Approximate Query Processing: Taming the TeraBytes!

A Tutorial

Minos Garofalakis and *Phillip B. Gibbons*

Information Sciences Research Center

Bell Laboratories

http://www.bell-labs.com/user/{minos, pbgibbons}/

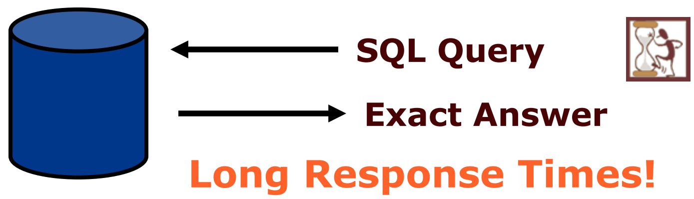
Outline

- 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
  - Using Histograms, Samples, Wavelets
- Discussion & Comparisons
- Advanced Techniques & Future Directions
  - Dependency-based, Workload-tuned, Streaming data
- Conclusions

Introduction & Motivation

Lucent Technologies Bell Labs Innovations

Decision Support Systems (DSS)



Long Response Times!

- 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

Lucent Technologies Bell Labs Innovations

- 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

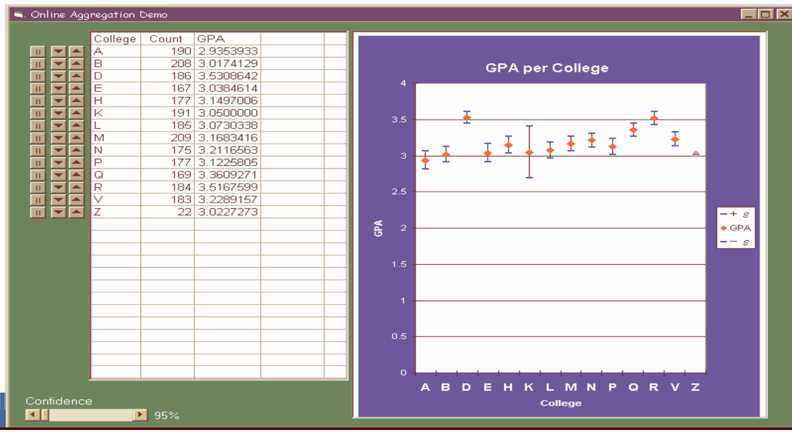
- Achieved by answering the query based on samples or other synopses of the data
- Speed-up obtained because synopses are orders of magnitude smaller than the original data

Approximate Query Answering

Lucent Technologies
Bell Labs Innovations

Basic Approach 1: Online Query Processing

- e.g., Control Project [HHW97, HH99, HAR00]
- Sampling at query time
- Answers continually improve, under user control



Approximate Query Answering

Lucent Technologies
Bell Labs Innovations

Basic Approach 2: Precomputed Synopses

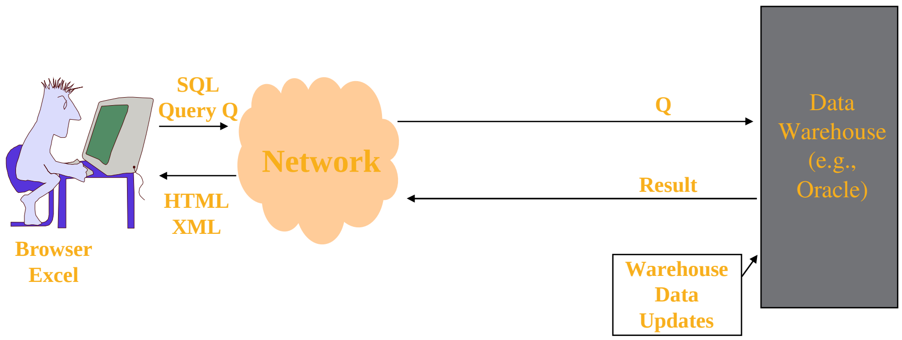
- Construct & store synopses prior to query time
- At query time, use synopses to answer the query

- 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]

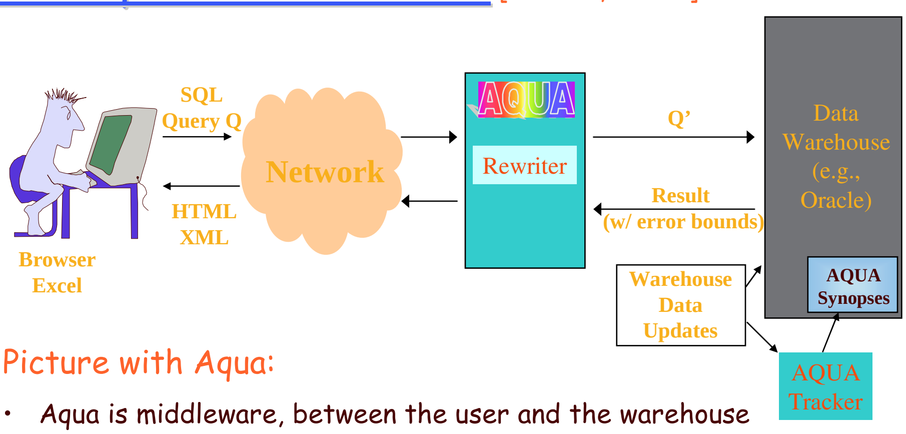
The Aqua Architecture



Picture without Aqua:

- User poses a query Q
- Data Warehouse executes Q and returns result
- Warehouse is periodically updated with new data

The Aqua Architecture [GMP97a, AGP99]



Picture with Aqua:

- Aqua is middleware, between the user and the warehouse
- Aqua Synopses are stored in the warehouse
- Aqua intercepts the user query and rewrites it to be a query Q' on the synopses. Data warehouse returns approximate answer

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
+ See [HH01] Online Query Processing tutorial for details

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

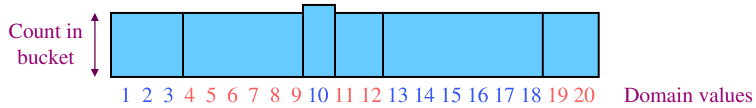
Outline

- Intro & Approximate Query Answering Overview
- One-Dimensional Synopses
  - **Histograms:** Equi-depth, Compressed, 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
- 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

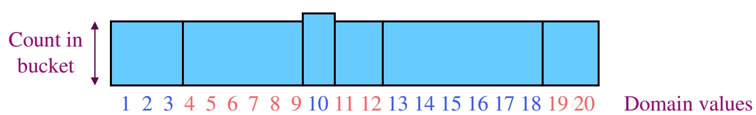


- Goal: Equal number of rows per bucket (B buckets in all)
- Can **construct** by first sorting then taking B-1 equally-spaced splits

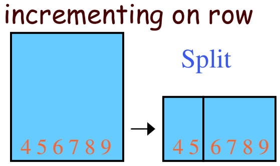
  1 2 2 3 4 7 8 9 10 10 10 10 11 11 12 12 14 16 16 18 19 20 20 20

- **Faster construction:** Sample & take equally-spaced splits in sample
  - Nearly equal buckets
  - Can also use one-pass quantile algorithms (e.g., [GK01])

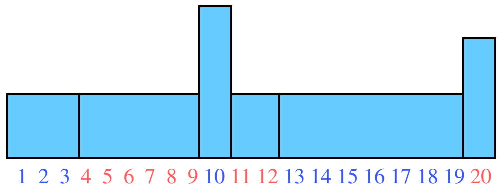
1-D Histograms: Equi-Depth



- 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



1-D Histograms: Compressed



[PIH96]

- Create singleton buckets for largest values, equi-depth over the rest
- Improvement over equi-depth since get exact info on largest values, e.g., join estimation in DB2 compares largest values in the relations

Construction: Sorting + O(B log B) + one pass; can use sample

Maintenance: Split & Merge approach as with equi-depth, but must also decide when to create and remove singleton buckets [GMP97b]

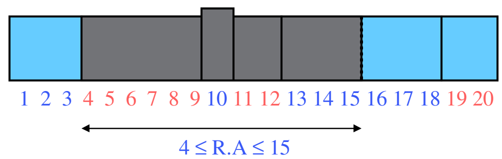
1-D Histograms: V-Optimal

[IP95] defined V-optimal & showed it minimizes the average selectivity estimation error for equality-joins & selections

- Idea: Select buckets to minimize frequency variance within buckets

- [JKM98] gave an O(B*N^2) time dynamic programming algorithm
  - F[k] = freq. of value k; AVGF[i:j] = avg freq for values i..j
  - SSE[i:j] = sum{k=i..j} (F[k]^2 – (j-i+1)*AVGF[i:j]^2)
  - For i=1..N, compute P[i] = sum{k=1..i} F[k] & Q[i] = sum{k=1..i} F[k]^2
  - Then can compute any SSE[i:j] in constant time
  - Let SSEP(i,k) = min SSE for F[1]..F[i] using k buckets
  - Then SSEP(i,k) = min{j=1..i-1} (SSEP(j,k-1) + SSE[j+1:i]), i.e., suffices to consider all possible left boundaries for kth bucket
  - Also gave faster approximation algorithms

