Effective computation of biased quantiles over data streams

Graham Cormode
cormode@bell-labs.com

Flip Korn
flip@research.att.com

S. Muthukrishnan
muthu@cs.rutgers.edu

Divesh Srivastava
divesh@research.att.com

Quantiles

Quantiles summarize data distribution concisely.

Given N items, the ϕ–quantile is the item with *rank* ϕN in the sorted order.

Eg. The median is the 0.5-quantile, the minimum is the 0-quantile.

Equidepth histograms put bucket boundaries on regular quantile values, eg 0.1, 0.2...0.9

Quantiles are a robust and rich summary: median is less affected by outliers than mean

Quantiles over Data Streams

Data stream consists of N items in arbitrary order.

Models many data sources eg network traffic, each packet is one item.

Requires linear space to compute quantiles exactly in one pass, $\Omega(N^{1/p})$ in p passes.

ε-approximate computation in sub-linear space

- Φ-quantile: item with rank between $(\Phi-\varepsilon)N$ and $(\Phi+\varepsilon)N$
- [GK01]: insertions only, space $O(1/\varepsilon \log(\varepsilon N))$
- [CM04]: insertions and deletions, space $O(1/\varepsilon \log 1/\delta)$

Biased Quantiles

IP network traffic is very skewed

- Long tails of great interest
- Eg: 0.9, 0.95, 0.99-quantiles of TCP round trip times

Issue: uniform error guarantees

- ε = 0.05: okay for median, but not 0.99-quantile
- ε = 0.001: okay for both, but needs too much space

Goal: support *relative* error guarantees in small space

- Low-biased quantiles: ϕ-quantiles in ranks $\phi(1\pm\varepsilon)N$
- High-biased quantiles: $(1-\phi)$-quantiles in ranks $(1-(1\pm\varepsilon)\phi)N$

Prior Work

Sampling approach given by Gupta and Zane [GZ03] in context of a different problem:

- Keep $O(1/\varepsilon)$ samplers at different sample rates, each keeping a sample of $O(1/\varepsilon^2)$ items
- Total space: $O(1/\varepsilon^3)$, probabilistic algorithm

Uses too much space in practice.

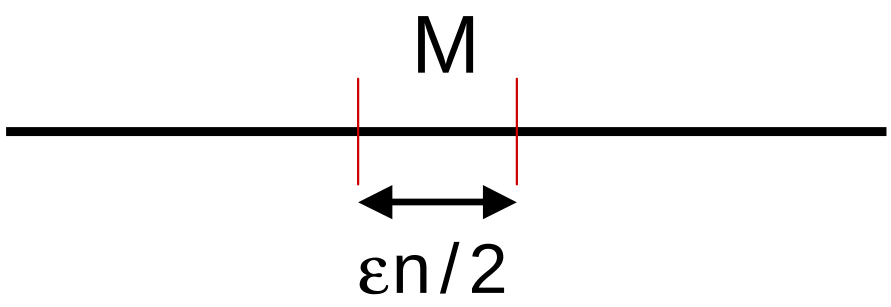
Is it possible to do better? Without randomization?

Intuition

Example shows intuition behind our approach.

Low-biased quantiles: give error $\epsilon\phi$ on ϕ-quantiles

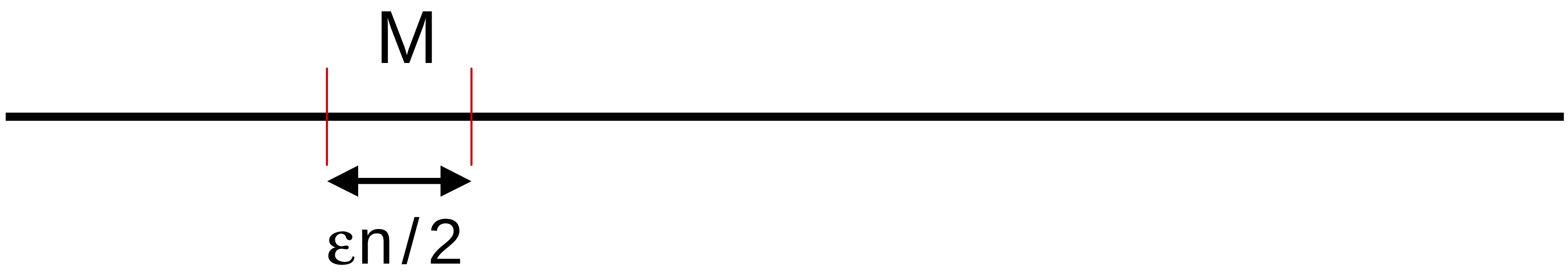
- Set ϵ=10%. Suppose we know approximate median of n items is M — so absolute error is $\epsilon n/2$



- Then there are n inserts, all above M
- M is now the first quartile, so we need error $\epsilon N/4$

Intuition

How can error bounds be maintained?



