

APPROXIMATE QUERY PROCESSING

Qing Liu

Tasmanian ICT Centre
CSIRO, Australia
{Q.Liu}@csiro.au

SYNONYMS

Approximate Query Answering

DEFINITION

Query processing in a database context is the process that deduces information that is available in the database. Due to the huge amount of data available, one of the main issues of query processing is how to process queries efficiently. In many cases, it is impossible or too expensive for users to get exact answers in the short query response time. Approximate query processing (AQP) is an alternative way that returns approximate answer using information which is similar to the one from which the query would be answered. It is designed primarily for aggregate queries such as count, sum and avg etc. Given a SQL aggregate query Q , the accurate answer is y while the approximate answer is y' . The relative error of query Q can be quantified as:

$$Error(Q) = \left| \frac{y - y'}{y} \right| \quad (1)$$

The goal of approximate query processing is to provide approximate answers with acceptable accuracy in orders of magnitude less query response time than that for the exact query processing.

HISTORICAL BACKGROUND

The earliest work on approximate answers to decision support queries appears in Morgenstein's dissertation from Berkeley [1]. And the approximate query processing problem has been studied extensively in the last ten years. The main motivations [9] which drive the techniques being developed are summarized as follows:

First, with the advanced data collection and management technologies, nowadays there are a large number of applications with data sets about gigabytes, terabytes or even petabytes. Such massive data sets necessarily reside on disks or tapes, making even a few accesses of the base data sets comparably slow. In many cases, precision to "last decimal" is not required for a query answer. Quick approximation with some error guarantee (e.g. the resident population in Australia 21,126,700 +/- 200) is adequate to provide insights about the data.

Second, decision support system (DSS) and data mining are popular approaches to analyzing large databases for decision making. The main characteristic of the DSS is that aggregation queries (e.g. count, sum, avg etc.) are executed on large portion of the databases, which can be very expensive and resource intensive even for a single analysis query. Due to the exploratory nature of decision making, iterative process involves multiple query attempts. Approximate answers with fast response time gives users the ability to focus on their explorations and quickly identify truly interesting data. It provides a great scalability of the decision support applications.

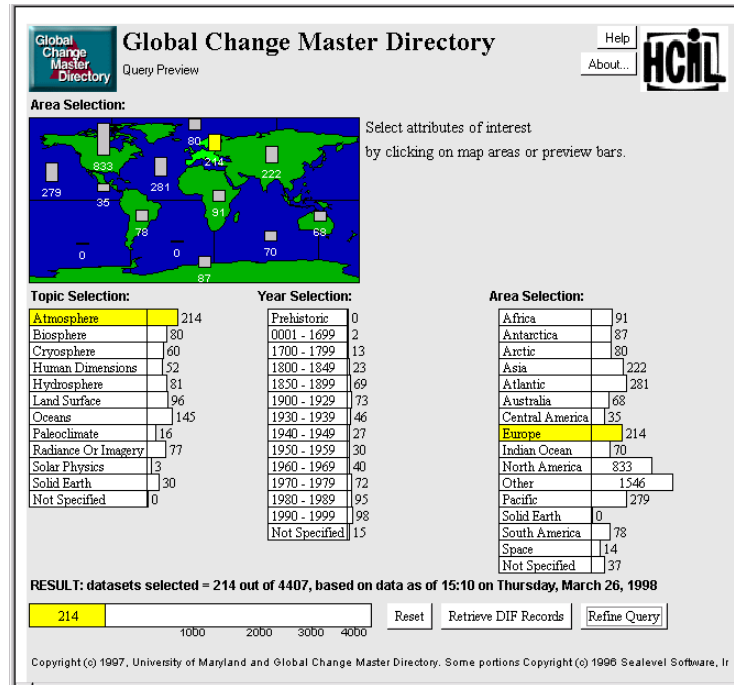


Fig. 1. NASA EOSDIS Interface of Query Preview

Third, approximate query processing is also used to provide query preview. In most cases, users are only interested in a subset of the entire database. Given a trial query, query preview provides an overview about the data distribution. The users can preview the number of hits and refine the queries accordingly. This prevents users from fruitless queries such as zero-hits or mega-hits. Figure 1 shows an example of query preview interface of NASA EOSDIS (Earth Observing System Data and Information System) project, which is attempting to provide online access to a rapidly growing archive of scientific earth data about the earth's land, water, and air. In the query preview, users select rough ranges for three attributes: area, topic (a menu list of parameters such as atmosphere, land surface, or oceans) and temporal coverage. The number of data sets for

each topic, year, and area is shown on preview bars. The result preview bar, at the bottom of the interface, displays the total approximate number of data sets which satisfy the query.