Answering Queries: Equi-Depth



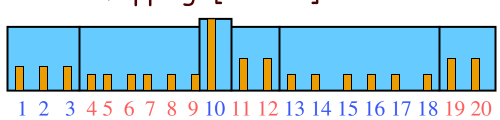
Answering queries:

- 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 * |R|/6 ± 0.5 * |R|/6

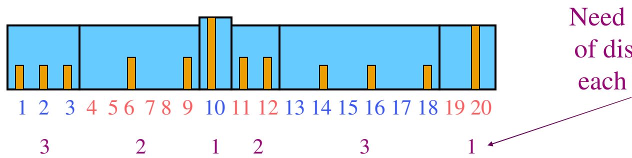
Answering Queries: Histograms

- Answering queries from 1-D histograms (in general):
  - (Implicitly) map the histogram back to an approximate relation, & apply the query to the approximate relation
  - Continuous value mapping [SAC79]:



Count spread evenly among bucket values

  - Uniform spread mapping [PIH96]:

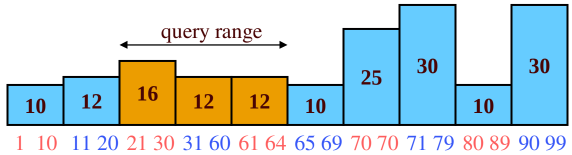


Need number of distinct in each bucket

Self-Tuning 1-D Histograms

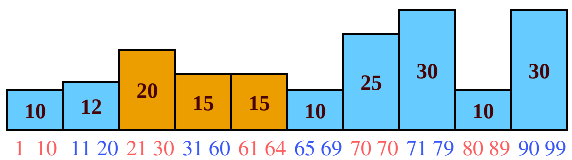
1. Tune Bucket Frequencies: [AC99]

- Compare actual selectivity to histogram estimate
- Use to adjust bucket frequencies



Actual = 60
Estimate = 40
Error = +20

- Divide d*Error proportionately, d=dampening factor



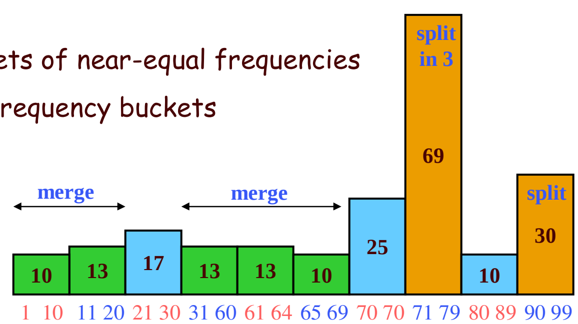
d=½ of Error
= +10
So divide
+4,+3,+3

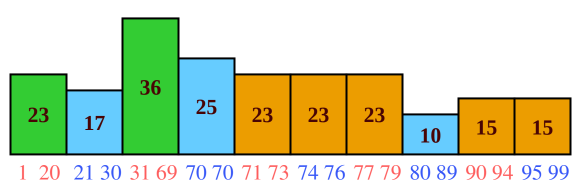
Garofalakis & Gibbons, VLDB 2001 #19

Self-Tuning 1-D Histograms

2. Restructure:

- Merge buckets of near-equal frequencies
- Split large frequency buckets





Also Extends
to Multi-D

Garofalakis & Gibbons, VLDB 2001 #20

Sampling: Basics

Idea: A small random sample S of the data often well-represents all the data

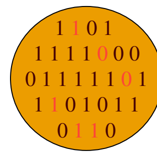
- For a fast approx answer, apply the query to S & "scale" the result
- E.g., R.a is {0,1}, S is a 20% sample

  select count(*) from R where R.a = 0

  ↓

  select 5 * count(*) from S where S.a = 0

  Est. count = 5*2 = 10, Exact count = 10



R.a

Red = in S

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 ± 600 with prob ≥ 90%

Sampling: Confidence Intervals

Method	90% Confidence Interval (±)	Guarantees?
Central Limit Theorem	1.65 * σ(S) / sqrt(|S|)	as |S| → ∞
Hoeffding	1.22 * (MAX-MIN) / sqrt(|S|)	always
Chebychev (known σ(R))	3.16 * σ(R) / sqrt(|S|)	always
Chebychev (est. σ(R))	3.16 * σ(S) / sqrt(|S|)	as σ(S) → σ(R)

Confidence intervals for Average: select avg(R.A) from R

(Can replace R.A with any arithmetic expression on the attributes in R)

σ(R) = standard deviation of the values of R.A; σ(S) = s.d. for S.A

- If predicates, S above is subset of sample that satisfies the predicate
- Quality of the estimate depends only on the variance in R & |S| after the predicate: So 10K sample may suffice for 10B row relation!
  - Advantage of larger samples: can handle more selective predicates

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
  - Need acceptance/rejection sampling to account for the variable number of rows in a page, children in an index node, etc
- 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-Pass Uniform Sampling

- Best choice for incremental maintenance
  - Low overheads, no random data access
- Reservoir Sampling [Vit85]: Maintains a sample S of a fixed-size M
  - Add each new item to S with probability M/N, where N is the current number of data items
  - If add an item, evict a random item from S
  - Instead of flipping a coin for each item, determine the number of items to skip before the next to be added to S
  - To handle deletions, permit |S| to drop to L < M, e.g., L = M/2
    - remove from S if deleted item is in S, else ignore
    - If |S| = M/2, get a new S using another pass (happens only if delete roughly half the items & cost is fully amortized) [GMP97b]

Biased Sampling

Lucent Technologies Bell Labs Innovations

- Often, advantageous to sample different data at different rates (Stratified Sampling)
  - E.g., outliers can be sampled at a higher rate to ensure they are accounted for; better accuracy for small groups in group-by queries
  - Each tuple j in the relation is selected for the sample S with some probability Pj (can depend on values in tuple j)
  - If selected, it is added to S along with its scale factor sf = 1/Pj
  - **Answering queries from S:** e.g.,

    select sum(R.a) from R where R.b < 5 ➡

    select sum(S.a * S.sf) from S where S.b < 5

    - Unbiased answer. Good choice for Pj's results in tighter confidence intervals

R.a	10	10	10	50	50
Pj	1/3	1/3	1/3	½	½
S.sf	---	3	---	---	2

Sum(R.a) = 130

Sum(S.a*S.sf) = 10*3 + 50*2 = 130

One-Dimensional Haar Wavelets

Lucent Technologies Bell Labs Innovations

- 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

Resolution	Averages	Detail Coefficients
3	[2, 2, 0, 2, 3, 5, 4, 4]	----
2	[2, 1, 4, 4]	[0, -1, -1, 0]
1	[1.5, 4]	[0.5, 0]
0	[2.75]	[-1.25]

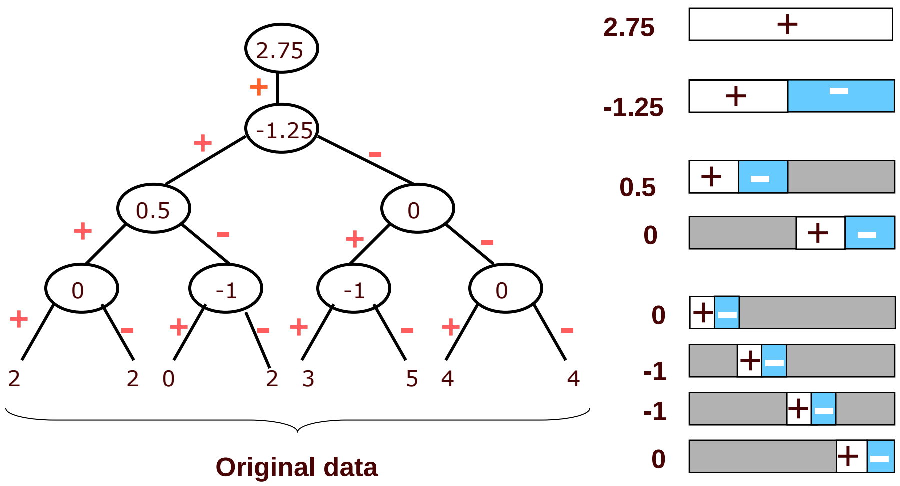
Haar wavelet decomposition: [2.75, -1.25, 0.5, 0, 0, -1, -1, 0]

Haar Wavelet Coefficients

Lucent Technologies Bell Labs Innovations

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

Coefficient "Supports"



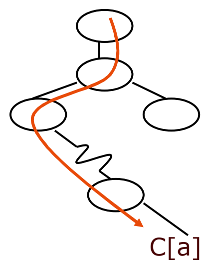
Wavelet-based Histograms [MVW98]

Lucent Technologies Bell Labs Innovations

- Problem: range-query selectivity estimation
- Key idea: use a compact subset of Haar/linear wavelet coefficients for approximating the data distribution
- Steps
  - compute cumulative data distribution C
  - compute Haar (or linear) wavelet transform of C
  - coefficient *thresholding* : only b<<|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

