A Quick Introduction to Approximate Query Processing

Part-IV

CS286, Spring'07
Minos Garofalakis

Logistics...

- Draft CS286 web site is finally up!
  - http://db.cs.berkeley.edu/cs286sp07/

- Project list and guidelines being worked on
  - Please email me & Raghu to discuss your own project ideas...

Outline

- Intro & Approximate Query Answering Overview
  - Synopses, System architectures, Commercial offerings
- One-Dimensional Synopses
  - Histograms: Equi-depth, Compressed, V-optimal, Incremental maintenance, Self-tuning
  - Samples: Basics, Sampling from DBs, Reservoir Sampling
  - Wavelets: 1-D Haar-wavelet histogram construction & maintenance
- Multi-Dimensional Synopses and Joins
- Set-Valued Queries
- Discussion & Comparisons
- Advanced Techniques & Future Directions

Outline

- Intro & Approximate Query Answering Overview
  - Synopses, System architecture, Commercial offerings
- One-Dimensional Synopses
  - Histograms, Samples, Wavelets
- Multi-Dimensional Synopses and Joins
  - Multi-D Histograms, Join synopses, Wavelets
- Set-Valued Queries
  - Error metrics; Using Histograms, Samples, Wavelets
  - Discussion & Comparisons
  - Advanced Techniques & Future Directions
  - Dependency-based, Streaming data

Relations as Frequency Distributions

One-dimensional distribution

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>MG</td>
<td>34</td>
<td>100K</td>
</tr>
<tr>
<td>JG</td>
<td>33</td>
<td>90K</td>
</tr>
<tr>
<td>RR</td>
<td>40</td>
<td>190K</td>
</tr>
<tr>
<td>JH</td>
<td>36</td>
<td>110K</td>
</tr>
<tr>
<td>MF</td>
<td>39</td>
<td>150K</td>
</tr>
<tr>
<td>DD</td>
<td>45</td>
<td>150K</td>
</tr>
<tr>
<td>JN</td>
<td>43</td>
<td>140K</td>
</tr>
<tr>
<td>AP</td>
<td>32</td>
<td>70K</td>
</tr>
<tr>
<td>EM</td>
<td>24</td>
<td>50K</td>
</tr>
<tr>
<td>DW</td>
<td>24</td>
<td>50K</td>
</tr>
</tbody>
</table>

Three-dimensional distribution

<table>
<thead>
<tr>
<th>Age</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Outline

- Intro & Approximate Query Answering Overview
  - Synopses, System architecture, Commercial offerings
- One-Dimensional Synopses
  - Histograms, Samples, Wavelets
- Multi-Dimensional Synopses and Joins
  - Multi-D Histograms, Join synopses, Wavelets
- Set-Valued Queries
  - Error metrics; Using Histograms, Samples, Wavelets
  - Discussion & Comparisons
  - Advanced Techniques & Future Directions
  - Dependency-based, Streaming data

Outline

- Intro & Approximate Query Answering Overview
  - Synopses, System architecture, Commercial offerings
- One-Dimensional Synopses
  - Histograms, Samples, Wavelets
- Multi-Dimensional Synopses and Joins
  - Multi-D Histograms, Join synopses, Wavelets
- Set-Valued Queries
  - Error metrics; Using Histograms, Samples, Wavelets
  - Discussion & Comparisons
  - Advanced Techniques & Future Directions
  - Dependency-based, Streaming data
**Two-dimensional Haar Wavelets -- Non-standard decomposition**

- Averaging &
- Differencing

Wavelet Transform Array:

```
+---+---+---+---+
|   | a+b+c+d   |   |
|---+---+---+---|
+---+---+---+---+
```

```
+---+---+---+---+
|   | a+b+c+d   |   |
|---+---+---+---|
+---+---+---+---+
```

**Multi-dimensional Haar Wavelets**

- Haar decomposition in d dimensions = d-dimensional array of wavelet coefficients
- Coefficient support region = d-dimensional rectangle of cells in the original data array
- Sign of coefficient's contribution can vary along the quadrants of its support

Support regions & signs for the 16 nonstandard 2-dimensional Haar coefficients of a 4X4 data array A

```
+---+---+---+---+
|   |   |   |
|---+---+---+---|
+---+---+---+---+
```

**Range-sum Estimation Using Wavelet Synopses**

- Coefficient thresholding
  - As in 1-d case, normalizing by appropriate constants and retaining the largest coefficients minimizes the overall L2 error
- Range-sums: selectivity estimation or OLAP-cube aggregates [VW99] ("measure attribute as count")
- Only coefficients with support regions intersecting the query hyper-rectangle can contribute
  - Many contributions can cancel each other [CGR00, VW99]

