Approximate Aggregate Query Processing: Olap

Stefania Marrara
Politecnico di Milano
marrara@elet.polimi.it

Outline
• Intro & Approximate Query Answering Overview
  - Synopses
• One-Dimensional Synopses
  - Histograms, Samples, Wavelets
• Multi-Dimensional Synopses and Joins
  - Multi-D Histograms, Join synopses, Wavelets
• Discussion & Comparisons
• Using synopses in the telecommunication environment
• Conclusions
Introduction and Motivations

Operational Database
GB/TB

SQL Query
Long Response Times!
Exact Answer

Compact Data Synopses
KB/MB

"Transformed" Query
Approximate Answer
FAST!!

Motivations

• Exact answers **NOT** always required
  - DSS applications usually *exploratory*: early feedback to help identify "interesting" regions
  - *Aggregate queries*: precision to "last decimal" not needed
    • e.g., "What percentage of the US sales are in NJ?" (display as bar graph)
  - *Preview* answers while waiting. *Trial* queries
  - Base data can be *remote or unavailable*: approximate processing using locally-cached *data synopses* is the only option
**Fast Approximate Answers**

- Primarily for *Aggregate* queries
- *Goal* is to quickly report the leading digits of answers
  - In seconds instead of minutes or hours
  - Most useful if can provide error guarantees

E.g., Average salary

- $59,000 +/- $500 (with 95% confidence) in 10 seconds
- vs. $59,152.25 in 10 minutes

**Fast Approximate Answers**

- Achieved by answering the query based on samples or other synopses of the data
- Speed-up obtained because synopses are orders of magnitude smaller than the original data
Approximate Query Answering

Basic Approach 1: Online Query Processing
- Sampling at query time
- Answers continually improve, under user control

Approximate Query Answering

Basic Approach 2: Precomputed Synopses
- Construct & store synopses prior to query time
- At query time, use synopses to answer the query
- Like estimation in query optimizers, but
  - reported to the user (need higher accuracy)
  - more general queries
- Need to maintain synopses up-to-date

Most work in the area based on the precomputed approach
  - e.g., Sample Views [OR92, Olk93], Aqua Project [GMP97a, AGP99, etc]
Online vs. Precomputed

Online:

+ Continuous refinement of answers (online aggregation)
+ User control: what to refine, when to stop
+ Seeing the query is very helpful for fast approximate results
+ No maintenance overheads

Precomputed:

+ Seeing entire data is very helpful (provably & in practice)
  (But must construct synopses for a family of queries)
+ Often faster: better access patterns,
  small synopses can reside in memory or cache
+ Middleware: Can use with any DBMS, no special index striding
+ Also effective for remote or streaming data
Commercial DBMS

- **Oracle, IBM Informix**: Sampling operator (online)
- **IBM DB2**: “IBM Almaden is working on a prototype version of DB2 that supports sampling. The user specifies a priori the amount of sampling to be done.”
- **Microsoft SQL Server**: “New auto statistics extract statistics [e.g., histograms] using fast sampling, enabling the Query Optimizer to use the latest information.”

The index tuning wizard uses sampling to build statistics.
- see [CN97, CMN98, CN98]

In summary, not much announced yet

Outline

- Intro & Approximate Query Answering Overview
- One-Dimensional Synopses
  - **Histograms**: Equi-depth, Compressed,
  - **Samples**: Basics, Sampling from DBs, Reservoir Sampling
  - **Wavelets**: 1-D Haar-wavelet histogram construction & maintenance
- Multi-Dimensional Synopses and Joins
- Discussion & Comparisons
- Using synopses in the telecommunication environment
- Conclusions
**Histograms**

- Partition attribute value(s) domain into a set of buckets
- **Issues:**
  - How to partition
  - What to store for each bucket
  - How to estimate an answer using the histogram
- Long history of use for selectivity estimation within a query optimizer [Koo80], [PSC84], etc.
- [PIH96] [Poo97] introduced a taxonomy, algorithms, etc.