– Total number of items is now N=2n, so required absolute error bound is for M is still εn/2

Error bound never shrinks too fast, so we can hope to guarantee relative errors.

Challenge is to guarantee accuracy in small space

Space for Biased Quantiles

Any solution to the Biased Quantiles problem must use space at least $\Omega(1/\varepsilon \log(\varepsilon N))$

Shown by a counting argument, there are $\Omega(1/\varepsilon \log(\varepsilon N))$ possible different answers based on choice of ϕ

For uniform quantiles, corresponding lower bound is $\Omega(1/\varepsilon)$ — biased quantiles problem is strictly harder in terms of space needed.

Our Approach

A deterministic algorithm that guarantees relative error for low-biased or high-biased quantiles

Three main routines:

- Insert(v) — inserts a new item, v
- Compress — periodically prune data structure
- Output(ϕ) — output item with rank $(1\pm\varepsilon)\phi N$

Similar structure to Greenwald-Khanna algorithm [GK01] for uniform quantiles ($\phi\pm\varepsilon$), but need new implementation and analysis.

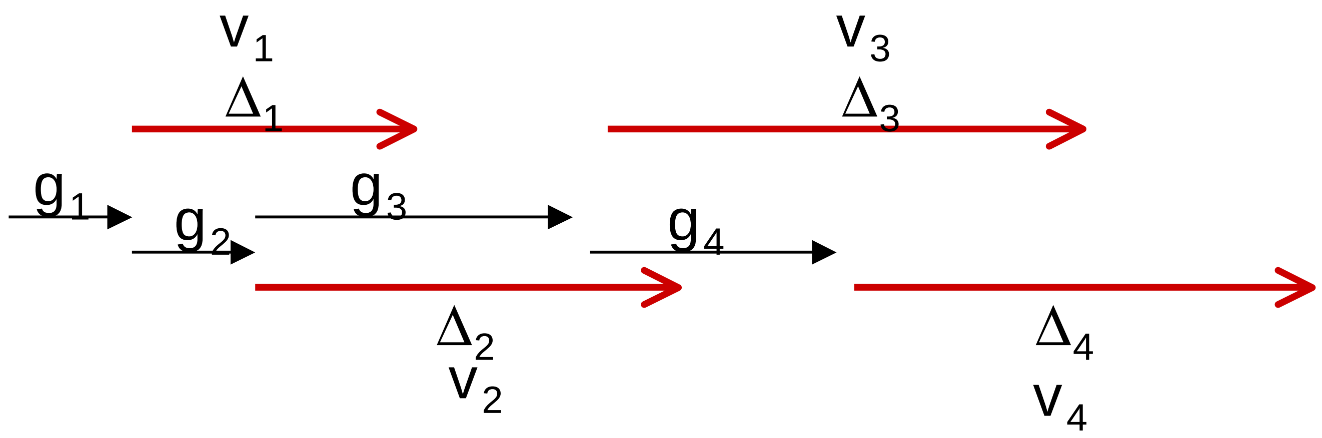
Data Structure

Store tuples $t_i = (v_i, g_i, \Delta_i)$ sorted by v_i

- v_i is an item from the stream
- $g_i = r_{min}(v_i) - r_{min}(v_{i-1})$
- $\Delta_i = r_{max}(v_i) - r_{min}(v_i)$

Define $r_i = \sum_{j=1}^{i-1} g_j$

We will guarantee that the true rank of v_i is between $r_i + g_i$ and $r_i + g_i + \Delta_i$



Biased Quantiles Invariant

In order to guarantee accurate answers, we maintain at all times for all i:

$$\underbrace{g_i + \Delta_i}_{\text{“uncertainty” in rank of } v_i} \leq \max \{\underbrace{2\varepsilon r_i, 1}_{2\varepsilon \text{ times lower bound on rank of } v_i}\}$$