Finally, sometimes network limitation or disk storage failure would cause the exact answers unaffordable or unavailable. An alternative solution is to provide an approximate answer based on the local cached data synopsis.

SCIENTIFIC FUNDAMENTALS

Due to the acceptability of approximate answers coupled with the necessity for quick query response time, approximate query processing has emerged as a cost effective approach for dealing with the large amount of data. This speed-up is achieved by answering queries based on samples or other synopses (summary) of data whose size is orders of magnitude smaller than that of the original data sets.

Ioannidis presented the generic flow of approximate query process (Figure 2) in [2]. Here, data analysis is the overall approach which derives the synopsis from original data to approximate the underlying data distribution. Typically, the **algorithm** partitions the data based on the **distance** function into groups of similar elements, called **buckets**, clusters, patterns, or several other names. The **data elements** that fall in each bucket are then represented by the synopses for **approximation use** which corresponds to the actual purpose of the whole data approximation process. The quality of approximation result can be measured by the **distance** between the synopses and the original data.

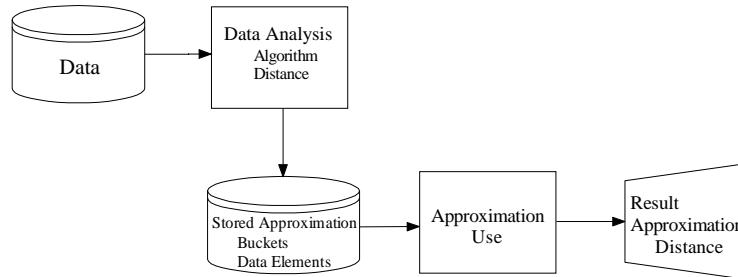


Fig. 2. Generic Flow of Approximation Process

There are two basic approaches to achieve approximate query processing: pre-computed synopsis and online query processing.

Pre-computed synopsis

Approximate query processing using pre-computed synopsis includes two steps: construct synopsis prior to query time, answer query approximately using synopsis at query time.

To provide high accurate query answer, the key issue to construct synopsis is how to represent the underlying data distribution precisely and compactly. Generally, the data distribution can be classified into two groups: uniform distribution and

non-uniform distribution. The synopsis for uniform data distribution assumes the objects are distributed uniformly in the data space. For point objects locating in 2 dimensional space $[U_{min}^x, U_{min}^y] \times [U_{max}^x, U_{max}^y]$, the query result size is estimated as $N \times area(Q) / ((U_{max}^x - U_{min}^x) \times (U_{max}^y - U_{min}^y))$, where N is the data set size and $area(Q)$ is the area of window query Q .

There are various techniques developed for non-uniform data distribution. They can also be divided into two groups: parametric and non-parametric. The parametric techniques try to use parameters to catch the original data distributions. Although the models can summarize data distributions with a few descriptive parameters, if the underlying data do not follow any known distributions, or their linear combinations, the model fitting techniques produce inferior results.

The non-parametric techniques use different approaches to summarize the data distributions. Generally we can classify these techniques into 3 categories according to the strategies adopted:

- Sampling techniques
- Histogram techniques
- Wavelet techniques

Sampling The basic idea of sampling is that a small random sample of the data often well-represent all the data. Therefore, query would be answered based on the pre-sampled small amount of data and then scaled up based on the sample rate. Figure 3 shows an example that 50% of data are sampled during the pre-computed stage. Given a query "how many Sony laptops are sold in R ", the approximate result is "select 2 * sum(*) from S where $S.product = 'SonyLaptop'$ ", which is 12. In R , the exact answer is 11.

R: Sales	
Product	Amount
Sony Laptop	1
Sony Laptop	1
Sony Laptop	2
Sony Laptop	3
Sony Laptop	4
Dell Printer	1
Dell Printer	2
LCD Monitor	1

S: Sampled Sales	
Product	Amount
Sony Laptop	1
Sony Laptop	2
Sony Laptop	3
Dell Printer	2

Fig. 3. Example of Sampling

The main issue of sampling method is to decide what sample criteria should be used to select data. The sampling techniques are classified into the following groups [5]:

