APPROXIMATE QUERY ANSWERS FOR MOBILE DECISION MAKING

Francis A. Méndez Mediavilla, Texas State University, 601 University Dr., San Marcos, TX 78666, 512-245-3303, fm16@txstate.edu
Hsun-Ming Lee, Texas State University, Texas State University, 601 University Dr., San Marcos, TX 78666, 512-245-3225, sl20@txstate.edu
James R. Cook, Texas State University, Texas State University, 601 University Dr., San Marcos, TX 78666, 512-245-3181, jc09@txstate.edu

ABSTRACT

With the advent of mobile computing devices, end-users may obtain valuable information from data warehouses, regardless of their physical locations. Nonetheless, connection availability is usually not guaranteed by the mobile network providers and it has been argued that the maintenance of constant quality OLAP services for mobile users is difficult. A mobile data warehouse system that uses approximate query answers is proposed. As the mobile users request information, a reduced set of data is transmitted from the data warehouse to the mobile computers. Approximate answers to OLAP queries can be obtained locally from the reduced data set. The goals are to reduce the transmission cost, reduce the load of requests on the data warehouse; the mobile device would achieve a certain degree of independence from the data warehouse.

INTRODUCTION

The increased popularity of hand-held computers, as well as the availability of light, but, yet, powerful laptop computers will make mobile computers the preferred front-end for hosting sophisticated decision-making applications [31]. In enterprise computing, a large amount of business data are stored in several data warehouses and the end-users issue OLAP (On-Line Analytical Processing) queries to retrieve suitable information through a network. However, maintaining efficient and consistent data access in mobile environments is a challenge because of weak connectivity and resource constraints [19]. Some difficulties with mobile clients in decision-making information systems include: connection unreliability, limited bandwidth, and limited storage space.

Two techniques suggested to improve mobile query performance are: 1) discovery and prediction of query patterns; and 2) pre-fetching information (cache). Future queries can be predicted based on patterns observed in the past. These observed patterns will determine the data that need to be downloaded to the mobile device in order to answer future queries. To improve mobile access to data in remote databases, part of the data is stored in the mobile devices. Cache mechanisms have been developed to improve query performance and the utilization of storage space [18, 33]. This research proposes a new approach that employs approximate queries as an alternative, especially suitable for multiple data warehouses. As detailed later, the queries return to the end-users statistical estimates for quicker responses; and a reduced set of data are stored in the mobile computers for later expansion to an approximation of the source data cube (from the data warehouse) without going through the network.

APPROXIMATE QUERIES

Multidimensional data models facilitate the generation of summarized information. Typically, a star schema is used to store the multidimensional data. The data consolidation is implemented using a data cube operator [4, 6, 8, 10, 15, 16, 22-24, 27, 28]. The set of detailed data to be stored is minimized, taking into consideration the derivation dependencies. A detailed summary cube can always be used to derive any subset of the original cube. To minimize data storage, the objective is to select the appropriate data set to materialize [21, 30]. When an OLAP query is issued, it must be resolved either locally (mobile device) or remotely (data warehouse). The costs involved in such a transaction include: communication, disk I/O, and computational cost. If a query is processed locally, the performance of the system improves, since communication cost is reduced. When the data are obtained from a remote site, the cost of local disk I/O is reduced, but additional transmission cost is incurred [18].

An approach to obtain approximate query answers locally is proposed. For some practical purposes a quick approximate query answer is sufficient [1, 4, 14, 25, 32]. The goal of an approximate query is to provide an estimated response in orders of magnitude in less time than the time to compute an exact answer, by avoiding or minimizing the number of accesses to the data source [14]. Most decision support queries (OLAP) involve some kind of aggregation of data and that summarization is widely implemented in data warehouses in general. Under the traditional data warehouse setup, exact answers to queries are obtained directly from the data warehouse. Several methods have been proposed to approximate query answers. For instance, a set up is proposed to obtain quick approximate answers to queries from an approximate query engine (AQE) [14]. The AQE maintains various summary statistics denoted as *synopses*. Synopses should carry enough information about the data in a concise representation [3]. Samples are obtained from the data set of interest; summary measures are computed; and, future queries are answered from these pre-computed summaries. Online aggregation is a method based on sequential sampling rather than on pre-computed summaries [17]. In this approach, the idea is to scan the base-data randomly at query time. The approximate answer for an aggregation query is updated as the scan proceeds until the desired minimum accuracy is obtained. Another approach to obtain approximate query answers is to estimate the original detailed data from stored summaries by maximizing an entropy measure and smoothing the estimate using a linear regularization [4, 11]. A statistical model is used to describe parts of the cube. The entire cube is reconstructed from a small set of model parameters. The iterative proportional fitting (IPF) algorithm has been proposed for approximating data cubes: [9]. The advantage of modeling the data using log linear models over the other techniques has also been discussed [5, 20, 25].