---

**1-D Histograms: Equi-Depth**

- **Goal:** Equal number of rows per bucket (B buckets in all)
- **Can construct** by first sorting then taking B-1 equally-spaced splits
- **Faster construction:** Sample & take equally-spaced splits in sample
  - Nearly equal buckets
  - Can also use one-pass algorithms (e.g., [GK01])
1-D Histograms: Equi-Depth

- Can maintain using one-pass algorithms (insertions only), or
- Use a backing sample [GMP97b]: Maintain a larger sample on disk in support of histogram maintenance
  - Keep histogram bucket counts up-to-date by incrementing on row insertion, decrementing on row deletion
  - Merge adjacent buckets with small counts
  - Split any bucket with a large count, using the sample to select a split value, i.e., take median of the sample points in bucket range
    - Keeps counts within a factor of 2; for more equal buckets, can recompute from the sample

Answering Queries: Equi-Depth

Answering queries:
- select count(*) from R where 4 <= R.A <= 15
- approximate answer: F * |R|/B, where
  - F = number of buckets, including fractions, that overlap the range
  - error guarantee: ± 2 * |R|/B

answer: $3.5 \times \frac{|R|}{6} \pm 0.5 \times \frac{|R|}{6}$
Answering Queries: Histograms

- Answering queries from 1-D histograms (in general):
  - (Implicitly) map the histogram back to an approximate relation, & apply the query to the approximate relation

Sampling: Basics

- Idea: A small random sample $S$ of the data often well-represents all the data
  - For a fast approx answer, apply the query to $S$ & “scale” the result
  - E.g., $R.a$ is (0,1), $S$ is a 20% sample
    - \[ \text{select count(*) from } R \text{ where } R.a = 0 \]
    - \[ \text{select 5 * count(*) from } S \text{ where } S.a = 0 \]
    - Est. count = 5*2 = 10, Exact count = 10
Sampling: Basics

- **Unbiased**: For expressions involving count, sum, avg: the estimator
  - is unbiased, i.e., the expected value of the answer is the actual answer,
  - even for (most) queries with predicates!

- Leverage extensive literature on **confidence intervals** for sampling
  - Actual answer is within the interval $[a,b]$ with a given probability
    - E.g., $54,000 \pm 600$ with prob $\geq 90\%$

Sampling from Databases

- Sampling disk-resident data is slow
  - Row-level sampling has high I/O cost:
    - must bring in entire disk block to get the row
  - Block-level sampling: rows may be highly correlated
  - Random access pattern, possibly via an index
Sampling from Databases

- Alternatives
  - Random physical clustering: destroys “natural” clustering
  - Precomputed samples: must incrementally maintain (at specified size)
    - Fast to use: packed in disk blocks, can sequentially scan, can store as relation and leverage full DBMS query support, can store in main memory

One-Dimensional Haar Wavelets

- Wavelets: mathematical tool for hierarchical decomposition of functions/signals
- Haar wavelets: simplest wavelet basis, easy to understand and implement
  - Recursive pairwise averaging and differencing at different resolutions

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Averages</th>
<th>Detail Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>[2, 2, 0, 2, 3, 5, 4, 4]</td>
<td>[0, -1, -1, 0]</td>
</tr>
<tr>
<td>2</td>
<td>[2, 1, 4, 4]</td>
<td>[0.5, 0]</td>
</tr>
<tr>
<td>1</td>
<td>[1.5, 4]</td>
<td>[-1.25]</td>
</tr>
<tr>
<td>0</td>
<td>[2.75]</td>
<td></td>
</tr>
</tbody>
</table>

Haar wavelet decomposition: [2.75, -1.25, 0.5, 0, 0, -1, -1, 0]
Wavelet-based Histograms [MVW98]

• Problem: range-query selectivity estimation

• Key idea: use a compact subset of Haar/linear wavelet coefficients for approximating the data distribution

Wavelet-based Histograms [MVW98]

• Steps
  - compute cumulative data distribution $C$
  - compute Haar (or linear) wavelet transform of $C$
  - coefficient thresholding: only $b\ll|C|$ coefficients can be kept
    • take largest coefficients in absolute normalized value
      - Haar basis: divide coefficients at resolution $j$ by $\sqrt{2^j}$
      - *Optimal* in terms of the overall Mean Squared (L2) Error
    • Greedy heuristic methods
      - Retain coefficients leading to large error reduction
      - Throw away coefficients that give small increase in error
Outline

• Intro & Approximate Query Answering Overview
• One-Dimensional Synopses
• Multi-Dimensional Synopses and Joins
  - Multi-dimensional Histograms
  - Join sampling
  - Multi-dimensional Haar Wavelets
• Discussion & Comparisons
• Using synopses in the telecommunication environment
• Conclusions

Multi-dimensional Data Synopses

• Problem: Approximate the joint data distribution of multiple attributes

  - Motivation
    • Selectivity estimation for queries with multiple predicates
    • Approximating OLAP data cubes and general relations
Multi-dimensional Data Synopses