Using Wavelet-based Histograms

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

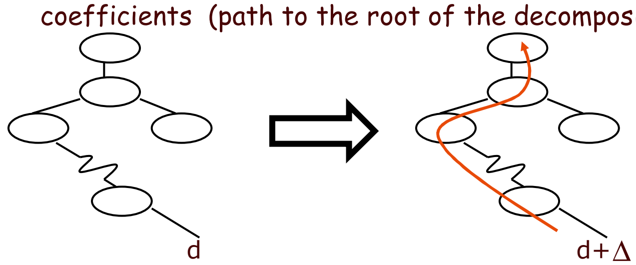


- At most logN+1 coefficients are needed to reconstruct any C value

- Empirical results over synthetic data
  - Improvements over random sampling and histograms (MaxDiff)

Dynamic Maintenance of Wavelet-based Histograms [MVW00]

- Build Haar-wavelet synopses on the original data distribution
  - Similar accuracy with CDF, makes maintenance simpler
- Key issues with dynamic wavelet maintenance
  - Change in single distribution value can affect the values of many coefficients (path to the root of the decomposition tree)

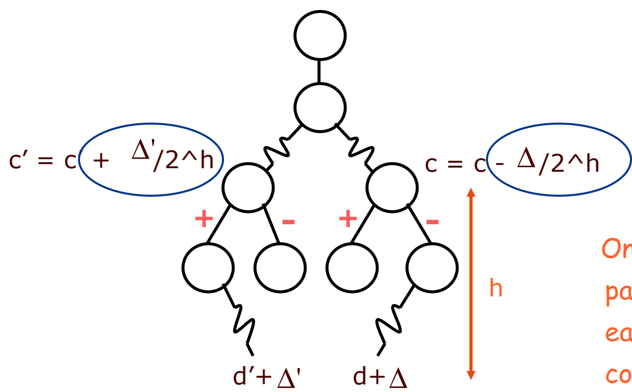


Change propagates up to the root coefficient

  - As distribution changes, "most significant" (e.g., largest) coefficients can also change!
    - Important coefficients can become unimportant, and vice-versa

Effect of Distribution Updates

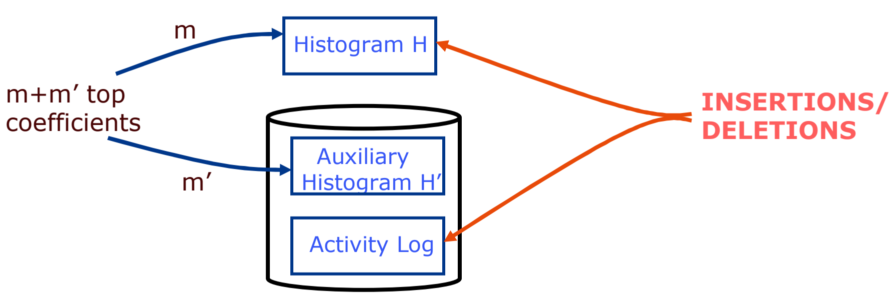
- Key observation: for each coefficient c in the Haar decomposition tree
  - c = (AVG(leftChildSubtree(c)) - AVG(rightChildSubtree(c))) / 2



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

Garofalakis & Gibbons, VLDB 2001 #31

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 the "height" of c (*Probabilistic Counting* [FM85])
  - Adjust H and H′ (move largest coefficients to H)

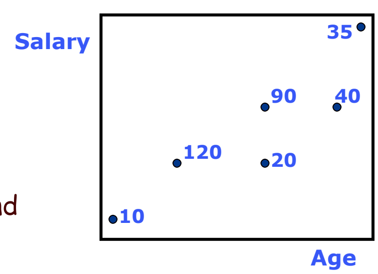
Garofalakis & Gibbons, VLDB 2001 #32

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

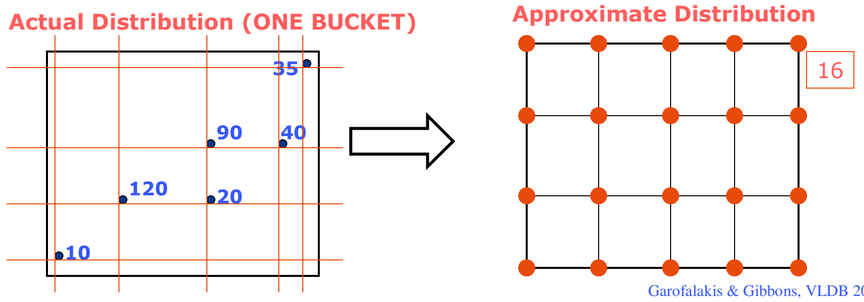


- Conventional approach: Attribute-Value Independence (AVI) assumption
  - sel(p(A1) & p(A2) & . . .) = sel(p(A1)) * sel(p(A2)) * . . .
  - Simple -- one-dimensional marginals suffice
  - BUT: almost always inaccurate, gross errors in practice (e.g., [Chr84, FK97, Poo97]

Multi-dimensional Histograms

Lucent Technologies Bell Labs Innovations

- 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 Ai, F = total bucket frequency
  - approximate data points on a n(1)*n(2)*. . . uniform grid, each with frequency F / (n(1)*n(2)*. . .)



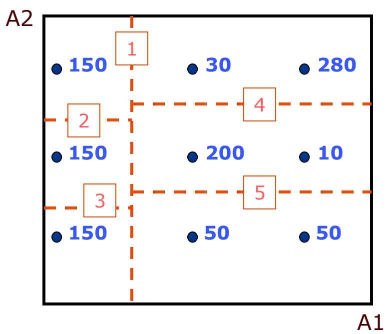
Multi-dimensional Histogram Construction

Lucent Technologies Bell Labs Innovations

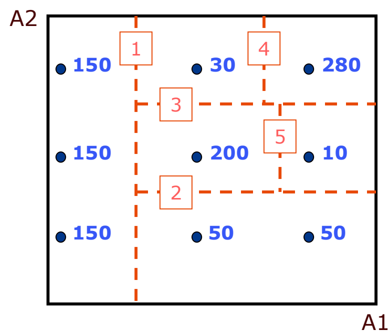
- Construction problem is much harder even for two dimensions [MPS99]
- *Multi-dimensional equi-depth histograms* [MD88]
  - Fix an ordering of the dimensions A1, A2, . . ., Ak, let $\alpha \approx$kth root of desired no. of buckets, initialize B = { data distribution }
  - For i=1, . . ., k: Split each bucket in B in α equi-depth partitions along Ai; return resulting buckets to B
  - Problems: limited set of bucketizations; fixed α and fixed dimension ordering can result in poor partitionings
- *MHIST-p histograms* [PI97]
  - At each step
    - Choose the bucket b in B containing the attribute Ai whose marginal *is the most in need of partitioning*
    - Split b along Ai into p (e.g., p=2) buckets

Equi-depth vs. MHIST Histograms

Equi-depth (a1=2,a2=3) [MD88]



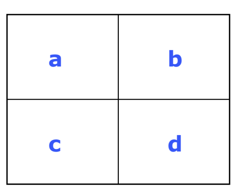
MHIST-2 (MaxDiff) [PI97]

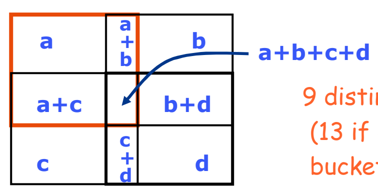


- MHIST: choose bucket/dimension to split based on its *criticality*; allows for much larger class of bucketizations (*hierarchical* space partitioning)
- Experimental results verify superiority over AVI and equi-depth

Other Multi-dimensional Histogram Techniques -- GENHIST [GKT00]

- Key idea: allow for *overlapping* histogram buckets
  - Allows for a much larger no. of distinct frequency regions for a given space budget (= #buckets)



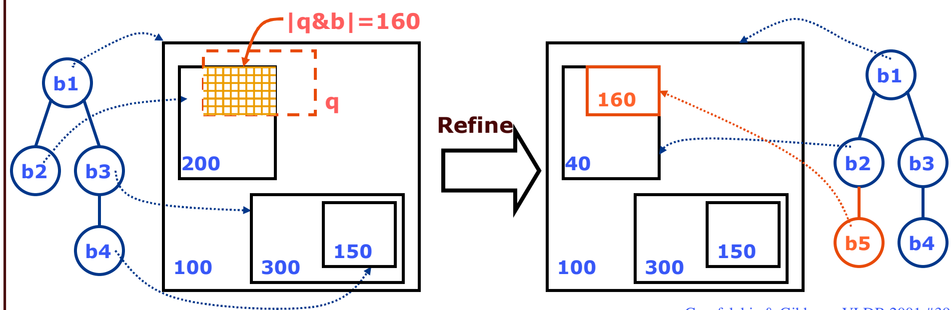


9 distinct frequencies (13 if different-size buckets are used)

- 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 MHIST

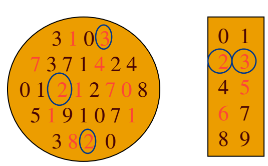
Other Multi-dimensional Histogram Techniques -- STHoles [BCG01]