**Approximate Query Processing Using Wavelets [CGR00]**

- Reduce relations into compact wavelet-coefficient synopses

**Entire query processing in the compressed (wavelet) domain**

```
  Querying in Wavelet Domain
    Query Results in Wavelet Domain
      Render
  Compressed domain (FAST)
  Relation domain (SLOW)
  Final Approximate Results
```

**Wavelet Query Processing**

- Each operator (e.g., select, project, join, aggregates, etc.)
  - input: set of wavelet coefficients
  - output: set of wavelet coefficients
- Finally, rendering step
  - input: set of wavelet coefficients
  - output: (multi)set of tuples

**Selection -- Relational Domain**

- In relational domain, interested in only those cells inside query range
- In wavelet domain, interested in only the coefficients that contribute to those cells
**Selection -- Wavelet Domain**

![Diagram](image)

**Equi-join -- Relational Domain**

- Consider all pairs of coefficients: (1) check joinability (overlap in join dimension(s)), (2) compute output coefficients

![Diagram](image)

**Equi-join -- Wavelet Domain**

![Diagram](image)

**Wavelet Query Processing**

- Each operator (e.g., select, project, join, aggregates, etc.)
  - input: set of wavelet coefficients
  - output: set of wavelet coefficients
- Finally, rendering step
  - input: set of wavelet coefficients
  - output: (multi)set of tuples

![Diagram](image)

**Outline**

- Intro & Approximate Query Answering Overview
- One-Dimensional Synopses
- Multi-Dimensional Synopses and Joins
- Set-Valued Queries
- Discussion & Comparisons
- Advanced Techniques & Future Directions
- Conclusions

![Diagram](image)

**Discussion & Comparisons (1)**

- Histograms & Waves: 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 dimensionality!
- 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 (2)**
- 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 issues with sampling...)
- 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
- Set-Valued Queries
- Discussion & Comparisons
  - Advanced Techniques & Future Directions
    - Dependency-based Synopses
    - Streaming Data
    - XML Synopses
    - Conclusions

**Dependency-based Histogram Synopses [DGRO1]**
- **Attribute Value Independence**: *inaccurate
- Multi-dimensional histograms on joint data distribution
  - Fully independent attributes
  - Fully correlated attributes
- Extremes in terms of the underlying correlations!
- Dependency-Based Histograms: explore space between extremes by explicitly identifying data correlations/independencies
  - Build a statistical interaction model on data attributes
  - Based on the model, build a collection of low-dimensional histograms
  - Use this histogram collection to provide approximate answers
- General methodology, also applicable to other synopsis techniques (e.g., wavelets)

**Dependency-based Histograms**
- Identify (and exploit) attribute correlation and independence
  - Partial Independence:
    - \( p(\text{salary}, \text{height}, \text{weight}) \neq p(\text{salary}) \cdot p(\text{height}, \text{weight}) \)
  - Conditional Independence:
    - \( p(\text{salary}, \text{age} | \text{YPE}) \neq p(\text{salary} | \text{YPE}) \cdot p(\text{age} | \text{YPE}) \)
- Use forward selection to build a decomposable statistical model [BFH75, Lue96] on the attributes
  - A, B are conditionally independent given C
  - \( p(\text{AB|C}) = p(\text{A|C}) \cdot p(\text{B|C}) \)
  - Joint distribution
  - \( p(\text{AB|C}) = p(\text{ABC}) / p(\text{C}) \)
  - Build histograms on model cliques
- Significant accuracy improvements (factor of 5) over pure MHIEST
- New histogram construction & usage algorithms, etc.

**Workload-tuned Biased Sampling -- Congressional Samples [AGPO0]**
- Decision support queries routinely segment data into groups & then aggregate the information within each group
  - Each table has a set of "grouping columns": queries can group by any subset of these columns
- **Goal**: Maximize the accuracy for all groups (large or small) in each group-by query
  - E.g., census DB with state (s), gender(g), and income (i)
  - Q: avg(s) group-by s: seek good accuracy for all 50 states
  - Q: avg(g) group-by g, i: seek good accuracy for all 100 groups
- **Technique**: Congressional Samples
  - House: Uniform sample: good for when no group-by
  - Senate: Same size sample per group when using all grouping columns: good for queries with all columns
  - Congress: Combines House & Senate, but considers all subsets of grouping columns, and then scales down

**Workload-tuned Biased Sampling -- TCICLES [GRO0]**
- Biased sampling scheme that dynamically adapts to query workload
  - Exploit data locality -- more focus (i.e., sample points) in frequentlyqueried regions
  - Let \( Q = (q_1, q_2, \ldots) \) be a query workload, \( R(q) \subseteq R \) used in answering query \( q \)
  - \( L(R, Q) = \) Extension of \( R \) wrt \( Q = R \cup \bigcup_{Q \in Q} R(Q) \) (multiset of tuples)
- **Icicle**: Uniform random sample of \( L(R, Q) \)
  - Incrementally maintained and adapt ("self-tune") to workload through Reservoir Sampling technique [VF85]
- **Unbiased Icicle estimators**: New formulas to account for duplicates and bias in sample selection
  - Provably better (smaller variance) than uniform for "focused" queries (that follow the workload model)
Workload-tuned Biased Sampling -- Lifted Workloads [CONJO]