- Conventional approach: Attribute-Value Independence (AVI) assumption
  - \( \text{sel}(p(A1) \& p(A2) \& \ldots) = \text{sel}(p(A1)) \ast \text{sel}(p(A2)) \ast \ldots \)
  - Simple -- one-dimensional marginals suffice
  - BUT: almost always inaccurate, gross errors in practice (e.g., [Chr84, FK97, Poo97])

Multi-dimensional Histograms

- Use small number of multi-dimensional buckets to directly approximate the joint data distribution
- Uniform spread & frequency approximation within buckets
  - \( n(i) \) = no. of distinct values along \( A_i \), \( F \) = total bucket frequency
  - approximate data points on a \( n(1) \ast n(2) \ast \ldots \) uniform grid, each with frequency \( F / (n(1) \ast n(2) \ast \ldots) \)
Multi-dimensional Histogram Construction

- Construction problem is much harder even for two dimensions [MPS99]

- Multi-dimensional equi-depth histograms [MD88]
  - Fix an ordering of the dimensions $A_1, A_2, \ldots, A_k$, let $\alpha = \text{kth}$ root of desired no. of buckets, initialize $B = \{ \text{data distribution} \}$
  - For $i=1, \ldots, k$: Split each bucket in $B$ in $\alpha$ equi-depth partitions along $A_i$; return resulting buckets to $B$
  - Problems: limited set of bucketizations; fixed $\alpha$ and fixed dimension ordering can result in poor partitionings

Multi-dimensional Histogram Construction

- MHIST-$p$ histograms [PI97]
  - At each step
    - Choose the bucket $b$ in $B$ containing the attribute $A_i$ whose marginal is the most in need of partitioning
    - Split $b$ along $A_i$ into $p$ (e.g., $p=2$) buckets
Sampling for Multi-D Synopses

• Taking a sample of the rows of a table captures the attribute correlations in those rows
  - Answers are unbiased & confidence intervals apply
  - Thus guaranteed accuracy for count, sum, and average queries on single tables, as long as the query is not too selective

Sampling for Multi-D Synopses

• Problem with joins [AGP99,CMN99]:
  - Join of two uniform samples is not a uniform sample of the join
  - Join of two samples typically has very few tuples

Foreign Key Join
40% Samples in Red
Size of Actual Join = 30
Size of Join of samples = 3
Join Synopses for Foreign-Key Joins [AGP99]

- Based on sampling from materialized foreign key joins
  - Typically < 10% added space required
  - Yet, can be used to get a uniform sample of ANY foreign key join
  - Plus, fast to incrementally maintain

- Significant improvement over using just table samples
  - E.g., for TPC-H query Q5 (4 way join)
    - 1%-6% relative error vs. 25%-75% relative error, for synopsis size = 1.5%, selectivity ranging from 2% to 10%
    - 10% vs. 100% (no answer!) error, for size = 0.5%, select. = 3%

Multi-dimensional Haar Wavelets

- Basic “pairwise averaging and differencing” ideas carry over to multiple data dimensions
- Two basic methodologies – no clear winner [SDS96]
  - Standard Haar decomposition
  - Non-standard Haar decomposition
Outline

• Intro & Approximate Query Answering Overview
• One-Dimensional Synopses
• Multi-Dimensional Synopses and Joins
• Discussion & Comparisons
• Using synopses in the telecommunication environment
• Conclusions

Discussion & Comparisons

• Histograms & Wavelets: Limited by “curse of dimensionality”
  - Rely on data space partitioning in “regions”
  - Ineffective above 5-6 dimensions
    - Value/frequency uniformity assumptions within buckets break down in medium-to-high dimensionalities!!
Discussion & Comparisons

- **Sampling:** No such limitations, BUT...
  - Ineffective for ad-hoc relational joins over arbitrary schemas
    - Uniformity property is lost
    - Quality guarantees degrade
  - Effectiveness for *set-valued* approximate queries is unclear
    - Only (very) small subsets of the answer set are returned
      (especially, when joins are present)

Discussion & Comparisons

- **Histograms & Wavelets:** Compress data by accurately capturing rectangular “regions” in the data space
  - Advantage over sampling for typical, “range-based” relational DB queries
  - BUT, unclear how to effectively handle unordered/non-numeric data sets (no such issues with sampling...)
Discussion & Comparisons

- **Sampling**: Provides strong probabilistic quality guarantees (unbiased answers) for individual aggregate queries
  - **Histograms & Wavelets**: Can guarantee a bound on the overall error (e.g., L2) for the approximation, BUT answers to individual queries can be heavily biased!!