- Multi-dimensional, workload-based histograms
  - Allow *bucket nesting* -- "bucket tree"
  - Intercept query result stream and count $|q \cap b|$ 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
- 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]

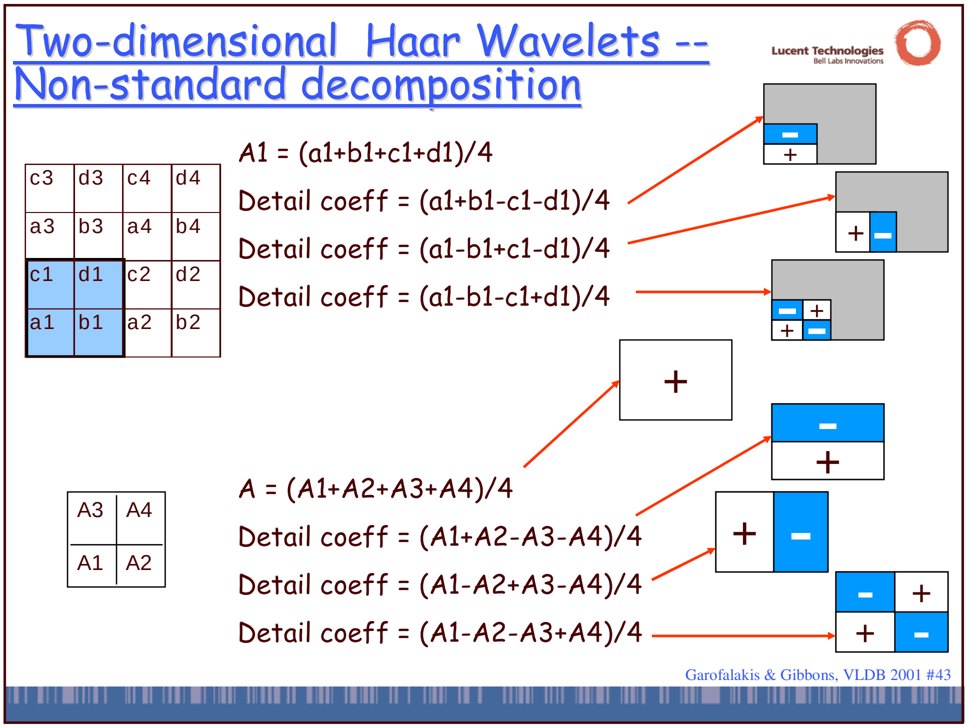
Lucent Technologies Bell Labs Innovations

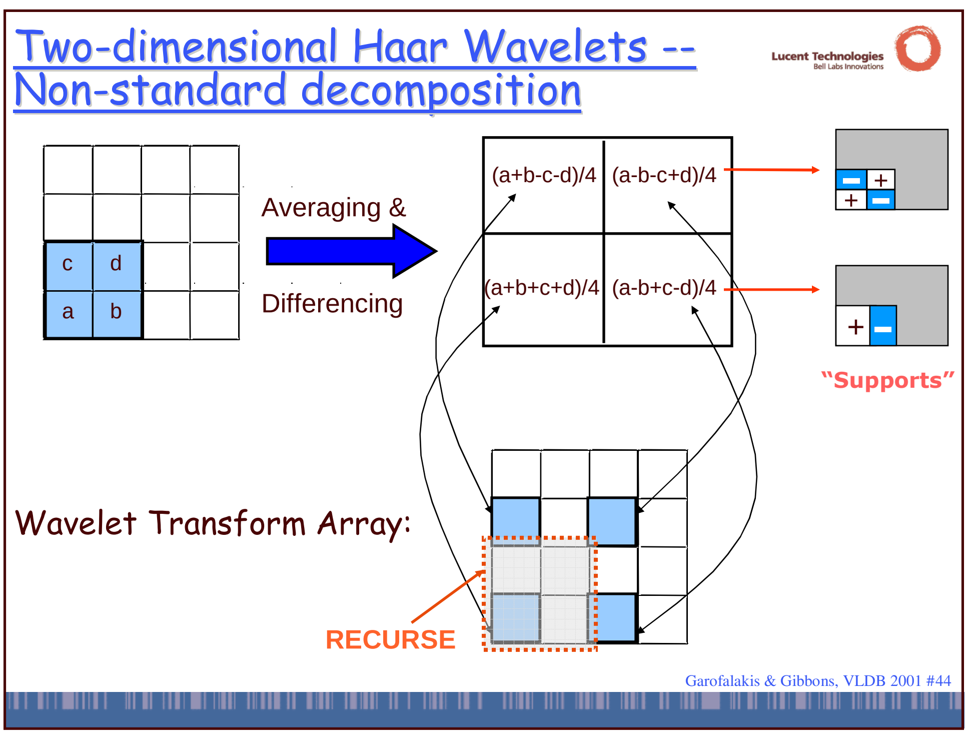
- Based on sampling from materialized foreign key joins
  - Typically < 10% added space required
  - Yet, can be used to get a uniform sample of ANY foreign key join
  - Plus, fast to incrementally maintain
- Significant improvement over using just table samples
  - E.g., for TPC-H query Q5 (4 way join)
    - 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

Lucent Technologies Bell Labs Innovations

- 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
- Discussion here: focus on *non-standard decomposition*
  - See [SDS96, VW99] for more details on standard Haar decomposition
  - [MVW00] also discusses *dynamic maintenance* of standard multi-dimensional Haar wavelet synopses


Two-dimensional Haar Wavelets --
Non-standard decomposition
Lucent Technologies
Bell Labs Innovations
c3 d3 c4 d4
a3 b3 a4 b4
c1 d1 c2 d2
a1 b1 a2 b2
A1 = (a1+b1+c1+d1)/4
Detail coeff = (a1+b1-c1-d1)/4
Detail coeff = (a1-b1+c1-d1)/4
Detail coeff = (a1-b1-c1+d1)/4
A3 A4
A1 A2
A = (A1+A2+A3+A4)/4
Detail coeff = (A1+A2-A3-A4)/4
Detail coeff = (A1-A2+A3-A4)/4
Detail coeff = (A1-A2-A3+A4)/4
Garofalakis & Gibbons, VLDB 2001 #43


Two-dimensional Haar Wavelets --
Non-standard decomposition
Lucent Technologies
Bell Labs Innovations
c d
a b
Averaging &
Differencing
(a+b-c-d)/4
(a-b-c+d)/4
(a+b+c+d)/4
(a-b+c-d)/4
"Supports"
Wavelet Transform Array:
RECURSE
Garofalakis & Gibbons, VLDB 2001 #44

Two-dimensional Haar Wavelets -- Non-standard decomposition

Data Array

3	4	3	4
1	2	1	2
3	4	3	4
1	2	1	2

After averaging and differencing

-1	0	-1	0
2.5	-.5	2.5	-.5
-1	0	-1	0
2.5	-.5	2.5	-.5

After distributing results

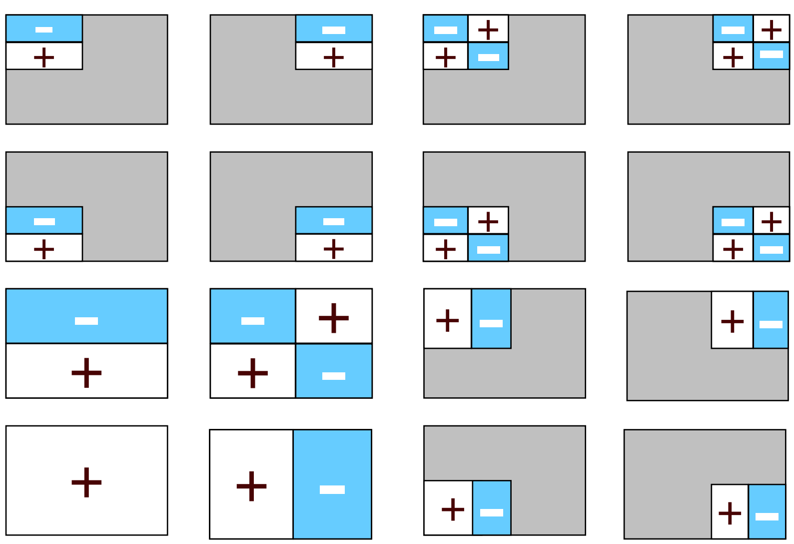
-1	-1	0	0
-1	-1	0	0
2.5	**2.5**	-.5	-.5
2.5	**2.5**	-.5	-.5