MOBILE DATA WAREHOUSES

In traditional systems, a query is constructed in the mobile device and transmitted to the query optimizer. The query optimizer prioritizes the query. The data warehouse is scanned and the query result is then transmitted to the mobile device (see Figure 1). The approach that is proposed in this paper addresses the problems of transmission cost, network unreliability, and overburden on the data warehouse side and data storage on the mobile device. The use of approximate data cubes could improve the performance of cache management for mobile data warehouse systems.

The data warehouse has been summarized at different levels of granularity, across several variables and stored as data cubes (see Figure 2). Remote data warehouses store raw data and all possible aggregated data. For instance, the Essbase system materializes all possible summary tables [18]. Once a data cube is constructed in the data warehouse, the data cube is fitted using a log linear model. A set of sufficient

statistics is determined for each data cube. In a log linear model, a set of sufficient statistics corresponds to a set of marginal sums of the data cube.

Figure 1: Traditional Mobile System

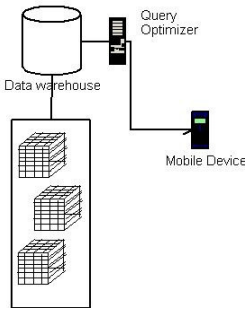
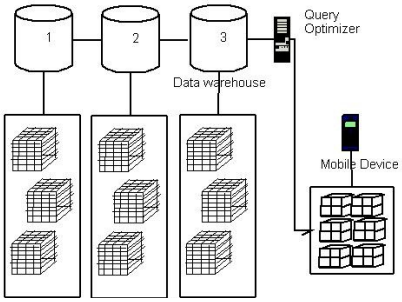


Figure 2: Mobile System



This set of sufficient statistics is kept and maintained in the data warehouse together with the table summaries. When a mobile device requests information, the data cube that contains the data to answer the query is selected. Instead of materializing the data cube, the corresponding set of sufficient statistics is transmitted from the data warehouse to the mobile device. Once in the mobile device, the aggregate materialized views also exist in the form of a reduced set of data cubes. An algorithm can be used to expand these sufficient statistics into an approximate data cube. Approximate answers to OLAP queries can be obtained locally from these reduced data cubes. If the query pattern of the user is determined, this information can be used to determine which data cubes should be selected for transmission. Most query requests are followed by subsequent requests for information that involve some modification of the initial query (i.e. widening the scope of the query, a drill-down operation, a roll-up operation). The reduced data cubes necessary to support subsequent operations can be downloaded accordingly. With these data cubes stored locally, further querying can be supported without having to access the data warehouse. This provides the mobile device a certain degree of independence from the data warehouse since the materialized data cubes will be able to support queries of greater detail than the initial query.

MODELING DATA CUBES

Two approaches to fit log linear models are explored: 1) maximum likelihood estimates [9]; and Bayesian posterior estimates: [12, 20, 29]. The iterative proportional fitting (IPF) algorithm or Deming-Stephan algorithm is the standard method to obtain maximum likelihood estimates (MLEs) for cell counts, assuming a log linear model [2, 7, 9]. Fitting a log linear model results in a set of sufficient configurations (i.e. C_{12}, C_{13}, C_{23}; where a sufficient configuration C_{12} indicates two sets of summary counts: marginal counts for variable 1 and marginal counts for variable 2.) that can be used to approximate the elementary cells of a cube. The end-user can set parameters to specify which models are acceptable. Models are selected based on how well they fit the observed data and the simplicity of the model. The main reason to assess the goodness of fit of a model is that a good fit implies that the model gives precise estimates of expected frequencies [7]. Traditionally, the goodness of fit of a particular model can be judged using a χ^2 goodness of fit statistic such as Pearson's X^2 and the log likelihood ratio G^2. However, it is known that the effectiveness of the Pearson's X^2 and the log likelihood ratio statistics deteriorates quickly when the data cubes are sparse. Other statistics are recommended to assess the goodness of fit when the data cubes are sparse [20]. Within the framework being proposed, the model selection and model confirmation processes take place in the data warehouse. The Bayesian iterative proportional fitting (BIPF) is a stochastic version of the IPF algorithm. In BIPF,

a Gibbs sampler is applied to the vector of cell probabilities $\{\pi_\theta\}$, where θ is any set of indexes. The goal is to randomly draw cell probabilities from the posterior distribution. This type of approach is known as Markov Chain Monte Carlo (MCMC), since the results are sampled from a certain Markov chain. Several methods to assess the convergence of these chains have been proposed. The least computationally expensive methods are more appealing, since this computation takes place on the mobile device [13]. The Bayesian method is more computationally demanding than the implementation of the IPF and requires generating large sample sizes from the posterior distribution. The advantage of the Bayesian method is that, in addition to providing a point estimate of the elementary cell value, it provides the Bayesian confidence intervals for the elementary cell values.

