Approximations and QoS Panel

Johannes Gehrke

http://www.cs.cornell.edu/johannes

Department of Computer Science
Cornell University
Semantic Approximations: How?

Thesis: If you approximate, you have to inform the user what this means.
This is hard.

Note 1: It is not clear that users want approximations!

Note 2: There will be competing axes!
(How to combine errors, what are nice properties of such functions that we can use (monotonicity), how do we know what is the right function?)

- **Outline**
  - Aggregates
  - Set-valued results
  - Composing operators
  - Multiple queries
Two Models

- Fast CPU, not enough main memory, and writing to disk is too slow
- Slow CPU, cannot keep up with the rate of arrival
What Works (?) : Approximating Aggregates

- **Problem**: Records of relation R are streaming in -- compute the 2nd frequency moment of attribute R.A, i.e.,

\[ F_2(R.A) = \sum_{1}^{N} (a_i)^2 \text{ where } a_i = \text{frequency( } i\text{-th value of R.A)} \]

\[ F_2(R.A) = \text{COUNT( } R_{\bowtie R}^A \text{ )} \]

(the size of the *self-join* on R.A)

- Exact solution: too expensive, requires O(N) space.
Sketches for 2nd Moment Estimation
[Alon et al.]

**Key intuition:** Define a random variable $X$ that can be easily computed over the stream, such that $E[X] = F^2$ (unbiased) and $\text{Var}[X]$ is small $\rightarrow$ probabilistic guarantees can be given.

**Technique**
- Define a family of 4-wise independent \{-1, +1\} random variables
  \[
  \{\xi_i : i = 1, \ldots, N\}
  \]
- Pseudo-random generator using only $O(\log N)$ space (for seeding)!

Define the random variable $Z = \sum_{i=1}^{N} a_i \xi_i$
- Simple linear projection -- simple to maintain online: just add $\xi_i$ to $Z$ whenever the $i$-th value is observed in the R.A stream

Data stream R.A: 2 0 1 3 1 2 . . . $Z = \xi_0 + 2\xi_1 + 2\xi_2 + \xi_3$
- Define $X = Z^2$
Sketches for 2nd Moment Estimation (Cont.)

- Given this basic X construction, build several iid copies of X and averaging+median-selection to “boost” accuracy and confidence

- Using Chebyshev/Chernoff bounds
  - Build approximation to $F_2$ within a relative error of $\epsilon$ with probability $\geq 1 - \delta$ using only $O(\log N \cdot \log \frac{1}{\delta} \sqrt{\frac{1}{\epsilon^2}})$ space

- Notes:
  - Sketches are one general class of approximation guarantees for aggregates
  - Many other results/query types (quantiles, $L_p$ norms, patterns, periodicities, data cleaning, sliding windows,...)
Aggregate Queries: Remarks

- Computation intensive?
- Multiple joins: Approximation errors go up exponentially, but we can still quantify them
- No additional statistics needed (no multi-dimensional histograms)
- It gets hard very quickly (Group-BY?)
- Somewhat understood?
Approximating Set-Valued Queries

- Problem: All existing synopsis data structures approximate answers to aggregate queries (e.g., sum, count, moments).
- How do we approximate set-valued queries?
- How do we load-shed intelligently?
Error Metrics for Set-Valued Query Answers

- Need an error metric for (multi)sets that accounts for:
  - Differences in record frequencies
  - Differences in record values
  - Differences in record importance (this depends on the query and the application)

- Old and new metrics:
  - MAC (Match-And-Compare)
  - EMD (Earth Mover’s Distance)
  - Symmetric multi-set difference
  - Archive metric
Set-Valued Queries via Samples

- Idea: Use a sample and then “scale” the sample to approximate the query answer.

- How can we scale the sample?
  - Can treat each sample point as the center of a cluster of points and then generate points surrounding the cluster according to some distribution, e.g., using kernels or other models of a cluster.
  - Aqua gives an approximate count of the number of records and a representative subset of the records.
Using Histograms

- Summary data via histograms and perform queries in the histogram space
  - Translate SQL query into relational algebra operations on histograms
  - Implementation of selection, projection, join, etc. is the straightforward implementation on the histograms
  - Each multidimensional histogram bucket corresponds to a set of approximate data records that could be generated using some distributional assumption in the bucket
- Experimental results demonstrate histograms give much lower MAC errors than random sampling

- Problems
  - For high-dimensional data, histograms are not very good (curse of dimensionality) and good histograms are expensive to construct
  - Join operation is expensive as histograms are converted to approximate relations (size can be larger than the original dataset!)
One Possible Approach

- Perform **semantic** loadshedding
  - Define a metric between sets
  - Drop records such that the distance between true query answer and approximate query answer is minimized

- Recall: Set metrics
  - MAC (Match-And-Compare)
  - EMD (Earth Mover’s Distance)
  - Symmetric multi-set difference
  - Archive metric
Symmetric Multi-Set Difference: Static Join

A
\{ tA.join\_attr=v1 \}

B
\{ tB.join\_attr=v1 \}

K(2,3)

K(3,2)

K(4,2)
Symmetric Multi-Set Difference: Window Join

R=1,1,1,3

Fixed memory allocation

S=2,3,1,1

cost

Capacity: 0..1, linear cost

M=2, w=3

Keep in memory
Replace

SWiM 1/9/2003
Open Problems

- Many algorithmic problems in approximations/load shedding
  - Appropriate new/meaningful metrics
  - Designing algorithms that maximize these metrics

- User interface for
  - Specifying approximations
  - Conveying approximations to the user
Composing Operators

We cannot compose approximation operators blindly. Example:

- The join of a sample is not a sample of the join!
  - Remedy: Sample from one relation according to the frequency in another relation
  - What about joins with more relations: Need multi-dimensional histograms
  - Generic negative results (arbitrary detailed statistics, composing two uniform samples → cannot get uniform sample of the join).

- Selections:
  - Relative error of a query is proportional to the inverse square of the selectivity

Aurora drop boxes have to be designed carefully.
Multiple Queries

- If you have multiple queries, you need to allocate your resources between these queries.
- Metrics (for aggregate queries):
  - Reduce max-error
  - Reduce average error
  - Queries with priorities?
  - Reduce variance? Others?
- Space allocation has to take semantics of approximation technique into account.
  - R1.a=R2.a and R1.a=R3.a && R3.b=R2.a
  - Reuse sketch for R1.a: OK
  - Reuse sketch for R1.a and R2.a: CYCLE
Summary

- Aggregate queries
- Set-valued queries
  - Need new metrics for measuring quality for set-valued query answers
  - Need new ways to specify application-specific permissible approximations
  - Need new ways to report what
- Composition of operators
  - Hard problem
  - Feedback and/or statistics are needed
- Multiple queries
- Need ways to specify QoS!

Quality is never an accident; it is always the result of intelligent effort.

*John Ruskin (1819 - 1900)*