RECURSE

Final wavelet transform array

-1	-1	0	0
-1	-1	0	0
0	0	-.5	-.5
2.5	0	-.5	-.5

Non-standard Two-dimensional Haar Basis -- Coefficient Supports



Constructing the Wavelet Decomposition

Joint Data Distribution Array

Attr2 \ Attr1	0	1	2	3
3				
2				
1		6		3
0			4	

Relation (ROLAP) Representation

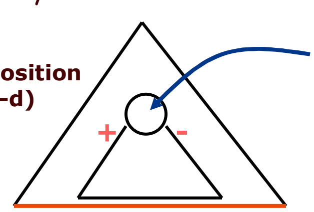
Attr1	Attr2	Count
2	0	4
1	1	6
3	1	3

- Joint data distribution can be very sparse!
- Key to I/O-efficient decomposition algorithms: *Work off the ROLAP representation*
  - Standard decomposition [VW99]
  - Non-standard decomposition [CGR00]
- Typically require a small (logarithmic) number of passes over the data

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]

Decomposition Tree (1-d)



Query Range

Contribution to range sum = 0

Only nodes on the path to range endpoints can have nonzero contributions (Extends naturally to multi-dimensional range sums)

Outline

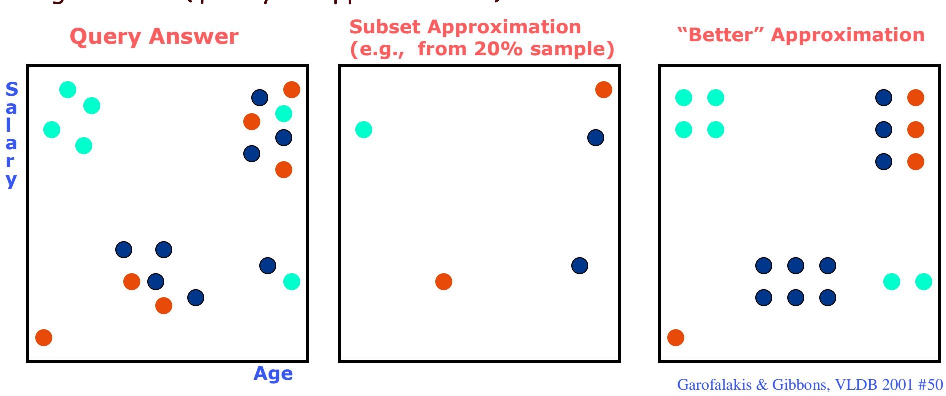
Lucent Technologies
Bell Labs Innovations

- Intro & Approximate Query Answering Overview
- One-Dimensional Synopses
- Multi-Dimensional Synopses and Joins
- Set-Valued Queries
  - Error Metrics
  - Using Histograms
  - Using Samples
  - Using Wavelets
- Discussion & Comparisons
- Advanced Techniques & Future Directions
- Conclusions

Approximating Set-Valued Queries

Lucent Technologies
Bell Labs Innovations

- Problem: Use synopses to produce "good" approximate answers to generic SQL queries -- selections, projections, joins, etc.
  - Remember: synopses try to capture the *joint data distribution*
  - Answer (in general) = *multiset of tuples*
- Unlike aggregate values, NO universally-accepted measures of "goodness" (quality of approximation) exist



Error Metrics for Set-Valued Query Answers

- Need an error metric for (multi)sets that accounts for both
  - differences in element *frequencies*
  - differences in element *values*
- Traditional set-comparison metrics (e.g., symmetric set difference, Hausdorff distance) fail
- Proposed Solutions
  - *MAC (Match-And-Compare) Error [IP99]:* based on perfect bipartite graph matching
  - *EMD (Earth Mover's Distance) Error [CGR00, RTG98]:* based on bipartite network flows

Using Histograms for Approximate Set-Valued Queries [IP99]

- Store histograms as relations in a SQL database and define a *histogram algebra* using simple SQL queries
- Implementation of the algebra operators (select, join, etc.) is fairly straightforward
  - Each multidimensional histogram bucket directly corresponds to a set of approximate data tuples
- Experimental results demonstrate histograms to give much lower MAC errors than random sampling
- Potential problems
  - For high-dimensional data, histogram effectiveness is unclear and construction costs are high [GKT00]
  - Join algorithm requires *expanding* into approximate relations
    - Can be as large (or larger!) than the original data set

Set-Valued Queries via Samples

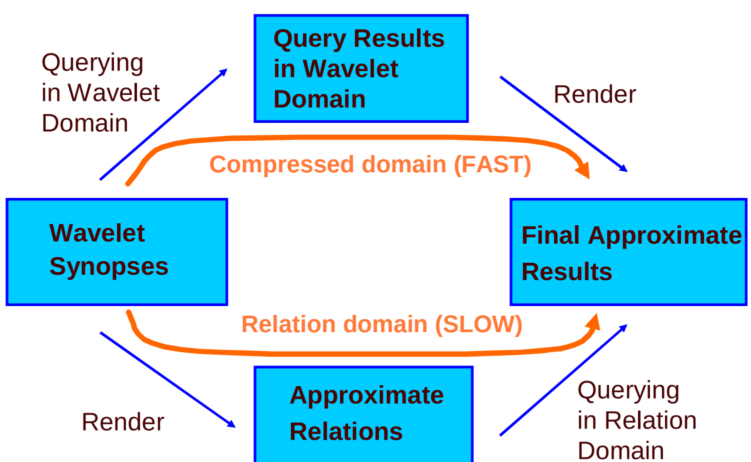
- Applying the set-valued query to the sampled rows, we very often obtain a **subset of the rows in the full answer**
  - E.g., Select all employees with 25+ years of service
  - Exceptions include certain queries with nested subqueries (e.g., select all employees with above average salaries: but the average salary is known only approximately)

- Extrapolating from the sample:
  - Can treat each sample point as the center of a cluster of points (generate approximate points, e.g., using *kernels* [BKS99, GKT00])
  - Alternatively, Aqua [GMP97a, AGP99] returns an approximate count of the number of rows in the answer and a representative subset of the rows (i.e., the sampled points)
    - Keeps result size manageable and fast to display

Approximate Query Processing Using Wavelets [CGR00]

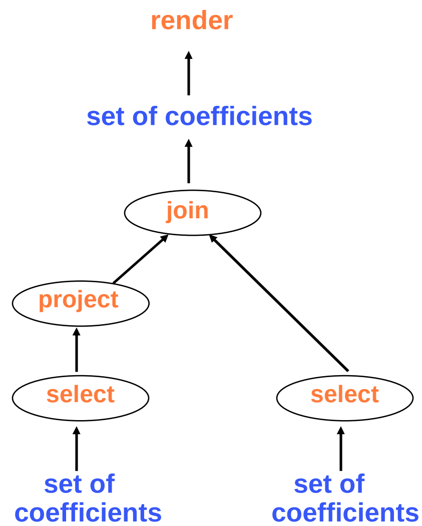
- Reduce relations into compact *wavelet-coefficient synopses*

Entire query processing in the compressed (wavelet) domain



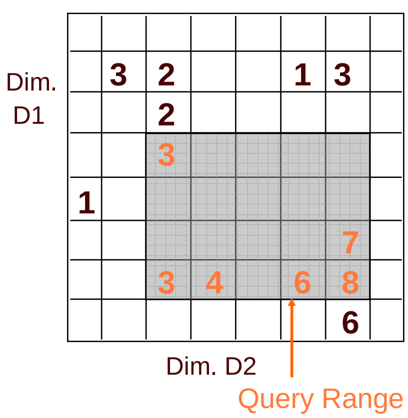
Wavelet Query Processing

- 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

Joint Data Distribution Array

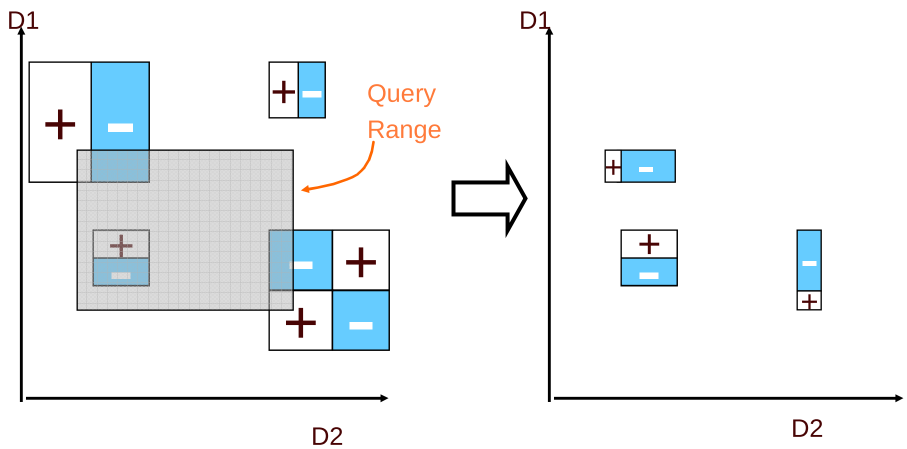


Relation

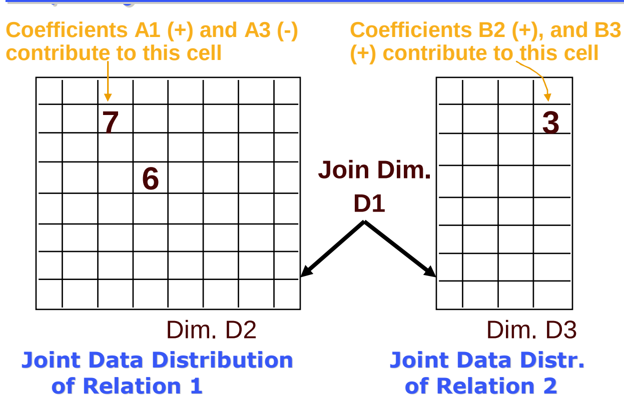
Dim D1 (Attr1)	Dim D2 (Attr2)	Count
0	6	6
1	2	3
1	3	4
1	5	6
1	6	8
2	6	7
3	0	1
4	2	3
5	2	2
6	1	3
6	2	2
6	5	1
6	6	3