Intuitively, if the uncertainty in rank is proportional to ε times a lower bound on rank, this should give required accuracy

Output Routine

Output(ϕ):

$r_0 := 0$;

for i := 1 to s do

  $r_i := r_{i-1} + g_{i-1}$;

  if $(r_i + g_i + \Delta_i > (\phi n + \epsilon \phi n))$

    print(v_{i-1}); break;

Compute r_i

Output *previous* item, v_{i-1}

max rank of v_i

Upper bound on allowed rank

Claim: Output(ϕ) correctly outputs ε–approximate ϕ-biased quantile

Proof

i is the smallest index such that

$$r_i + g_i + \Delta_i > \phi n + \varepsilon\phi n \quad (*)$$

So $r_{i-1} + g_{i-1} + \Delta_{i-1} \leq (1 + \varepsilon)\phi\, n$. [+]

Using the invariant on (*), $(1 + 2\varepsilon)r_i > (1+\varepsilon)\phi n$ and (rearranging) $r_i > (1-\varepsilon)\phi n$. [-]

Since $r_i = r_{i-1} + g_{i-1}$, we combine [-] and [+]:

[-] $(1-\varepsilon)\phi n < r_{i-1} + g_{i-1}$

$\leq$ (true rank of v_{i-1}) $\leq$

$r_{i-1} + g_{i-1} + \Delta_{i-1} \leq (1+\varepsilon)\phi n$ [+]

Inserting a new item

We must show update operations maintain bounds on the rank of v_i and the BQ invariant

To insert a new item, we find smallest i such that $v < v_i$

- Set $g = 1$ (rank of v is at least 1 more than v_{i-1})
- Set $\Delta = \max\{2\varepsilon r_i,1\}-1$ (uncertainty in rank at most one less than $\Delta_i \leq \max\{2\varepsilon r_i,1\}$)
- Insert (v,g,Δ) before t_i in data structure

Easy to see that Insert maintains the BQ invariant

Compressing the Data Structure

Insert(v) causes data structure to grow by one tuple per update. Periodically we can Compress the data structure by pruning unneeded tuples.

Merge tuples $t_i = (v_i, g_i, \Delta_i)$ and $t_{i+1} = (v_{i+1}, g_{i+1}, \Delta_{i+1})$ together to get $(v_{i+1}, g_i + g_{i+1}, \Delta_{i+1})$.

$\Rightarrow$ Correct semantics of g and Δ

Only merge if $g_i + g_{i+1} + \Delta_{i+1} \leq \max\{2\varepsilon r_i, 1\}$

$\Rightarrow$ Biased Quantiles Invariant is preserved

k-biased Quantiles

Alternate version: sometimes we only care about, eg, $\phi = \frac{1}{2}, \frac{1}{4}, \ldots \frac{1}{2}^k$

Can reduce the space requirement by weakening the Biased Quantiles invariant:

k-BQ invariant:

$$g_i + \Delta_i \leq 2\varepsilon \max\{r_i, \phi^k n, \varepsilon/2\}$$

Implementations were based on the algorithm using this invariant.

Experimental Study

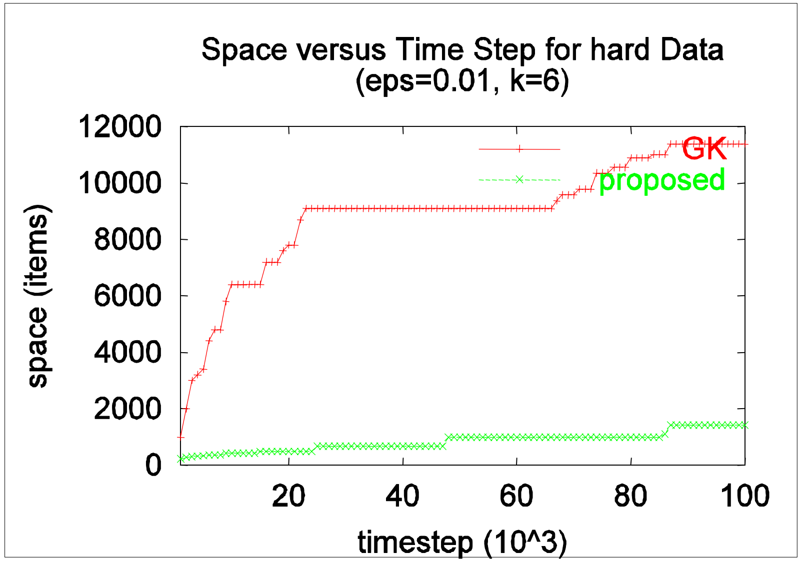
The k-biased quantiles algorithm was implemented in the Gigascope data stream system.

