queries for many applications. Queries can be characterized by their huge input-small output. The iceberg refers to the input, and the tip of it refers to the output. This paper is going to present some of the existing iceberg query processing using data mining.

Keywords-Iceberg Query, Counting co-occurrence, Bitmap index.

1. INTRODUCTION
Data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks.

1.1 Uses of data mining
Data mining brings a lot of benefits to businesses, society, governments, sales, marketing, insurance, health care, transportation and medicine and so on.

- ***Market segmentation:*** Identify the common characteristics of customers who buy the same products from your company.
- ***Fraud detection*****:** Identify which transactions are most likely to be fraudulent.
- ***Direct marketing*****:** Identify which prospects should be included in a mailing list to obtain the highest response rate.
- Banking/Finance: Used to identify customer loyalty by analyzing the data of customer purchasing activities.
- ***Interactive marketing*****:** Predict what each individual accessing a Web site is most likely interested in seeing.
- ***Health Care and Insurance*****:** The growth of insurance industry entirely depends on the ability of converting data into the knowledge, information or intelligence about customer, competitors and its markets.
- ***Market basket analysis*****:** Understand what products or services are commonly purchased together; e.g., beer and diapers [29] [30].

1.2 Data mining Relationship
Data mining consists of any of four types of relationships are sought:

- ***Classes:*** Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- ***Clusters****:* Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- ***Associations:*** Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- ***Sequential patterns:*** Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Data mining consists of Different level of analysis are available such as artificial neural networks, genetic algorithms, decision trees, nearest neighbor method, rule induction, data visualization.

1.3 Data mining Activities
Data mining consists of five activities provide the information needed to create the data base. Data mining activities are given below

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.
- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table.

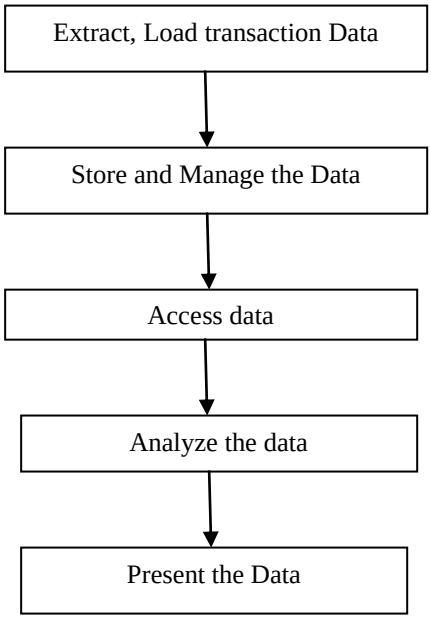

Figure 1: Data Mining Activities

Figure 1 shows the activities to be carried out in data mining applications.

2. BACK GROUND

2.1 *Data Cubes*

The cube is used to represent data along some measure of interest. Although called a "cube", it can be 2-dimensional, 3-dimensional, or higher-dimensional. Each dimension represents some attribute in the database and the cells in the data cube represent the measure of interest. For example, they could contain a count for the number of times that attribute combination occurs in the database, or the minimum, maximum, sum or average value of some attribute. Queries are performed on the cube to retrieve decision support information.

Recently, [14] introduced the CUBE operator for conveniently supporting multiple aggregates in OLAP database. The CUBE operator is the n-dimensional generation of group-by operator. It computes group-by corresponding to all possible combinations of a list of attributes.

The CUBE operator as follows:
SELECT P, D, C, SUM(S)
From Transaction CUBE-BY P, D, C.

2.2 Counting Co-occurrences

Data mining consists of two types of Counting Co-occurrences for given below

- Frequent Item sets.
- Iceberg Queries.

2.2.1 Frequent Item sets

A market basket is a collection of items purchased by a customer in a single customer transaction. Identify items that are purchased together. Every subset of a frequent item set must also be a frequent itemset.Itemset: a set of items. Support of an item set: the fraction of transactions in the database that contain all items in the item set.