- 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



Equi-join -- Relational Domain



Relation 1

Dim D1 (Attr1)	Dim D2 (Attr2)	Count
6	2	7
4	3	6

Relation 2

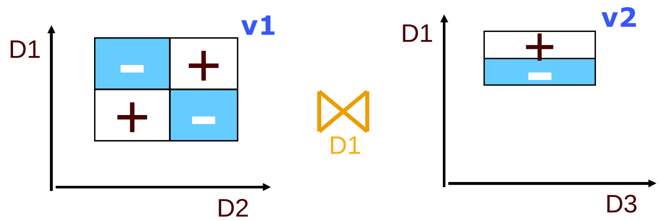
Dim D1 (Attr1)	Dim D3 (Attr3)	Count
6	3	3

Join along D1

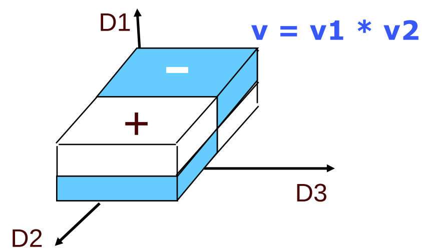
Dim D1 (Attr1)	Dim D2 (Attr2)	Dim D3 (Attr3)	Count
6	2	3	21

- *Relational domain:* Join count= 7*3 = (A1-A3)*(B2+B3)
- *Wavelet domain:* A1*B2 + A1*B3 - A3*B2 - A3*B3
- Consider all pairs of coefficients: (1) check joinability (overlap in join dimension(s)), (2) compute output coefficients

Equi-join -- Wavelet Domain



Join output coefficient:



Outline

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

Discussion & Comparisons (1)

- *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!!
- *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 such 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

Lucent Technologies Bell Labs Innovations

- 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
  - Workload-tuned Biased Sampling
  - Distinct-values Queries
  - Streaming Data
- Conclusions

Dependency-based Histogram Synopses [DGR01]

Lucent Technologies Bell Labs Innovations

Attribute Value Independence
*** simplistic**
*** inaccurate**

Multi-dimensional histograms on joint data distribution
*** expensive**
*** ineffective in high dimensions**

Fully independent attributes ⟷ **Fully correlated attributes**

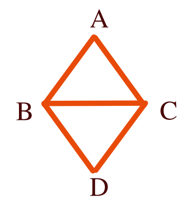
- Extremes in terms of the underlying correlations!!
- Dependency-Based Histograms: explore space between extremes by explicitly identifying data correlations/independences
  - Build a *statistical interaction model* on data attributes
  - Based on the model, build a collection of low-dimensional histograms
  - 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(salary, height, weight) = p(salary) * p(height, weight)
  - Conditional Independence :

    p(salary, age | YPE) = p(salary| YPE) * p(age | YPE)
- Use forward selection to build a *decomposable statistical model* [BFH75], [Lau96] on the attributes
  - A,D are conditionally independent given B,C
    - p(AD|BC) = p(A|BC) * p(D|BC)
  - Joint distribution
    - p(ABCD) = p(ABC) * p(BCD) / p(BC)
  - ***Build histograms on model cliques***



- Significant accuracy improvements (factor of 5) over pure MHIST
- More details, construction & usage algorithms, etc. in the paper ☺

Workload-tuned Biased Sampling -- Congressional Samples [AGP00]

- 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(i) group-by s : seek good accuracy for all 50 states
  - Q: Avg(i) group-by s,g : 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 use 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 -- ICICLES [GLR00]

Lucent Technologies
Bell Labs Innovations

- Biased sampling scheme that *dynamically adapts* to query workload
  - Exploit data locality -- more focus (i.e., #sample points) in frequently-queried regions
  - Let Q = {q1, q2, . . .} be a query workload, R(qi) = subset of R used in answering query qi
  - L(R, Q) = Extension of R wrt Q = $R \bigcup_{qi \in Q} R(qi)$ (multiset of tuples)

Icicle: Uniform random sample of L(R,Q)

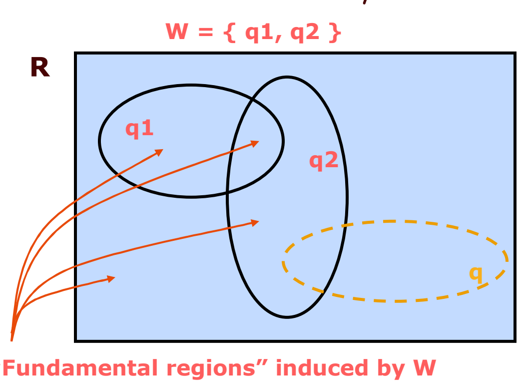
- Incrementally maintained and adapt ("self-tune") to workload through *Reservoir Sampling* technique [Vit85]
- *Unbiased Icicle estimators:* New formulas to account for duplicates and bias in sample selection
- *Provably better* (smaller variance) than uniform for "focused" queries (that follow the workload model)

Garofalakis & Gibbons, VLDB 2001 #67

Workload-tuned Biased Sampling -- Lifted Workloads [CDN01]

Lucent Technologies
Bell Labs Innovations

- 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")



prob(q|W) = parametric function of q's overlap with queries in W

"Fundamental regions" induced by W

Garofalakis & Gibbons, VLDB 2001 #68

Workload-tuned Biased Sampling -- Lifted Workloads

Lucent Technologies
Bell Labs Innovations

- Problem: Find sample of size k that minimizes expected error for a given "lifted" workload
- Solution: *Stratified sampling* [Coc77]
  - Collection of uniform samples (of total size k) over disjoint subsets ("strata") of the population
  - Much better estimates when variance within strata is small [Coc77]
- *Stratification:* Selecting appropriate partitioning of R
  - Using "fundamental regions" as strata is *optimal* for COUNT
  - For SUM, partition "fundamental regions" further to reduce variance of the aggregated attribute (Neymann technique [Coc77])
- *Allocation:* Dividing k among strata
  - Closed form solutions (valid under certain simplifying assumptions)

Distinct Values Queries

Lucent Technologies
Bell Labs Innovations

- select count(distinct target-attr)
- from rel
- where P

Template

- select count(distinct o_custkey)
- from orders
- where o_orderdate >= '2001-01-01'

TPCH example

  - How many distinct customers have placed orders this year?

- Includes: column cardinalities, number of species, number of distinct values in a data set / data stream

Distinct Values Queries

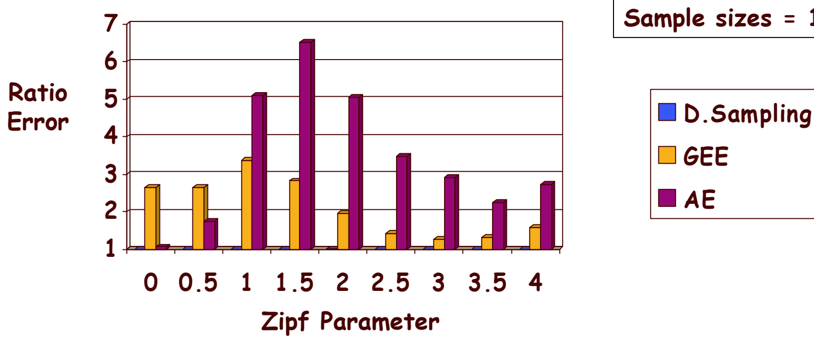
- Uniform Sampling-based approaches
  - Collect and store uniform sample. At query time, apply predicate to sample. Estimate based on a function of the distribution. **Extensive** literature (see, e.g., [CCM00])
    - Many functions proposed, but estimates are often inaccurate
    - [CCM00] proved must examine (sample) almost the entire table to guarantee the estimate is within a factor of 10 with probability > 1/2, regardless of the function used!
- One pass approaches
  - A hash function maps values to bit position according to an exponential distribution [FM85] (cf. [Coh97,AMS96])
    - 00001011111 estimate based on rightmost 0-bit
    - Produces a single count: Does not handle subsequent predicates

Distinct Values Queries

- One pass, sampling approach: Distinct Sampling [Gib01]:
  - A hash function assigns random priorities to domain values
  - Maintains $O(\log(1/\delta)/\varepsilon^2)$ highest priority values observed thus far, and a random sample of the data items for each such value
  - Guaranteed within ε relative error with probability $1 - \delta$
  - Handles ad-hoc predicates: E.g., How many distinct customers today vs. yesterday?
    - To handle q% selectivity predicates, the number of values to be maintained increases inversely with q (see [Gib01] for details)
  - Good for data streams: Can even answer distinct values queries over physically distributed data. E.g., How many distinct IP addresses across an entire subnet? (Each synopsis collected independently!)
  - Experimental results: 0-10% error vs. 50-250% error for previous best approaches, using 0.2% to 10% synopses