- Uniform sampling: data is sampled uniformly
- Biased sampling: a non-uniform random sample is pre-computed such that parts of the database deemed more important than the rest
- Icicles: a biased sampling technique that is based on known workload information
- Outlier indexing: indexing outliers and biased sampling the remaining data
- Congressional sampling: targeting group by queries with aggregation and trying to maximize the accuracy for all groups (large or small) in each group-by query
- Stratified sampling: generalization of outlier indexing, Icicles and congressional sampling. It targets minimizing error in estimation of aggregates for the given workload

Sample-based procedures are robust in the presence of correlated and nonuniform data. Most importantly, sampling-based procedures permit both assessment and control of estimation errors. The main disadvantage of this approach is the overhead it adds to query optimization. Further more, join operation could lead to significant quality degradations because join operator applied on two uniform random sample can result in a non-uniform sample of the join result which contains very few tuples.

Histograms Histogram techniques are the most commonly used form of statistics in practice (e.g. they are used in DB2, Oracle and Microsoft SQL Server). It is because they incur almost no run-time overhead and produce low-error estimates while occupying reasonably small space.

The basic idea is to partition attribute value domain into a set of buckets and query is answered based on the buckets. The main issues of histogram construction and query are as follows:

- How to partition data into bucket
- How to represent data in each bucket
- How to estimate answer using the histogram

For 1 dimensional space, a histogram on an attribute X is constructed by partitioning the data distribution of X into B ($B \geq 1$) buckets and approximating the frequencies and values in each bucket. Figure 4(a) is an example of original data set and Figure 4(b) shows its data distribution. Figure 4(c) is an example of histogram constructed accordingly, where $B = 3$.

If there are several attributes involved in a query, a multi-dimensional histogram is needed to approximate the data distribution and answer such a query. A multi-dimensional histogram on a set of attributes is constructed by partitioning the joint data distribution of the attributes. They have the exact same characteristics as 1 dimensional histograms, except that the partition rule needs to be more intricate and cannot always be clearly analyzed because there can be no ordering in multiple dimensions [3].

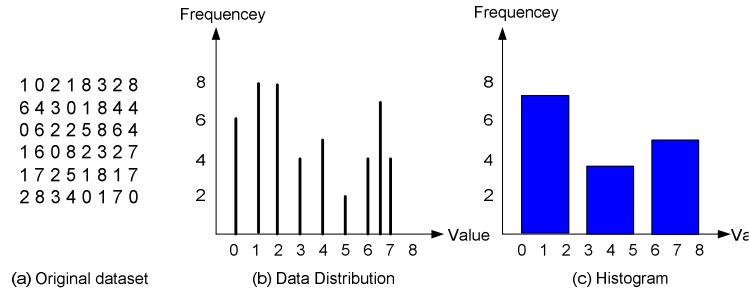


Fig. 4. Example of Histogram

To represent data in each bucket, it includes value approximation and frequency approximation. Value approximation captures how attribute values are approximated within a bucket. And frequency approximation captures how frequencies are approximated within a bucket.

The two main approaches for value approximation are *continuous value assumption* and *uniform spread assumption* [4]. Continuous value assumption only maintains min and max value without indication of how many values there are or where they might be. Under the uniform spread assumption, one also maintain the number of values within each bucket and approximates the actual value set by the set that is formed by (virtually) placing the same number of values at equal distances between the min and max value in multi-dimensional space [10].

With respect to frequency approximation, almost all work deal with *uniform distribution assumption*.

The benefit of a histogram synopsis is that it can be easily used to answer many query types, including the aggregate and non-aggregate queries. However, one of the issues of histogram approach is it is hard to calculate a theoretical error bound. Thus the evaluations on the histogram synopsis usually rely heavily on the experiment results. Further more, histogram-based approaches become problematic when dealing with the high-dimensional data sets that are typical for modern decision support applications. This is because as the dimensionality of the data increases, both the storage overhead (i.e., number of buckets) and the construction cost of histograms that can achieve reasonable error rates increase in an explosive manner.

Wavelet Wavelet is a mathematical tool for hierarchical decomposition of functions using recursive pairwise averaging and differencing at different resolutions. It represents a function in terms of a coarse overall shape, plus details that range from broad to narrow. It is widely used in the signal and image processing.

Matias et al. [6] first proposed the use of Haar-wavelet coefficients as synopsis for estimating the selectivity of window queries. The basic idea is to apply wavelet decomposition to the input data collection to obtain a compact data synopsis that comprises a select small collection of wavelet coefficients. Figure 5 shows an example of hierarchical decomposition tree of Haar-wavelet. The leaf nodes are

original data and non-leaf nodes are wavelet coefficients generated by averaging and differencing from their two children.