2.2.2 Iceberg Queries

This paper presents efficient ways for executing iceberg queries. An iceberg query computes an aggregate function over an attribute (or set of attributes) in order to find aggregate values above a specified threshold. AGG represents an aggregation function. Iceberg query can have other aggregation function such as COUNT, MIN, MAX and SUM.

The prototypical iceberg query the paper considers (can be easily extended to the other forms of iceberg queries) is:

```
SELECT     attr1, attr2, …., attrk, COUNT ( rest)
FROM       R
GROUP BY attr1, attr2, attrk
HAVING     COUNT (rest) >= T
```

Where R is a relation that contains attributes attr1, attr2, ..., attrk, rest and T is a threshold.

3. SURVEY ON ICEBERG QUERIES

3.1 Iceberg Query Application

The relational database system like ORACLE, SQL SERVER, and MYSQL are using general aggregation algorithms [12] [31] to answer the iceberg queries. Many practical application including data warehousing [1], market-basket analyses [29] rely on iceberg query.

3.2 Iceberg CUBE

Iceberg queries were Introduced in [22] and iceberg CUBE problem introduced in [18].The recent research [22] [18] has paid attention to iceberg problem. Iceberg problem in database means relation between a lot of data and few results is similar to it between an iceberg and the tip of one.

Recently, [17] a variant of the problem, called iceberg data cube computation was introduced by BUC. In order to meet similar objectives, in [16] proposed "multifeature cubes". When computing such cubes, aggregates not satisfying a selection condition specified by user (similar to the clause having in SQL) are discarded.

In [24] proposed an approach for computing a condensed representation of either full or iceberg data cubes. Author introduced a novel and sound characterization of data cubes based on dimensional-measurable partitions. Such partitions have an attractive advantage: avoiding sorting techniques which are replaced by a linear product of dimensional-measurable partitions. Account the critical problem of memory limitation. In [15] proposed analytical and experimental performance study shows that APIC and BUC are promising candidates for scalable computation and the best efficiency of APIC.

3.3 Pruning Techniques

From the previous works, it is known that static index pruning techniques can reduce the size of an index (and the underlying collection) while providing comparative effectiveness performance with that of the unpruned case [3, 5].

This pruning strategy is extended to handle complex measures, including averages, in [14].Pruning the lattice that has to be computed. A bottom-up approach to computing the iceberg cube

using the Apriori technique [23] for pruning is proposed in [18]. Iceberg queries which returns frequently occurring values from a set of attributes. It uses a set of estimates of quantiles of the input relation to focus the search of the query result.

3.4 Structured and Text data

In author presented a strategy to efficiently answer joint queries on both structured and text types of data. The records in data warehouses are usually extracted from other database systems and therefore contain only what is known as structured data [6, 7, 28]. A large amount of text document is inadequate for processing efficiently joint queries over structured and text data.

In [10], a scheme for providing quick approximate answers to the iceberg query is devised with the intention of helping the user refine the threshold before issuing the "final" iceberg query with the appropriate threshold. That is, it tries to eliminate the need of a domain expert or histogram statistics to decide whether the query will actually return the desired "tip" of the iceberg. This strategy for coming up with the right threshold is complementary to the efficient processing of iceberg queries.

3.5 Multiple base Relations

In general, these strategies appear wasteful since they do not take the threshold predicate into account, that is, they are not output sensitive. In case of an iceberg query involving a join of multiple base relations, the iceberg relation ***I*** is derived from the base relations ***B*** using one of the efficient join algorithms: sort-merge join, hybrid-hash join, and others mentioned in [12]. For the case where the group-by clause shares some attributes with the join attributes, the query optimizer opts for algorithms that produce "interesting" orders ([12],[27]). As a result of this, the tuples from the result of the join can be piped to the following aggregate operation, which can then aggregate the tuples in memory to produce the final query result.