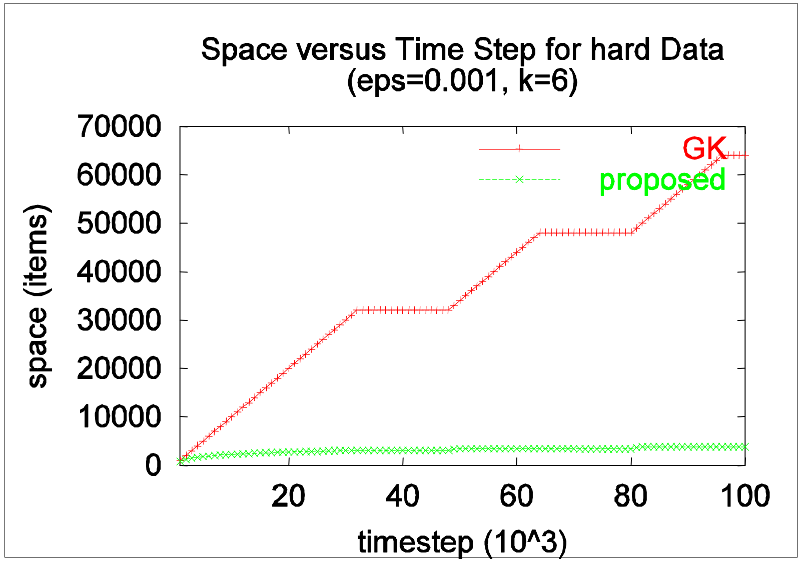
Ran on a mixture of real (155Mbs live traffic streams) and synthetic (1Gbs generated traffic) data.

Experimented to study:

- Space Cost
- Observed accuracy for queries
- Update Time Cost

Experiments: Space Cost

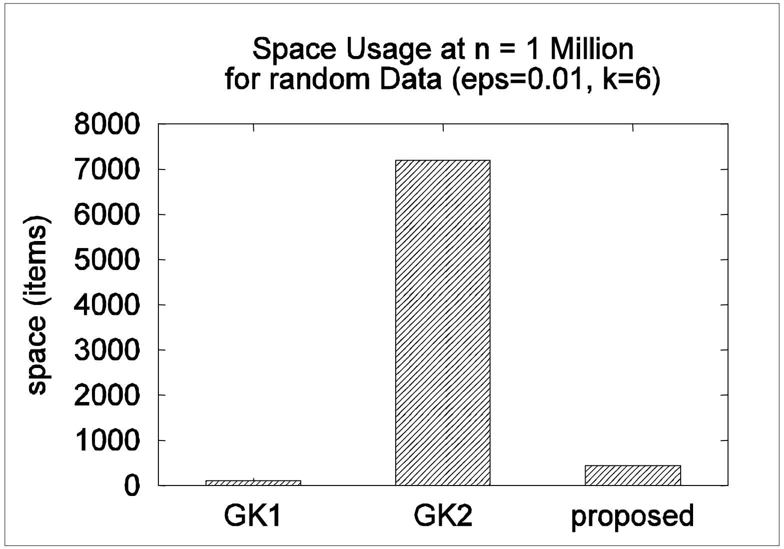


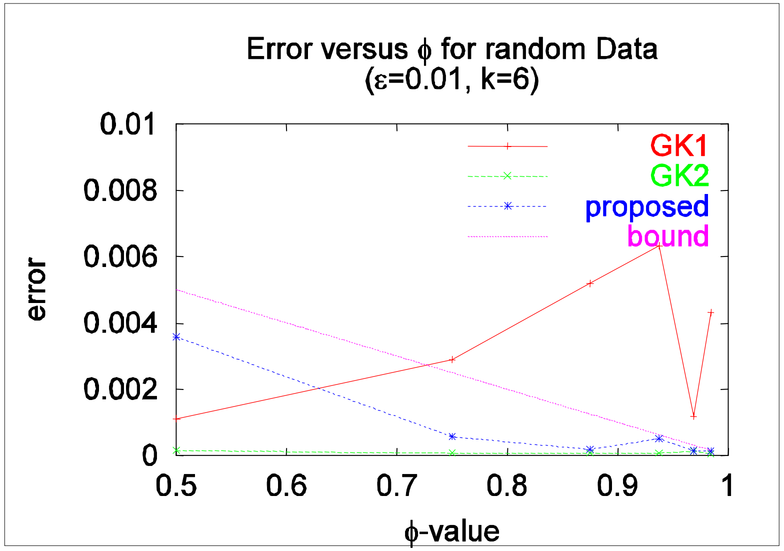


k-biased quantiles, vs. GK with ε = eps ϕ^k

$\Rightarrow$ Space usage scales roughly as $k/\varepsilon \log^c \varepsilon N$ on real data, but grows more quickly in worst case.

Experiments: Accuracy





GK1: ε = eps

GK2: ε = eps ϕ^k

Good tradeoff between space and error on real data

Experiments: Time Cost

Overhead per packet was about 5 – 10μs

Few packet drops (<1%) at Gigabit ethernet speed.

Choice of data structure to implement the list of tuples was an important factor.

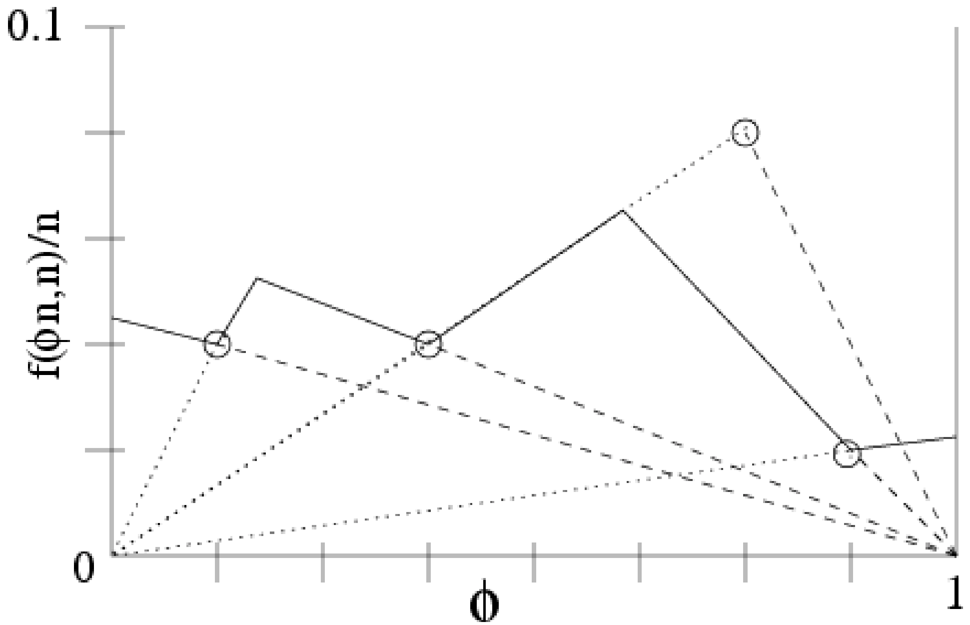
- running compress periodically is blocking operation. Instead, do a partial compression per update
- “cursor” + sorted list (5μs / packet) does better than balanced tree structure (22μs / packet)

Extension: Targeted Quantiles

Further generalization: before the data stream, we are given a set T of (ϕ,ε) pairs.

We must be able to answer ϕ-quantile queries over data streams with error $\pm\varepsilon n$.

From T, generate new invariant f(r,n) to maintain:



In paper, we show that maintaining $g_i + \Delta_i \leq f(r_i,n)$ guarantees targeted quantiles with required accuracy.

Deletions

For uniform quantile guarantees, can handle item deletions in probabilistic setting [CM04].

But, provably need linear space for biased quantiles (with a strong “adversary”), even probabilistically

Sliding window also requires large space.

Conclusions

Skew is prevalent in many realistic situations

Biased Quantiles give a non-uniform way to study skewed data.

We have given efficient algorithms to find biased quantiles over streams of data using small space.

Many other tasks can benefit from incorporating skew either into the problem, or into the analysis of the solution.