Distinct Values Queries

Data set size = 1M
Sample sizes = 1%



Over the entire range of skew :

- Distinct Sampling has 1.00-1.02 ratio error (1.00=no error)
- At least 25 times smaller relative error than best approaches based on uniform sampling (GEE & AE)

Approximate Reports

- Distinct sampling also provides fast, highly-accurate approximate answers for report queries arising in high-volume, session-based event recording environments
- **Environment:** Record events, produce precanned reports
  - Many overlapping sessions: multiple events comprise a session (single IP flow, single call set-up, single customer service call)
  - Events are time-stamped and tagged with session id, and then dumped to append-only databases
  - Logs sent to central data warehouse. Precanned reports executed every minute or hour. TPC-R benchmark
- Must maintain a uniform sample of the sessions & all the events in those sessions in order to produce good approximate reports. Distinct sampling provides this. Improves accuracy by factor of 10+

Data Streams

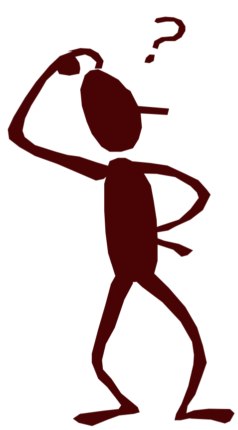
- Data is continually arriving. Collect & maintain synopses on the data. Goal: Highly-accurate approximate answers
  - State-of-the-art: Good techniques for narrow classes of queries
  - E.g., Any one-pass algorithm for collecting & maintaining a synopsis can be used effectively for data streams
- Alternative 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: Altavista
  - 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 [DGR01] or mining-based [BGR01]) synopses
- Improved approximate query processing over continuous data streams
  - see, e.g., [GKS01a], [GKS01b], [GKM01b]

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!!*



http://www.bell-labs.com/user/{minos, pbgibbons}/

References (1)

- [AC99] A. Aboulnaga and S. Chaudhuri. "Self-Tuning Histograms: Building Histograms Without Looking at Data". ACM SIGMOD 1999.
- [AGM99] N. Alon, P. B. Gibbons, Y. Matias, M. Szegedy. "Tracking Join and Self-Join Sizes in Limited Storage". ACM PODS 1999.
- [AGP00] S. Acharya, P. B. Gibbons, and V. Poosala. "Congressional Samples for Approximate Answering of Group-By Queries". ACM SIGMOD 2000.
- [AGP99] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. "Join Synopses for Fast Approximate Query Answering". ACM SIGMOD 1999.
- [AMS96] N. Alon, Y. Matias, and M. Szegedy. "The Space Complexity of Approximating the Frequency Moments". ACM STOC 1996.
- [BCC00] A.L. Buchsbaum, D.F. Caldwell, K.W. Church, G.S. Fowler, and S. Muthukrishnan. "Engineering the Compression of Massive Tables: An Experimental Approach". SODA 2000.
  - Proposes exploiting simple (differential and combinational) data dependencies for effectively compressing data tables.
- [BCG01] N. Bruno, S. Chaudhuri, and L. Gravano. "STHoles: A Multidimensional Workload-Aware Histogram". ACM SIGMOD 2001.
- [BDF97] D. Barbara, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. Ioannidis, H. V. Jagadish, T. Johnson, R. Ng, V. Poosala, K. A. Ross, and K. C. Sevcik. "The New Jersey Data Reduction Report". IEEE Data Engineering bulletin, 1997.

References (2)

- [BFH75] Y.M.M. Bishop, S.E. Fienberg, and P.W. Holland. "Discrete Multivariate Analysis". The MIT Press, 1975.
- [BGR01] S. Babu, M. Garofalakis, and R. Rastogi. "SPARTAN: A Model-Based Semantic Compression System for Massive Data Tables". ACM SIGMOD 2001.
  - 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.
- [BKS99] B. Blohsfeld, D. Korus, and B. Seeger. "A Comparison of Selectivity Estimators for Range Queries on Metric Attributes". ACM SIGMOD 1999.
  - Studies the effectiveness of histograms, kernel-density estimators, and their hybrids for estimating the selectivity of range queries over metric attributes with large domains.
- [CCM00] M. Charlikar, S. Chaudhuri, R. Motwani, and V. Narasayya. "Towards Estimation Error Guarantees for Distinct Values". ACM PODS 2000.
- [CDD01] S. Chaudhuri, G. Das, M. Datar, R. Motwani, and V. Narasayya. "Overcoming Limitations of Sampling for Aggregation Queries". IEEE ICDE 2001.
  - 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.
- [CDN01] S. Chaudhuri, G. Das, and V. Narasayya. "A Robust, Optimization-Based Approach for Approximate Answering of Aggregate Queries". ACM SIGMOD 2001.
- [CGR00] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim. "Approximate Query Processing Using Wavelets". VLDB 2000. (Full version to appear in The VLDB Journal)

References (3)

- [Chr84] S. Christodoulakis. "Implications of Certain Assumptions on Database Performance Evaluation". ACM TODS 9(2), 1984.
- [CMN98] S. Chaudhuri, R. Motwani, and V. Narasayya. "Random Sampling for Histogram Construction: How much is enough?". ACM SIGMOD 1998.
- [CMN99] S. Chaudhuri, R. Motwani, and V. Narasayya. "On Random Sampling over Joins". ACM SIGMOD 1999.
- [CN97] S. Chaudhuri and V. Narasayya. "An Efficient, Cost-Driven Index Selection Tool for Microsoft SQL Server". VLDB 1997.
- [CN98] S. Chaudhuri and V. Narasayya. "AutoAdmin "What-if" Index Analysis Utility". ACM SIGMOD 1998.
- [Coc77] W.G. Cochran. "Sampling Techniques". John Wiley & Sons, 1977.
- [Coh97] E. Cohen. "Size-Estimation Framework with Applications to Transitive Closure and Reachability". JCSS, 1997.
- [CR94] C.M. Chen and N. Roussopoulos. "Adaptive Selectivity Estimation Using Query Feedback". ACM SIGMOD 1994.
  - Presents a parametric, curve-fitting technique for approximating an attribute's distribution based on query feedback.
- [DGR01] A. Deshpande, M. Garofalakis, and R. Rastogi. "Independence is Good: Dependency-Based Histogram Synopses for High-Dimensional Data". ACM SIGMOD 2001.

References (4)

- [FK97] C. Faloutsos and I. Kamel. "Relaxing the Uniformity and Independence Assumptions Using the Concept of Fractal Dimension". JCSS 55(2), 1997.
- [FM85] P. Flajolet and G.N. Martin. "Probabilistic counting algorithms for data base applications". JCSS 31(2), 1985.
- [FMS96] C. Faloutsos, Y. Matias, and A. Silbershcatz. "Modeling Skewed Distributions Using Multifractals and the `80-20' Law". VLDB 1996.
  - Proposes the use of "multifractals" (i.e., 80/20 laws) to more accurately approximate the frequency distribution within histogram buckets.
- [GGM96] S. Ganguly, P.B. Gibbons, Y. Matias, and A. Silberschatz. "Bifocal Sampling for Skew-Resistant Join Size Estimation". ACM SIGMOD 1996.
- [Gib01] P. B. Gibbons. "Distinct Sampling for Highly-Accurate Answers to Distinct Values Queries and Event Reports". VLDB 2001.
- [GK01] M. Greenwald and S. Khanna. "Space-Efficient Online Computation of Quantile Summaries". ACM SIGMOD 2001.
- [GKM01a] A.C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M.J. Strauss. "Optimal and Approximate Computation of Summary Statistics for Range Aggregates". ACM PODS 2001.
  - 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)

- [GKM01b] A.C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M.J. Strauss. "Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries". VLDB 2001.
- [GKT00] D. Gunopulos, G. Kollios, V.J. Tsotras, and C. Domeniconi. "Approximating Multi-Dimensional Aggregate Range Queries over Real Attributes". ACM SIGMOD 2000.
- [GKS01a] J. Gehrke, F. Korn, and D. Srivastava. "On Computing Correlated Aggregates over Continual Data Streams". ACM SIGMOD 2001.
- [GKS01b] S. Guha, N. Koudas, and K. Shim. "Data Streams and Histograms". ACM STOC 2001.
- [GLR00] V. Ganti, M.L. Lee, and R. Ramakrishnan. "ICICLES: Self-Tuning Samples for Approximate Query Answering". VLDB 2000.
- [GM98] P. B. Gibbons and Y. Matias. "New Sampling-Based Summary Statistics for Improving Approximate Query Answers". ACM SIGMOD 1998.
  - 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.
- [GMP97a] P. B. Gibbons, Y. Matias, and V. Poosala. "The Aqua Project White Paper". Bell Labs tech report, 1997.
- [GMP97b] P. B. Gibbons, Y. Matias, and V. Poosala. "Fast Incremental Maintenance of Approximate Histograms". VLDB 1997.

