Approximate Quantile Sketching

David Lin Kewei
linkewei@stanford.edu

Jensen Jinhui Wang
wangjh97@stanford.edu

July 5, 2022

Ranks and Quantiles

Given $x_1, ..., x_n$ in an ordered universe $\mathcal{U}$, the *rank* of x is

$$R(x) = \# \text{ of } x_i \leq x \ , \tag{1}$$

and the *quantile* of x is

$$Q(x) = \frac{R(x)}{n} \ . \tag{2}$$

For example, if $x_1, x_2, x_3 = [1, 5, 9]$, $R(3) = 1$ and $R(7) = 2$ so $Q(3) = \frac{1}{3}$ and $Q(7) = \frac{2}{3}$.

Problem Statement

Definition (Single quantile approximation)

Given $x_1, ..., x_n \in \mathcal{U}$ in a streaming fashion, find an approximate (random) rank function $\tilde{R}$, such that:

For any item x, $\tilde{R}(x)$ approximates the true rank $R(x)$ to within $\pm\varepsilon n$ (additively) with probability at least $1 - \delta$.

Equivalence to Finding Quantiles

If $\tilde{R}(x)$ is a (ε, δ)-single quantile approximation that is non-decreasing in x, we can find an approximate value of the α-quantile $\tilde{x}_\alpha$ such that $Q(\tilde{x}_\alpha) \in [\alpha - \varepsilon, \alpha + \varepsilon]$.

- Algorithm: Return any $\tilde{x}_\alpha$ such that $\tilde{Q}(\tilde{x}_\alpha) = \alpha$.
- Applying the single quantile approximation guarantee to $x_{\alpha \pm \varepsilon}$ satisfying $Q(x_{\alpha \pm \varepsilon}) = \alpha \pm \varepsilon$, $\tilde{Q}(x_{\alpha - \varepsilon}) \leq \alpha$ and $\tilde{Q}(x_{\alpha + \varepsilon}) \geq \alpha$ with probability $\geq 1 - 2\delta$.
- Non-decreasing $\tilde{Q}$ implies $Q(\tilde{x}_\alpha) \in [\alpha - \varepsilon, \alpha + \varepsilon]$.

$\tilde{x}_\alpha$ can be found efficiently in the sketch we will present.

Problem Significance

Quantiles:

- Natural method of summarizing non-parameteric distributions.
- For example, $\frac{1}{2}$-quantile or the median is widely known to be a statistic robust to outliers.
- Useful for hypothesis testing and outlier detection.
- All-quantile version of the problem yields cdf and hence essentially compresses a distribution!

Summary of Results

Memory lower bounds (for comparison-based algorithms):

- Deterministic: $\Omega((1/\varepsilon)\log(n\varepsilon))$ [CV20]
- Randomized: $\Omega((1/\varepsilon)\log\log(1/\delta))$ [KLL16]

Deterministic sketches:

- MRL Sketch: $O((1/\varepsilon)\log^2(n\varepsilon))$ [MRL99]
- *GK Sketch:* $O((1/\varepsilon)\log(n\varepsilon))$ [GK01]

Randomized sketches:

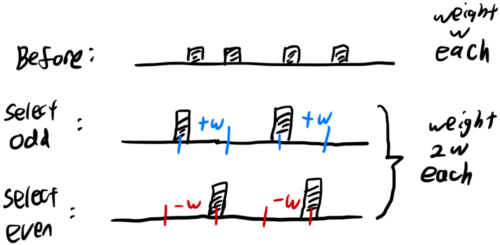
- MRL + Subsampling: $O((1/\varepsilon)\log(1/\varepsilon))$ [MRL99]
- *KLL Sketch:* $O((1/\varepsilon)\log\log(1/\delta))$ [KLL16]–presented today

MRL Sketch and Compactors

The KLL sketch is based on the MRL sketch, whose fundamental building block is a *compactor*. A size-k compactor taking in elements of weight w can either:

- Store k elements in a sorted order.
- Compact and output $\frac{k}{2}$ elements (at even or odd indices) of weight $2w$.

Each compaction operation introduces at most w rank error.