  **No clear winner exists!! (Hybrids??)**

Outline

- Intro & Approximate Query Answering Overview
- One-Dimensional Synopses
- Multi-Dimensional Synopses and Joins
- Discussion & Comparisons
- Using synopses in the telecommunication environment
- Conclusions
GSM traffic compression and analysis by means of wavelets decomposition.

• This work, which has the aim to provide fast approximate results to the analysis of GSM data, was born from a cooperation between the University of Milan, Department of Computer Technologies, and Siemens Mobile Communications S.p.A., a multinational company in the field of telecommunication networks. The software developed by Siemens for mobile networks performance analysis, and basis of this work, is OTS (O&M ToolSet).

Introduction

• Third generation mobile networks were designed to satisfy raising requests of performance, reliability and availability on behalf of customers all over the world. To meet these requirements, a brand new architecture has been built where, with particular reference to the UMTS Radio Access Network (UTRAN), new Network Entities carry out more complex functionalities and can serve many more calls at the time than the previous networks did. On the other hand, in order to grant high efficiency and the best performance of the network, such a complex scenario must be monitored and controlled by a certain number of Operation & Maintenance Centres (OMCs).
The Net

• In this situation, a few thousands of NodeBs together with few tens of Radio Network Controllers (RNCs) give a snapshot of the amount of Network Entities directly managed by a unique OMC whilst, in GSM system, the same analysis was done by less than an hundred of Base Station Systems (BSSs).

The Net

• Finally, an important role to optimize the usage of UMTS networks is played by the evaluation of performance measurements periodically given from NodeBs and RNCs. These measurements can be mainly grouped into:
  - Physical Link Measurements (e.g. Bit Error Rate, Uplink/Downlink SNR);
  - Message Flow Measurements (e.g. Packet Drop Rate);
  - Call Measurements (e.g. call drop, call establishments, total calls, handover requests, handover failures).
Analysis of the Net

• These huge amount of raw measurement data are difficult to evaluate and, in general, do not give a clear idea of the actual course of the network health. Therefore, starting from these huge amount of raw measurement data got directly from controlled Network Elements, many performance indicators can be computed applying, for instance:
  - primitive mathematic operators like addition, subtraction, multiplication and division;
  - trends like average, maximum and minimum applied to the entire set of data or to well-defined period of time;
  - time and object aggregations;
  - telecommunication operators.

QMS synopses

• This project was born with the aim to integrate the next version of the QMS database with important functionalities:
  - to compress the data stored in the database in order to keep one year history of a GSM net. Without this tool QMS was able to store not more than 15 days of data;
  - to analyze the behavior of the net obtaining fast approximate answers to trend queries. In this situation the analyzer is most interested in obtaining fast results paying a small loss in precision than obtaining exact answer to his queries in long computational times (e.g., hours or at worst days);
  - to query the single signal behavior with simple, immediate, queries, avoiding the complex system of views that was necessary with the first version of QMS;
  - to analyze the behavior of the signals, in order to discover periodicity and other interesting statistical information (e.g., variance, average....)
The idea

- For each table in the QMS database, a new table syn-table is constructed containing the compressed dataset.

- Techniques used:
  - Haar wavelets
  - Daubechies4 e Daubechies6
  - Adaptative Haar wavelets (new idea...)
  - Correlation and Haar

Haar wavelet decomposition

- Wavelets: mathematical tool for hierarchical decomposition of functions/signals
- Haar wavelets: simplest wavelet basis, easy to understand and implement
  - Recursive pair wise averaging and differencing at different resolutions.
  - A linear time algorithm.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Averages</th>
<th>Detail Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>[2, 2, 0, 2, 3, 5, 4, 4]</td>
<td>--</td>
</tr>
<tr>
<td>2</td>
<td>[2, 1, 4, 4]</td>
<td>[0, -1, -1, 0]</td>
</tr>
<tr>
<td>1</td>
<td>[1.5, 4]</td>
<td>[0.5, 0]</td>
</tr>
<tr>
<td>0</td>
<td>[2.75]</td>
<td>[-1.25]</td>
</tr>
</tbody>
</table>
The Haar Basis

Haar Wavelet Decomposition

- With the Haar wavelet family, we can approximate an arbitrary $f(x) \in L^2([0,1])$ on Haar family's compact support $[0,1)$ with an arbitrary precision given as many generations as needed.
- By scaling and translation, we can get more general compact support in $\mathcal{R}$. 
Haar Wavelet Decomposition