References (6)

- [GTK01] L. Getoor, B. Taskar, and D. Koller. "Selectivity Estimation using Probabilistic Relational Models". ACM SIGMOD 2001.
  - Proposes novel, Bayesian-network-based techniques for approximating joint data distributions in relational database systems.
- [HAR00] J. M. Hellerstein, R. Avnur, and V. Raman. "Informix under CONTROL: Online Query Processing". Data Mining and Knowledge Discovery Journal, 2000.
- [HH99] P. J. Haas and J. M. Hellerstein. "Ripple Joins for Online Aggregation". ACM SIGMOD 1999.
- [HHW97] J. M. Hellerstein, P. J. Haas, and H. J. Wang. "Online Aggregation". ACM SIGMOD 1997.
- [HNS95] P.J. Haas, J.F. Naughton, S. Seshadri, and L. Stokes. "Sampling-Based Estimation of the Number of Distinct Values of an Attribute". VLDB 1995.
  - Proposes and evaluates several sampling-based estimators for the number of distinct values in an attribute column.
- [HNS96] P.J. Haas, J.F. Naughton, S. Seshadri, and A. Swami. "Selectivity and Cost Estimation for Joins Based on Random Sampling". JCSS 52(3), 1996.
- [HOT88] W.C. Hou, Ozsoyoglu, and B.K. Taneja. "Statistical Estimators for Relational Algebra Expressions". ACM PODS 1988.
- [HOT89] W.C. Hou, Ozsoyoglu, and B.K. Taneja. "Processing Aggregate Relational Queries with Hard Time Constraints". ACM SIGMOD 1989.

References (7)

- [IC91] Y. Ioannidis and S. Christodoulakis. "On the Propagation of Errors in the Size of Join Results". ACM SIGMOD 1991.
- [IC93] Y. Ioannidis and S. Christodoulakis. "Optimal Histograms for Limiting Worst-Case Error Propagation in the Size of join Results". ACM TODS 18(4), 1993.
- [Ioa93] Y.E. Ioannidis. "Universality of Serial Histograms". VLDB 1993.
  - 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.
- [IP95] Y. Ioannidis and V. Poosala. "Balancing Histogram Optimality and Practicality for Query Result Size Estimation". ACM SIGMOD 1995.
- [IP99] Y.E. Ioannidis and V. Poosala. "Histogram-Based Approximation of Set-Valued Query Answers". VLDB 1999.
- [JKM98] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. Sevcik, and T. Suel. "Optimal Histograms with Quality Guarantees". VLDB 1998.
- [JMN99] H. V. Jagadish, J. Madar, and R.T. Ng. "Semantic Compression and Pattern Extraction with Fascicles". VLDB 1999.
  - Discusses the use of "fascicles" (i.e., approximate data clusters) for the semantic compression of relational data.
- [KJF97] F. Korn, H.V. Jagadish, and C. Faloutsos. "Efficiently Supporting Ad-Hoc Queries in Large Datasets of Time Sequences". ACM SIGMOD 1997.

References (8)

  - Proposes the use of SVD techniques for obtaining fast approximate answers from large time-series databases.
- [Koo80] R. P. Kooi. "The Optimization of Queries in Relational Databases". PhD thesis, Case Western Reserve University, 1980.
- [KW99] A.C. Konig and G. Weikum. "Combining Histograms and Parametric Curve Fitting for Feedback-Driven Query Result-Size Estimation". VLDB 1999.
  - Proposes the use of linear splines to better approximate the data and frequency distribution within histogram buckets.
- [Lau96] S.L. Lauritzen. "Graphical Models". Oxford Science, 1996.
- [LKC99] J.H. Lee, D.H. Kim, and C.W. Chung. "Multi-dimensional Selectivity Estimation Using Compressed Histogram Information". ACM SIGMOD 1999.
  - Proposes the use of the Discrete Cosine Transform (DCT) for compressing the information in multi-dimensional histogram buckets.
- [LM01] I. Lazaridis and S. Mehrotra. "Progressive Approximate Aggregate Queries with a Multi-Resolution Tree Structure". ACM SIGMOD 2001.
  - Proposes techniques for enhancing hierarchical multi-dimensional index structures to enable approximate answering of aggregate queries with progressively improving accuracy.
- [LNS90] R.J. Lipton, J.F. Naughton, and D.A. Schneider. "Practical Selectivity Estimation through Adaptive Sampling". ACM SIGMOD 1990.
  - Presents an adaptive, sequential sampling scheme for estimating the selectivity of relational equi-join operators.

References (9)

- [LNS93] R.J. Lipton, J.F. Naughton, D.A. Schneider, and S. Seshadri. "Efficient sampling strategies for relational database operators", Theoretical Comp. Science, 1993.
- [MD88] M. Muralikrishna and D.J. DeWitt. "Equi-Depth Histograms for Estimating Selectivity Factors for Multi-Dimensional Queries". ACM SIGMOD 1988.
- [MPS99] S. Muthukrishnan, V. Poosala, and T. Suel. "On Rectangular Partitionings in Two Dimensions: Algorithms, Complexity, and Applications". ICDT 1999.
- [MVW98] Y. Matias, J.S. Vitter, and M. Wang. "Wavelet-based Histograms for Selectivity Estimation". ACM SIGMOD 1998.
- [MVW00] Y. Matias, J.S. Vitter, and M. Wang. "Dynamic Maintenance of Wavelet-based Histograms". VLDB 2000.
- [NS90] J.F. Naughton and S. Seshadri. "On Estimating the Size of Projections". ICDT 1990.
  - Presents adaptive-sampling-based techniques and estimators for approximating the result size of a relational projection operation.
- [Olk93] F. Olken. "Random Sampling from Databases". PhD thesis, U.C. Berkeley, 1993.
- [OR92] F. Olken and D. Rotem. "Maintenance of Materialized Views of Sampling Queries". IEEE ICDE 1992.
- [PI97] V. Poosala and Y. Ioannidis. "Selectivity Estimation Without the Attribute Value Independence Assumption". VLDB 1997.

References (10)

- [PIH96] V. Poosala, Y. Ioannidis, P. Haas, and E. Shekita. "Improved Histograms for Selectivity Estimation of Range Predicates". ACM SIGMOD 1996.
- [PSC84] G. Piatetsky-Shapiro and C. Connell. "Accurate Estimation of the Number of Tuples Satisfying a Condition". ACM SIGMOD 1984.
- [Poo97] V. Poosala. "Histogram-Based Estimation Techniques in Database Systems". PhD Thesis, Univ. of Wisconsin, 1997.
- [RTG98] Y. Rubner, C. Tomasi, and L. Guibas. "A Metric for Distributions with Applications to Image Databases". IEEE Intl. Conf. On Computer Vision 1998.
- [SAC79] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. T. Price. "Access Path Selection in a Relational Database Management System". ACM SIGMOD 1979.
- [SDS96] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin. "Wavelets for Computer Graphics". Morgan-Kauffman Publishers Inc., 1996.
- [SFB99] J. Shanmugasundaram, U. Fayyad, and P.S. Bradley. "Compressed Data Cubes for OLAP Aggregate Query Approximation on Continuous Dimensions". KDD 1999.
  - 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.
- [V85] J. S. Vitter. "Random Sampling with a Reservoir". ACM TOMS, 1985.

References (11)

- [VL93] S. V. Vrbsky and J. W. S. Liu. "Approximate—A Query Processor that Produces Monotonically Improving Approximate Answers". IEEE TKDE, 1993.
  - 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.
- [VW99] J.S. Vitter and M. Wang. "Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets". ACM SIGMOD 1999.

- *This is only a partial list of references on Approximate Query Processing. Further important references can be found, e.g., in the proceedings of SIGMOD, PODS, VLDB, ICDE, and other conferences or journals, and in the reference lists given in the above papers.*

Additional Resources

- **Related Tutorials**
  - [FJ97] C. Faloutsos and H.V. Jagadish. "Data Reduction". KDD 1998.
    - http://www.research.att.com/~drknow/pubs.html
  - [HH01] P.J. Haas and J.M. Hellerstein. "Online Query Processing". SIGMOD 2001.
    - http://control.cs.berkeley.edu/sigmod01/
  - [KH01] D. Keim and M. Heczko. "Wavelets and their Applications in Databases". IEEE ICDE 2001.
    - http://atlas.eml.org/ICDE/index_html
- **Research Project Homepages**
  - The AQUA and NEMESIS projects (Bell Labs)
    - http://www.bell-labs.com/project/{aqua, nemesis}/
  - The CONTROL project (UC Berkeley)
    - http://control.cs.berkeley.edu/
  - The Approximate Query Processing project (Microsoft Research)
    - http://www.research.microsoft.com/research/dmx/ApproximateQP/
  - The Dr. Know project (AT&T Research)
    - http://www.research.att.com/~drknow/