3.6 Relational algebra

For characterizing cuboids, author state an equivalence between our representation and the result of the aggregate formation defined by [2] which is chosen because it is on one hand the original definition of the aggregation operator in the relational algebra [20][21].

3.7 N-iceberg queries

The number of tuples satisfying the query is very less compared to the size of the database,[19] coin the term N-iceberg(Negative) queries for such a type of queries [20] proposed an algorithm to evaluate N-iceberg queries and compare them with ORACLE and traditional sorting algorithms, with very little main memory.

With the rapid increase of the databases and data repositories sizes, new types of queries have been emerged where the output is significantly small compared to the input. Iceberg queries have been recently identified as important queries for many applications belonging to this category. These applications can be found in data mining [1],information retrieval [26], decision support and data warehouse [4], web mining and top k queries [8, 9]. The iceberg queries are formally introduced by Fang et al. [11]. Detailed application examples have been also presented in [12]. These queries have been extended to data cubes in [4].

3.8 Bitmap indices

Today's bitmap indices can be applied on all types of attributes.Studes have shown that compressed bitmap indices occupy less space than the raw data. Bitmap are provides better query performance. Nowadays bitmap index is supported in many commercial database systems (e.g., ORACLE, Informix) and so on. A bit map index is a data structure used to efficiently access large database.Generally,the purpose of an index is to provide pointers to row in a table containing given key values. In a common index, this is achieved by storing a list of records for each key corresponding to the row with that key value.

Partitioning algorithms to handle iceberg queries with AVERAGE aggregate function have been proposed in [13].Using PC clusters for parallelizing the computation of the iceberg-cube is investigated in [25][26].

4. OBJECTIVIES

The main objective of Iceberg queries is to retrieve data quickly. Query optimization is the refining process in database administration and it help bring down speed of execution. Data mining techniques are often measured by their speed. The reason behind this that the faster the tool can run and the larger the data set which it can be applied. Iceberg queries are generally very expensive to compute since they require several scans of relations.

The common objectives for any iceberg query is as mentioned below

- Speed of execution
- Ability to process large data set
- Reduce the number of scans in data set

5. CONCLUSION

This paper gives brief introduction about data mining used of data mining and its activities. This paper presents a detailed survey on the existing most significant information about the evaluation of iceberg queries, the need for ice berg queries and algorithm employed for evaluation of iceberg queries. The objectives of iceberg queries are studied in this paper. This gives us the future direction to work on efficient evaluation of iceberg queries.

6. REFERENCES

[1] Agrawal, R. and Srikant, R. "Fast Algorithms for Mining Association Rules." Proceedings of the 20th Int'l Conference on Very Large s Databases (VLDB '94), September 1994.

A. C. Klug. "Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions". Journal of ACM, 29(3):699–717, 1982.

[2] Altingovde, I. S., Ozcan, R., Ulusoy, Ö.: "Exploiting query views for static index pruning in web search engines". In: Proc. of CIKM'09. (2009) 1951-1954

[3] Beyer, K. and Ramakrishnan, R. "Bottom-up Computation of Sparse and Iceberg CUBEs." Proceedings of 1999 ACM SIGMOD Int'l Conference on Management of Data, pp. 359-370, 1999.

[4] Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. S., Soffer, A.," Static index pruning for information retrieval systems. In: Proc. of SIGIR'01. (2001)

[5] Comer, D.: The ubiquitous B-tree. Computing Surveys 11(2), 121–137 (1979)

[6] Chaudhuri, S., Dayal, U.: "An Overview of Data Warehousing and OLAP Technology". ACM SIGMOD Record 26(1), 65–74 (1997)

[7] Chaudhuri, S. and Gravano, L. "Evaluating Top-A: Selection Queries." Proceedings of the 25th Int'l s on Very Large Databases (VLDB '99), pp. 399-410, 1999.

