Approximate Query Processing using Data Synopses

- How to construct effective data synopses??

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

Decision Support Systems

- Data Warehousing: Consolidate data from many sources in one large repository.
  - Loading, periodic synchronization of replicas.
  - Semantic integration.
- OLAP:
  - Complex SQL queries and views.
  - Queries based on spreadsheet-style operations and "multidimensional" view of data.
  - Interactive and "online" queries.
- Data Mining:
  - Exploratory search for interesting trends and anomalies. (Another lecture!)

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>2</td>
<td>[2, 0, 2, 0, 2, 0, 2, 0]</td>
<td>[0, -1, 0]</td>
</tr>
<tr>
<td>1</td>
<td>[2, 1, 1, 1, 1]</td>
<td>(0.5, 0)</td>
</tr>
<tr>
<td>0</td>
<td>[2, 2]</td>
<td>[-1, 2.5]</td>
</tr>
</tbody>
</table>

Haar wavelet decomposition: [2.75, -1.25, 0.5, 0, 0, -1, -1, 0]
**Haar Wavelet Coefficients**

- Hierarchical decomposition structure (a.k.a. "error tree")
- Coefficient "Supports"

Original data

**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

**Using Wavelet-based Histograms**

- Selectivity estimation: \( s_{ol}(a: X < b : C[b], \bar{C}[a-1]) \)
- \( C \) is the (approximate) "reconstructed" cumulative distribution
- Time: \( O(\text{min}(b, \log N)) \), where \( b \) = size of wavelet synopsis (no. of coefficients), \( N \) = size of domain

- At most \( \log N + 1 \) coefficients are needed to reconstruct any \( C \) value

**Haar Wavelet Coefficients**

- Reconstruct data values \( d(i) \)
  - \( d(i) = \sum a_k \cdot c_k \) (coefficient on path)

- Range sum calculation \( d(h) \)
  - \( d(h) = \text{simple linear combination of coefficients on path to } h \)

- Only \( O(\log N) \) terms

**Dynamic Maintenance of Wavelet-based Histograms (MVW00)**

- Build Haar-wavelet synopses on the original data distribution

**Effect of Distribution Updates**

- Key observation: for each coefficient \( c \) in the Haar decomposition tree
  - \( c = (\text{AVG(leftChildSubtree}(c)) - \text{AVG(rightChildSubtree}(c)) / 2 \)

- Change propagates up to the root coefficient

- As distribution changes, "most significant" (e.g., largest) coefficients can also change

- Important coefficients can become unimportant, and vice-versa

- Only coefficients on path(\( d \)) are affected and each can be updated in constant time
Maintenance Architecture

- "Shake up" when log reaches max size: for each insertion at d
  - for each coefficient c on path(d) and in H: update c
  - for each coefficient c on path(d) and not in H or H'
    - insert c into H' with probability proportional to 1/2^|h|
      where h is
      - "height" of c (Probabilistic Counting (FM85))
    - Adjust H and H' (move largest coefficients to H)

Approach: Optimize for Maximum-Error Metrics

- Key metric for effective approximate answers: Relative error with
  sanity bound:
  \[ \frac{|d_i - d_j|}{|d_i|, |d_j|} \]
  - Sanity bound "a" to avoid domination by small data values
  - To provide tight error guarantees for all reconstructed data values
    Minimize
    \[ \max \left( \frac{|d_i - d_j|}{|d_i|, |d_j|} \right) \]
    - Minimize maximum relative error in the data reconstruction
    - Another option: Minimize maximum absolute error:
      \[ \max \left( |d_i - d_j| \right) \]
    - Algorithms can be extended to general "distributive" metrics
      (e.g., average relative error)

Our Approach: Deterministic Wavelet Thresholding for Maximum Error

- Key Idea: Dynamic-Programming formulation that conditions the
  optimal solution on the error that "enters" the subtree (through the
  selection of ancestor nodes)

- S = subset of proper ancestors of j included in the synopsis

- Our DP table:
  \[ M_{j,b,S} = \text{optimal maximum relative (or absolute) error in } T(j) \]
  with space budget of b coefficients (chosen in T(i)), assuming subset
  S of j's proper ancestors have already been selected for the synopsis

- Clearly, |S| \leq \min(b, \log n - 1)
- Want to compute \( M(0, b, \emptyset) \)

- Basic Observation: Depth of the error tree is only \( \log n + 1 \)
  we can explore and tabulate all \( S \)-subsets for a given node at a
  space/time cost of only |O(N)|

Base Case for DP Recurrence: Leaf (Data) Nodes

- Base case in the bottom-up DP computation: Leaf (i.e., data) node \( d_j \)
- Assume for simplicity that data values are numbered \( N, 2N - 1 \)
- Selected coefficient \( M_{i,b,S} \) is not defined for b=0
- Never allocate space to leaves

- For b=0
  \[ M_{j,0,S} = \frac{|d_j - \sum_{c \in S} \text{sign}(c, d_j) c|}{\text{max}(1, |d_j|, |S|)} \]
  for each coefficient subset \( S \subseteq \text{path}(d_j) \)
  with |S| \leq \min(b, \log N-1)
  - Similarly for absolute error

- Again, time/space complexity per leaf node is only |O(N)|

DP Recurrence: Internal (Coefficient) Nodes

- Two basic cases when examining node/coordinate j for inclusion in the
  synopsis: (1) Drop j; (2) Keep j

Case (1): Drop Coefficient j

- In this case, the minimum possible maximum relative error in \( T(j) \) is
  \[ M_{\text{rel,0,j}} = \min \{ M(2j, b', S) \} \]

- Optimally distribute space \( b \) between j's two child subtrees

- Note that the RHS of the recurrence is well-defined
  - Ancestors of \( j \) are obviously ancestors of \( 2j \) and \( 2j+1 \)
DP Recurrence: Internal (Coefficient) Nodes (cont.)