- Formulate sample selection as an optimization problem
- Minimize query-answering error for a given workload model
- Technique for "lifting a fixed workload W" to produce a probability distribution over all possible queries
- Similar to kernel density estimation (queries in W = "sample points")
- \( W = \{ q_1, q_2 \} \)

\[ \text{prob}(q \mid W) = \text{parametric function of } q \text{'s overlap with queries in } W \]

"Fundamental regions" induced by W

Data Streams

- Data is continually arriving. Collect & maintain synopses on the data. Goal: Highly-accurate approximate answers
  - State-of-the-art: Good techniques for narrow classes of queries
  - E.g., Any one-pass algorithm for collecting & maintaining a synopsis can be used effectively for data streams
- Alternative scenario: A collection of data sets. Compute a compact sketch of each data set & then answer queries (approximately) comparing the data sets
  - E.g., detecting near-duplicates in a collection of web pages: Altovista
  - E.g., estimating join sizes among a collection of tables [AGM99]

Looking Forward...

- Optimizing queries for approximation
  - e.g., minimize length of confidence interval at the plan root
- Exploiting mining-based techniques (e.g., decision trees) for data reduction and approximate query processing
  - see, e.g., [BGR01], [GTK01], [JMN99]
- Dynamic maintenance of complex (e.g., dependency-based [BGR01] or mining-based [BGR01]) synopses
- Synopses and approximate query processing for richer data models and data streams
  - e.g., XPath/XQuery over XML databases

XML Data (Text)

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<booklist>
  <book>
  <book>
</booklist>
```

XML Data (Tree)

```
<booklist>
  <book>
    <@bookgen>Science</bookgen> <title>The character of physical Law</title> <author><firstname>Richard</firstname> <lastname>Feynman</lastname></author> 
    <@bookgen>Fiction</bookgen> <title>Waiting for the Mahatma</title>
  </book>
</booklist>
```
**XML Basics**

- **Elements**
  - Encode "concepts" in the XML database
  - Nesting denotes association/inclusion
- **Attributes**
  - Record information specific to an element (e.g., the genre of a book)
- **References**
  - Links between elements in different parts of the document

**XML vs. Relational Data**

- A relation instance is basically a tree with:
  - Unbounded fanout at level 1 (i.e., any # of rows)
  - Fixed fanout at level 2 (i.e., fixed # fields)

- XML data is essentially an arbitrary tree:
  - Unbounded fanout at all nodes/levels
  - Any number of levels
  - Variable # of children at different nodes, variable path lengths

**XPath Expressions**

Examples:
- /booklist/book
- /booklist/book/author
- /booklist/book/author/lastname

Given an XML document, the value of a path expression is a set of elements (≠ XML subtrees)

**Path Expressions**

- XPath expressions
  - Simple: /A/P/T
  - Branching: /A[B]/P/T
  - Values: /A/P/T[vvII]
  - Result is a set
**Path Expressions**

- **XPath expressions**
  - Simple: `/A/P/T`
  - Branching: `/[A][B]/P/T`
  - Values: `/A/P/T[v<11]`
  - Result is a **set**

**XPath Syntax**

- Path wildcards
  - `//` = descendant at any level (or self)
  - `*` = any (single) tag
  - Example: `/booklist//lastname`
- Query attributes and attribute content
  - Use `@`:
- Branching predicates: `A[pre]`
  - Predicate on A's subtree using logical connectives (and, or, etc.), path expressions, built-in functions (e.g., `contains()`), etc.
  - Example: `/author[contains(/lastname, 'Fey')]`

**Synopses for XML**

- Summarize labeled tree/graph structure for approximate path navigation queries
  - **Selectivity estimation**: How many elements satisfy p?
  - **Approximate answers**: Return an approximate XML document as output of an XQuery fragment
- **Key idea**: Build a concise **Graph Synopsis** that captures the path/branching distribution in limited space
  - Use appropriate uniformity/independence assumptions to approximate path structure
  - Refine synopsis in parts of the XML document where assumptions fail
  - XS Sketches [SIGMOD’02, VLDB’02], Tree Sketches [SIGMOD’04]

**Conclusions**

- Commercial data warehouses: approaching several 100’s TB and continuously growing
  - Demand for high-speed, interactive analysis (click-stream processing, IP traffic analysis) also increasing
- **Approximate Query Processing**
  - "Tame" these Terabytes and satisfy the need for interactive processing and exploration
  - Great promise
  - Commercial acceptance still lagging, but will most probably grow in coming years
  - Still lots of interesting research to be done