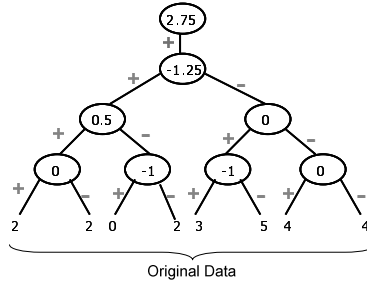


Fig. 5. Example of Haar-wavelet Hierarchical Decomposition

Later, the wavelet concept was extended to answer more general approximate queries. The results of recent studies have clearly shown that wavelets can be very effective in handling aggregates over high-dimensional online analytical processing (OLAP) cubes, while avoiding the high construction costs and storage overheads of histogram techniques.

Another important part of the above three technologies for approximate query processing is synopsis maintenance. If the data distribution is not changed significantly, the data synopsis would be updated accordingly to reflect such change. Otherwise, a new data synopsis will be constructed and discard the old one. Please refer to the specific techniques for more details.

There are a few other work that do not belong to the above 3 categories to approximate the underlying data distribution. For example, recently Das et al. [7] present a framework that is based on randomized projections. This is the first work in the context of spatial database which provides probability quality guarantees with respect to query result size approximation.

Online query processing

The motivation of online query processing is that the data analysis is to extract unknown information from data. It is an iterative process starting by asking broad questions and continually refining them based on approximate or partial results. Therefore, instead of optimizing for query response time, it needs to balance two conflicting performance goals: minimizing uneventful "dead time" between updates for the users, while simultaneously maximizing the rate at which partial or approximate answers approach a correct answer. Please refer [8] for details.

Compared with pre-computed synopses approach, the advantages of online query processing approach is it does not require pre-processing and progressive refinement of approximate results at runtime can quickly lead to a satisfied results. However, the main obstacles for this technique to be practical is it significantly changes the query processor of current commercial query processing system which is not desirable.

KEY APPLICATIONS

AQP in Relational Data Management

The AQUA system proposed by the Bell lab can support various kinds of approximate queries over the relational database.

AQP in Spatial Data Management

Alexandria Digital Library allows user to define spatial queries and returns the approximate number of hits quickly as the initial result and then user can refine the query accordingly.

AQP in Stream Data Management

MIT proposed Aurora as a data stream management system, which virtually supports answering approximate queries over the data stream.

AQP in Sensor Network

The techniques of TinyDB system, proposed by Massachusetts Institute of Technology and UC Berkeley, can lead to orders of magnitude improvements in power consumption and increased accuracy of query results over non-acquisitional systems that do not actively control when and where data is collected.

AQP in Semantic Web Search

For semantic web search, the viewpoints of users performing a web search, ontology designers and annotation designers don't match. This leads to missed answers. Research studies have shown AQP is of prime importance for efficiently searching the Semantic Web.

CROSS REFERENCES

Decision Support System, Data Mining, Aggregate Query, Sampling, Histogram, Wavelet, Query Optimization, Selectivity, Online Analytical Processing

RECOMMENDED READING

- [1] Morgenstein, J. Computer Based Management Information Systems Embodying Answer Accuracy as a User Parameter. PhD thesis, U.C. Berkeley, 1980.
- [2] Ioannidis, Y. Approximation in Database Systems. ICDT, 2003.
- [3] Poosala, V. and Ioannidis, Y. Selectivity Estimation Without the Attribute Value Independence Assumption. VLDB, 1997.
- [4] Poosala, V., Ioannidis, Y., Haas, P. and Shekita, E. Improved histograms for selectivity estimation of range predicates. SIGMOD, 1996.
- [5] Das, G. Sampling Methods in Approximate Query Answering Systems. Invited Book Chapter, Encyclopedia of Data Warehousing and Mining. Editor John Wang, Information Science Publishing, 2005.
- [6] Matias, Y., Vitter, J. and Wang, M. Wavelet based histograms for selectivity estimation. SIGMOD, 1998.
- [7] Das, A., Gehrke, J. and Riedewald, M. Approximation techniques for spatial data. SIGMOD, 2004.
- [8] Hellerstein, J., Haas, P. and Wang, H. Online Aggregation. SIGMOD, 1997.

- [9] Garofalakis, M. and Gibbons, P. Approximate Query Processing: Taming the TeraBytes: A Tutorial. VLDB, 2001
- [10] Ioannidis, Y. The history of histograms (abridged). VLDB, 2003.