Case (2): Keep Coefficient j

• In this case, the minimum possible maximum relative error in $T(j)$ is
  $M_{max}(j,b,S) = \min \max (M[j,b',S_{U(c_j)}], m_{b+1})$
  
  $M[j+1,b'-1,S_{U(c_j)}])$

• Take 1 unit of space for coefficient $j$, and optimally distribute remaining space
• Select subsets in RHS change, since we choose to retain $j$

• Again, the recurrence RHS is well-defined

• Finally, define $M[j,b,S] = \min (M_{max}(j,b,S), M_{max}(j,b',S))$
• Overall complexity: $O(N^3)$ time, $O(N \times \min(B, \log N))$ space

Outline

• Intro & Approximate Query Answering Overview
• One-Dimensional Synopses
• Multi-Dimensional Synopses and Joins
  - Multi-dimensional Histograms
  - Join sampling
  - Multi-dimensional Haar Wavelets
  - Set-Valued Queries
  - Discussion & Comparisons
  - Advanced Techniques & Future Directions
• 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

• Conventional approach: Attribute-Value Independence (AVI) assumption
  - sel(p(A1) & p(A2) & ... ) = sel(p(A1)) * sel(p(A2)) * ...
  - Simple --> one-dimensional marginals suffice
  - BUT: almost always inaccurate, gross errors in practice (e.g., [Ch94, FK97, P997])

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 $n(1)*n(2)*...$ uniform grid, each
  with frequency $F / (n(1)*n(2)*...)$

Actual Distribution (ONE BUCKET)

Approximate Distribution

Multi-dimensional Histogram Construction

• Construction problem is much harder even for two dimensions [MP99]

• Multi-dimensional equi-depth histograms [MD88]
  - Fix an ordering of the dimensions $A_1, A_2, \ldots, A_k$, let $\alpha = k$th root of desired no. of buckets, initialize $B = \{ $ 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 partitions

• $M\#EAST_p$ histograms [P97]
  - 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

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>Name</th>
<th>Age</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>10</td>
<td>20K</td>
</tr>
<tr>
<td>G</td>
<td>20</td>
<td>50K</td>
</tr>
<tr>
<td>A</td>
<td>30</td>
<td>100K</td>
</tr>
<tr>
<td>J</td>
<td>40</td>
<td>150K</td>
</tr>
<tr>
<td>D</td>
<td>50</td>
<td>200K</td>
</tr>
<tr>
<td>C</td>
<td>60</td>
<td>250K</td>
</tr>
<tr>
<td>T</td>
<td>70</td>
<td>300K</td>
</tr>
<tr>
<td>S</td>
<td>80</td>
<td>350K</td>
</tr>
<tr>
<td>H</td>
<td>90</td>
<td>400K</td>
</tr>
</tbody>
</table>

Multi-dimensional Data Synopses

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>10</td>
<td>15K</td>
</tr>
<tr>
<td>P</td>
<td>20</td>
<td>20K</td>
</tr>
<tr>
<td>Q</td>
<td>30</td>
<td>25K</td>
</tr>
<tr>
<td>R</td>
<td>40</td>
<td>30K</td>
</tr>
<tr>
<td>S</td>
<td>50</td>
<td>35K</td>
</tr>
<tr>
<td>T</td>
<td>60</td>
<td>40K</td>
</tr>
<tr>
<td>U</td>
<td>70</td>
<td>45K</td>
</tr>
<tr>
<td>V</td>
<td>80</td>
<td>50K</td>
</tr>
<tr>
<td>W</td>
<td>90</td>
<td>55K</td>
</tr>
</tbody>
</table>

Multi-dimensional Histogram Construction

- Construction problem is much harder even for two dimensions [MP99]

- Multi-dimensional equi-depth histograms [MD88]
  - Fix an ordering of the dimensions $A_1, A_2, \ldots, A_k$, let $\alpha = k$th root of desired no. of buckets, initialize $B = \{ $ 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 partitions

- $M\#EAST_p$ histograms [P97]
  - 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
Equi-depth vs. MHIST Histograms

- WHIST: choose bucket/dimension to split based on its criticality; allows for much larger class of bucketizations (hierarchical space partitioning)
- Experimental results verify superiority over AVI and equi-depth

Other Multi-dimensional Histogram Techniques -- GENHIST [GK100]

- Key idea: allow for overlapping histogram buckets
  - Allows for a much larger no. of distinct frequency regions for a given space budget (= #buckets)
  - Greedy construction algorithm: Consider increasingly-coarser grids
    - At each step select the cell(s) c of highest density and move enough randomly-selected points from c into a bucket to make c and its neighbors “close-to-uniform”
    - "Truly multi-dimensional "split decisions" based on tuple density -- unlike WHIST

Other Multi-dimensional Histogram Techniques -- STHoles [BG601]

- Multi-dimensional, workload-based histograms
  - Allow bucket nesting -- "bucket tree"
  - Intersect query result stream and count |q / b| for each bucket b (< 10% overhead in MS SQL Server 2000)
  - Drill "holes" in b for regions of different tuple density and "pull" them out as children of b (first-class buckets)
  - Consolidate/merge buckets of similar densities (keep #buckets constant)

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

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 = 15%, selectivity ranging from 2% to 10%
    - 10% vs. 100% (no answer) error, for size = 0.5%, select. = 3%