[8] Donjerkovic, D. and Ramakrishnan, R. "Probabilistic Optimization of Top n Queries." Proceedings of the 25th Int'l Conference on Very Large Databases (VLDB'99), pp. 411-422, 1999.

[9] E. Segal, Y. Matias and P. Gibbons, "Online Iceberg Queries".

[10] Fang, M., Shivakumar, N., Garcia-Molina, H., Motwani, R. and Ullman, J. "Computing Iceberg Queries Efficiently." Proceedings of the 24th Int'l Conference on Very Large Databases (VLDB '98), 1998.

[11] G. Graefe, "Query Evaluation Techniques for Large Databases", ACM Comput. Surv., 25, 2, 73–170, June 1993.

[12] J. Bae and S.Lee, "Partitioning Algorithms for the Computation of Average Iceberg Queries", DAWAK,2000.

[13] J. Han et al., "Efficient Computation of Iceberg Cubes with Complex Measures", Proc. of ACM SIGMOD Conf., 2000.

[14] K.A. Ross and D. Srivastava." Fast Computation of Sparse Datacubes". In VLDB'97, Athens, Greece, pages 116–125, 1997.1

[15] K.A. Ross, D. Srivastava, and D. Chatziantoniou." Complex Aggregation at Mutiple Granularities". In EDBT'98, LNCS vol. 1377, pages 263–277. Springer Verlag, 1998.

[16] Kevin S. Beyer and Raghu Ramakrishnan "Bottom-up computation of sparse and iceberg cubes". In Proc. of the Int. Conf. on Management of Data (ACM SIGMOD),pages 359-370, 1999.

[17] K.Beyer and R.Ramakrishnan,"Bottom-Up Computation of sparse and iceberg CUBEs",In Proc.of the ACM SIGMOD Conf.,Pages 359-370,1999.

[18] Leela krishna poola"Efficiently evaluating N-iceberg queries".

[19] L. Cabibbo and R. Torlone. "A Framework for the Investigation of Aggregate Functions in Database Queries". In C. Beeri and P. Buneman, editors, ICDT'99, Jerusalem, Israel, LNCS vol. 1540, pages 383–397.

[20] L. Libkin, L. Cabibbo" the aggregation operator in the relational algebra ".Springer Verlag, 1999.

[21] L. Libkin. Expressive Power of SQL. In ICDT'01, London, UK, LNCS vol. 1973, pages 1–21. Springer Verlag, January 2001.

[22] M.Fang,N.Shivakumar,H.Garua-Molina,R.Motwani,and J.D.Ullam,"Computing iceberg queries Efficiently",In Proc.of 24th VLDB conf..,Pages 299-310,1998.

[23] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules ", Proc. of 20th Intl. Conf. On Very Large Data Bases, 1994.

[24] Rosine C ICCHETTI, Noël N OVELLI, Lotfi L AKHAL LIM "APIC: An Efficient Algorithm for Computing Iceberg Datacubes", CNRS FRE-2246 - Université de la Méditerranée, Case 901

[25] R. Ng, A. Wagner and Y. Yin, "Iceberg-cube Computation with PC Clusters", Proc. of ACM SIGMOD Conf., 2000.

[26] Salton, G. "A Theory of Indexing." Society for Industrial and Applied Mathematics, 1975.

[27] Selinger et al., "Access Path Selection in a Relational Database Management System", Proc. of ACM SIGMOD Conf., 1979.

[28] Shoshani, A.: "OLAP and statistical databases: similarities and differences. In: Principles Of Database Systems (PODS)", pp. 185–196 (1997)

[29] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur".Dynamic itemset counting and implication rules for market basket data". In Proc. of the Int. Conf. on Management of Data (ACM SIGMOD), pages 255-264, 1997.

[30] W. P. Yan and Larson, "Data Reduction through EarlyGrouping", In CASCON, page 74, 1994.