EVALUATION

The proposed approach is related to the work that adopts various cache management mechanisms for a data warehouse system. Huang et al (2005) built a mobile data warehouse that allows users to pre-fetch appropriate data effectively by combining cache management with a personalized prediction model. In the study of Park et al (2003), cache for OLAP systems is effectively managed based on the usability of query results [26]. However, queries in both systems are failed during network downtime if required data is not in the cache. In our mobile data warehouse, the reduced data cubes are expanded to answer specific queries. This feature reduces the necessity for accesses to the data warehouse and alleviates the storage situation in the mobile device, which is due to the fact that approximate answers can be provided for end-users while the mobile network connection is not available. Moreover, since the sufficient statistics have been saved in the mobile clients locally, the detailed data cubes can be discarded once the query has been answered. The data cube can be expanded again if needed. In order to issue queries to many data warehouses in large enterprise or cross-enterprise environments, the advantage of this data reduction approach becomes obvious. With a relatively small amount of memory, the mobile clients are allowed to execute a sequence of approximate queries to different data warehouses without the dependence of network availability. The approach involves shifting some of the computational burden from the data warehouse (remote) to the mobile device (local). While this means a heavier computational burden on the mobile device, it also means less dependency on the data warehouse and a lesser computational burden on the data warehouse. The expansion of the cube requires local computation resources. Power expenditure on the local site is considered more costly since the usual source of power is a battery. The Bayesian method, in particular, is computationally intensive. In our MCMC implementation, it seems convenient to follow the least costly procedure. It shall be assumed, as Geyer (1992) suggests, that a single sufficiently long chain will suffice to obtain samples from the posterior distribution of the estimates.

CONCLUSION

With mobile computers, decision makers are allowed to obtain valuable information from data warehouses regardless of their physical locations. However, connection availability is usually not guaranteed by mobile network providers and we have argued that the maintenance of constant quality OLAP services for mobile users is difficult. Therefore, we have proposed a mobile data warehouse system that employs approximate queries. Using this system, the mobile users request information; and, then a set of sufficient statistics is transmitted from the data warehouse to the mobile computers. Approximate answers to OLAP queries can be obtained locally from the reduced data set. More importantly, a statistical model is used to describe the data set and the entire cube can be reconstructed without connecting to the remote data warehouses. The proposed system provides a possibility for enterprises to offer effective decision support for mobile users with: 1) Approximate answers by

statistical methods; 2) Better application transparency, which means the OLAP applications are unaware of the user mobility; 3) Lower transmission cost since approximate queries can be executed entirely on the mobile computers; and, 4) Better portability by managing data storage, which allows queries to multiple data warehouses without network availability. Since this study focuses on the application of approximate queries to constructing mobile data warehouses, the memory requirements of the mobile computers is still high. For a query execution, one of reduced data sets for multiple data warehouses may be expanded to a full data cube. In addition, our mobile data warehouses are computationally intensive. Consequently, the proposed system is effective for laptop computers, but is not feasible for other mobile devices with less memory and slower CPU. In the future, the authors will conduct quantitative experiments to evaluate the system performance and investigate techniques to reduce the memory requirements for this mobile data warehouse system.

REFERENCES

[1] Acharya, S., et al., *Join Synopses for Approximate Query Answering.* SIGMOD ACM Press, 1999.

[2] Agresti, A., *Categorical Data Analysis*. 1990, New York: John Wiley & Sons. 558.

[3] Barbará, D., et al., *The New Jersey data reduction report.* Bulletin of the Technical Committee on Data Engineering, 1997. **20**(4).

[4] Barbará, D. and M. Sullivan, *Quasi-Cubes: A space-efficient way to support approximate multidimensional databases*. 1998, School of Information Technology and Engineering, George Mason University: Fairfax, Virginia.

[5] Barbará, D. and X. Wu, *Loglinear-Based Quasi Cubes.* Journal of Intelligent Information Systems, 2001. **16**: p. 255-276.

[6] Beyer, K. and R. Ramakrishnan. *Bottom-up Computation of Sparse and Iceberg CUBE*. in *Proceedings of the 1999 ACM SIGMOD, International Conference on Management of Data*. 1999. Philadelphia, Pennsylvania, United States: ACM Press.

[7] Bishop, Y.M.M., S.E. Fienberg, and P.W. Holland, *Discrete Multivariate Analysis: Theory and Practice*. 1975, Cambridge, Massachusetts: MIT Press. 557.

[8] De Giacomo, G. and M. Lenzerini, *What's in an Aggregate: Foundations for Description Logics with Tuples and Sets*. 1995.

[9] Deming, W.E. and F.F. Stephan, *On a least square adjustment of a sampled frequency table when the expected marginal totals are known.* Annals of Mathematical Statistics, 1940. **11**: p. 427-444.