Iterated Compactors and Analysis

MRL sketch:

- Stream elements have weight 1 and fed into compactor 1, whose output is fed into compactor 2, and so on...
- Number of compactors $H = \lfloor \log(n/k) \rfloor + 1$.
- h-th compactor has weight $w_h = 2^{h-1}$ such that since the total weight is "conserved", it processes at most $\frac{n}{w_h}$ elements and compacts $m_h = \frac{n}{w_h k}$ times.

$$\begin{aligned}\text{(Worst-case) Error} &\leq \sum_{h=1}^{H} m_h w_h \\ &\leq H \cdot (n/k) \\ &\lesssim \frac{n}{k} \log\left(\frac{n}{k}\right) .\end{aligned}$$

Setting $k = O((1/\varepsilon)\log(\varepsilon n))$ gives $\leq \varepsilon n$ error and a deterministic sketch with $kH = O((1/\varepsilon)\log^2(\varepsilon n))$ space.

Optimizations

The multi-layer compactor algorithm has a space complexity of $O(\frac{1}{\varepsilon}\log^2(\varepsilon n))$. How do we do better?

- Odd/even randomness
- Compactor size decay
- Replacing the largest layers with a GK sketch

Odd/even randomness

Idea. during compaction operations, pick "discard odd/even" uniformly at random.

Effect. rank either doesn't change, or changes by $\mathrm{Unif}\{\pm\text{weight}\}$ independently.

Cancellation effect across compaction operations!

Odd/even randomness

Lemma (Hoeffding, restated)

Let S be a linear combination of independent Rademacher variables. Then, with probability $1-\delta$,

$$|S| \leq \sqrt{Var(S) \cdot \log \frac{1}{\delta}}.$$

Hence instead of adding up the errors, we add up the squared errors.

Odd/even randomness

$$
\begin{aligned}
(\text{Error}) &\le \sqrt{\sum_{h=1}^{H} m_h w_h^2} \cdot \sqrt{\log \frac{1}{\delta}} \\
&\lesssim \frac{n}{k} \log \frac{1}{\delta}
\end{aligned}
$$

since $m_h w_h$ exponentially increasing. ($m_h w_h = n/k$, $w_H \approx n/k$.)

Progress:

- Error: $\frac{n}{k} \log \frac{n}{k} \to \frac{n}{k} \sqrt{\log \frac{1}{\delta}}$.
- Space: no change.

Compactor size decay

Intuition. With limited memory, we rather keep track of heavy items than light ones.

Idea. Let compactor size decay exponentially, starting from the last layer.

$$k_h \approx \left(\frac{2}{3}\right)^{H-h} k_H$$

Compactor size decay

Why? Use $m_h = \frac{n}{k_h w_h}$:

$$(\text{Error}) \leq \sqrt{n \sum_{h=1}^{H} \frac{w_h}{k_h}} \cdot \sqrt{\log(1/\delta)}$$

$$(\text{Space}) \leq \sum_{h=1}^{H} k_h$$

Making the bottom sum exponential gives space $k_H \log(n/k) \to k_H$!

Compactor size decay

One small caveat... we need $k_h \geq 2$ for all h, so even with exp. decay $\sum_{h=1}^{H} k_h = O(k_H + H)$.

Solution. Actually, a compactor of size 2 is just doing sampling! N compactors of size 2 chained together are just selecting one item unif. at random from 2^N consecutive items.

Use resevoir sampling to simulate uniform choice.

This gets us $\sum_{h=1}^{H} k_h = O(k_H)$!

Progress:

- Error: $nk_H\sqrt{\log \frac{1}{\delta}} \leq \varepsilon n$.
- Space: $k_H = O((1/\varepsilon)\sqrt{\log(1/\delta)})$. Constant in n!

Last few layers

Now we have correct growth rate in $1/\varepsilon$ and independence from n. Where is the space complexity coming from? *The last few compactors.*

We want to knock down the space complexity in terms of δ...

Idea. Intercept the items h^* compactors before the end. (The effect is a compressed stream.) Feed the rest into a deterministic stream (e.g. GK sketch).