- Example

![Haar Wavelet Diagram](image)

Daubechies4 e Daubechies6

- It is similar to the Haar wavelets but involves more complex coefficients that consider the average of the three nearest coefficients instead of just the nearest two.
- \( D = 4\sqrt{2} \)
- \( H_0 = (1 + \sqrt{3})/d \)
- \( H_1 = (3 + \sqrt{3})/d \)
- \( H_2 = (3 - \sqrt{3})/d \quad \text{it correct negative values} \)
- \( H_3 = (1 - \sqrt{3})/d \)
- \( W[i] = h_0 a_0 + h_1 a_1 + h_2 a_2 + h_3 a_3 \)
- \( W[i+n/2] = a[n-2] h_3 + a[n-1] h_2 + a[0] h_1 + a[1](-h_0) \)

- Daubechies6 uses 6 coefficients but the idea is similar
**Haar vs Daubechies**

- Haar -> it allows to control the error but loses the shape of the signal
- Daubechies -> it follows the shape but it is difficult to control the error

---

**Adaptative Haar wavelets**

- The idea was born from the study of temporal series behavior
- We use different Haar bases along the same signal, choosing the best one, interval after interval, on the basis of the accepted error.
- This mechanism outlines the correlation among contiguous signals.
Correlation and Haar

- Using a combination between Haar wavelets and correlation information among signals we are able to reduce redundancy among signals with the same statistical behavior obtaining synopses that are 10% of the original dataset and are affected by an error of 5% during the querying process.

- Now we can store one year of history of a GSM net... (happiness of the customers 😊 )

Conclusions

- Wavelets are an optimal compression technique for about 80% of the telecommunication signals
- The remaining 20% (random) are luckily not important to the final user than they can be omitted in our synopsis
- The use of correlation and wavelets allows to store histories of the signals longer than with traditional storage, therefore it is simple to detect similar behaviors, errors, singularities.
References (1)

  - Proposes exploiting simple (differential and combinational) data dependencies for effectively compressing data tables.

References (2)

  - Proposes a novel, "model-based semantic compression" methodology that exploits mining models (like CaRT trees and clusters) to build compact, guaranteed-error synopses of massive data tables.
  - Studies the effectiveness of histograms, kernel-density estimators, and their hybrids for estimating the selectivity of range queries over metric attributes with large domains.
  - Precursor to [CDN01]. Proposes a method for reducing sampling variance by collecting outliers to a separate "outlier index" and using a weighted sampling scheme for the remaining data.
References (3)

  - Presents a parametric, curve-fitting technique for approximating an attribute's distribution based on query feedback.

References (4)

  - Proposes the use of "multifractals" (i.e., 80/20 laws) to more accurately approximate the frequency distribution within histogram buckets.
  - Presents algorithms for building "range-optimal" histogram and wavelet synopses: that is, synopses that try to minimize the total error over all possible range queries in the data domain.
References (5)

  - Proposes the "concise sample" and "counting sample" techniques for improving the accuracy of sampling-based estimation for a given amount of space for the sample synopsis.

References (6)

  - Proposes novel, Bayesian-network-based techniques for approximating joint data distributions in relational database systems.
  - Proposes and evaluates several sampling-based estimators for the number of distinct values in an attribute column.
References (7)

  - The above three papers propose and study serial histograms (i.e., histograms that bucket “neighboring” frequency values, and exploit results from majorization theory to establish their optimality wrt minimizing (extreme cases of) the error in multi-join queries.
  - Discusses the use of “fascicles” (i.e., approximate data clusters) for the semantic compression of relational data.

References (8)

- Proposes the use of SVD techniques for obtaining fast approximate answers from large time-series databases.
  - Proposes the use of linear splines to better approximate the data and frequency distribution within histogram buckets.
  - Proposes techniques for enhancing hierarchical multi-dimensional index structures to enable approximate answering of aggregate queries with progressively improving accuracy.
  - Presents an adaptive, sequential sampling scheme for estimating the selectivity of relational equi-join operators.
References (9)

  - Presents adaptive-sampling-based techniques and estimators for approximating the result size of a relational projection operation.

References (10)

  - Discusses the use of mixture models composed of multi-variate Gaussians for building compact models of OLAP data cubes and approximating range-sum query answers.
References (11)

  - Uses class hierarchies on the data to iteratively fetch blocks relevant to the answer, producing tuples certain to be in the answer while narrowing the possible classes containing the answer.