[10] Deshpande, P.M., et al., *Cubing Algorithms, Storage Estimation, and Storage and Processing Alternatives for OLAP.* Bulleting of the Technical Committee on Data Engineering, 1997. **20**(1): p. 3-11.

[11] Faloutsos, C., H.V. Jagadish, and N.D. Sidiropoulos. *Recovering Information from Summary Data*. in *23rd International Conf. on Very Large Data Bases*. 1997.

[12] Gelman, A., et al., *Bayesian Data Analysis*. 1995, New York: Chapman & Hall/CRC.

[13] Geyer, C.J., *Practical Markov Chain Monte Carlo*, in *Statistical Science*. 1992. p. 473-511.

[14] Gibbons, P.B. and Y. Matias. *New Sampling-Based Summary Statistics for Improving Approximate Query Answers*. in *SIGMOD Conference 1998*. 1998.

[15] Gupta, A., et al. *On the computation of multidimensional aggregates*. in *22nd Intl. Conf. Very Large Data Bases*. 1996. Mumbai (Bombay).

[16] Han, J., et al. *Efficient Computation of Iceberg Cubes with Complex Measures*. in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*. 2001. Santa Barbara, California, United States: ACM Press, New York, NY, USA.

[17] Hellerstein, J.M., P.J. Haas, and H.J. Wang. *Online aggregation*. in *ACM SIGMOD International Conf. on Management Data*. 1997.
[18] Huang, S.M., B. Lin, and Q.S. Deng, *Intelligent Cache Management for Mobile Data Warehouse Systems.* Journal of Database Management, 2005. **16**(2): p. 46-65.
[19] Jing, J., A. Helal, and A. Elmagarmid, *Client-Server Computing in Mobile Environments.* ACM Computing Surveys, 1999. **31**(2): p. 117-157.
[20] Jones, D.H. and F.A. Méndez Mediavilla. *A Bayesian Method for Query Approximation*. in *Northeast Business and Economics Association*. 2006. Long Island, New York.
[21] Kotidis, Y. and N. Roussopoulos, *A Case for Dynamic View Management.* ACM Transactions on Database Systems, 2001. **26**(4): p. 388-423.
[22] Mumick, I.S., D. Quass, and B.S. Mumick, *Maintenance of Data Cubes and Summary Tables in a Warehouse*, in *Materialized Views: Techniques, Implementations, and Applications*, A. Gupta and I.S. Mumick, Editors. 1999, The MIT Press: London. p. 387407.
[23] Muto, S. and M. Kitsuregawa. *A Dynamic Load Balancing Strategy for Parallel Datacube Computation*. in *Proceedings of the 2nd ACM International Workshop on Data Warehousing and OLAP*. 1999. Kansas City, Missouri, United States.
[24] Muto, S. and M. Kitsuregawa. *Improving main memory utilization for array-based datacube computation*. in *Proceedings of the 1st ACM International Workshop on Data Warehousing and OLAP*. 1998. Washington, D.C., United States.
[25] Palpanas, T. and N. Koudas. *Entropy Based Approximate Querying and Exploration of Datacubes*. in *Proceedings of the 13th International Conference on Scientific and Statistical Database Management*. 2001. George Mason University, Fairfax, Virginia, USA: IEEE Computer Society.
[26] Park, C.-S., M.H. Kim, and Y.-J. Lee, *Usability-based caching of query results in OLAP systems.* The Journal of Systems and Software, 2003. **68**: p. 103-119.
[27] Ramakrishnan, R. and J. Gehrke, *Database Management Systems*. 2000, New York: McGraw Hill.
[28] Ross, K.A. and D. Srivastava. *Fast Computation of Sparse Datacubes Source*. in *Proceedings of the 23rd International Conference on Very Large Data Bases*. 1997. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
[29] Schafer, J.L., *Analysis of Incomplete Multivariate Data*. 1997, Boca Raton, Florida: Chapman & Hall/CRC. 430.
[30] Sharaf, M.A. and P.K. Chrysanthis. *Facilitating Mobile Decision Making*. in *2nd ACM International Workshop on Mobile Commerce (WMC)*. 2002. Atlanta, Georgia: ACM.
[31] Sharaf, M.A. and P.K. Chrysanthis, *On-Demand Data Broadcasting for Mobile Decision Making.* Mobile Networks and Applications, 2004. **9**: p. 703-714.
[32] Vitter, J.S. and M. Wang. *Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets*. in *SIGMOD International Conference on Management of Data*. 1999. Philadelphia, Pennsylvania, USA: ACM Press.
[33] Wolfson, O., et al., *View Maintenance in Mobile Computing*, in *SIGMOD RECORD*. 1995, Association of Computing Machinery Press.