Last few layers

For the truncated compactor sequence: n items $\to k \cdot 2^{h^*}$ items.

$$\text{(Error)} \lesssim \left(\frac{2}{\sqrt{3}}\right)^{h^*} \frac{n}{k_H}$$
$$\text{(Space)} \lesssim \left(\frac{2}{3}\right)^{h^*} k_H$$

Intuition: both are exponential sums, so shrinks exponentially with truncated layers.

Last few layers

Add in the GK sketch: for $n' = k_H \cdot 2^{h^*}$ items of weight 2^{H-h^*} and error $\varepsilon' n' \times 2^{H-h^*} \asymp \varepsilon' n$, require $O(1/\varepsilon \log(\varepsilon n'))$ memory.

$$\text{(Error)} \lesssim \left(\frac{2}{\sqrt{3}}\right)^{h^*} \frac{n}{k_H} + \varepsilon' n$$

$$\text{(Space)} \lesssim \left(\frac{2}{3}\right)^{h^*} k_H + \frac{1}{\varepsilon'} \log\left(\varepsilon' k_H 2^{h^*}\right)$$

To set error $\leq \varepsilon n$, we need

$$k_H \leq \frac{(2/\sqrt{3})^{h^*}}{\varepsilon - \varepsilon'} \sqrt{\log(1/\delta)}$$

Last few layers

Set $\varepsilon' = \varepsilon/2$ and plug this back in:

$$(\text{Space}) \lesssim \frac{1}{\varepsilon}\left(\left(\frac{4}{3\sqrt{3}}\right)^{h^*}\sqrt{\log(1/\delta)} + h^* + \log\log(1/\delta)\right) \quad (3)$$

Tradeoff at $h^* \asymp \log\log(1/\delta)$ (phew!) to get the optimal space usage of $O((1/\varepsilon)\log\log(1/\delta))$.

Matching lower bound

Theorem ([HT10])

Any **deterministic, comparison-based** *algorithm that solves the single quantile ε-approximation problem for all streams of length $\Omega((1/\varepsilon)^2 \log(1/\varepsilon)^2)$ must store at least $\Omega((1/\varepsilon)\log(1/\varepsilon))$ stream elements.*

Proof is long and involved. Easy reduction to use this to prove the matching lower bound.

Matching lower bound

Suppose a randomized algorithm can solve quantile approximation with probability $1-\delta$ using $o((1/\varepsilon)\log\log(1/\delta))$ space.

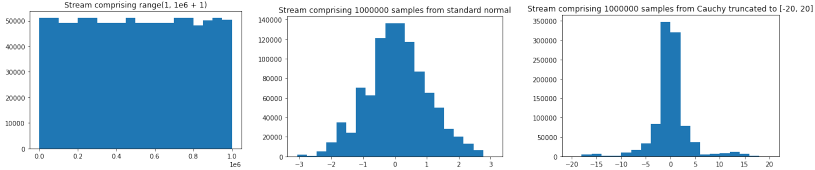
- Set $\delta = 1/(2n)!$, then said algorithm solves the problem for all $n!$ possible streams of n items with probability $1/2$.
- i.e., we can set a seed and have a *deterministic* algorithm that solves the problem for all $n!$ possible streams of length n.
- Space usage: $o((1/\varepsilon)\log n)$, length n stream.
- Set $n = \Theta((1/\varepsilon)^2 \log(1/\varepsilon)^2))$, but space usage is $o((1/\varepsilon)\log(1/\varepsilon))$, contradiction!

Experiments

Below, $n = 1000000$.

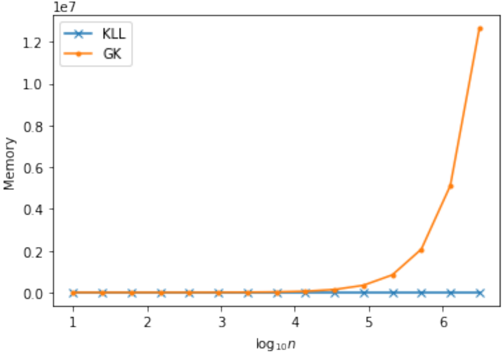
Stream	Quantile Error
[1, ..., n]	0.0006636 ± 0.0004346
n samples from $\mathcal{N}(0, 1)$	0.01591 ± 0.01125
n samples from standard Cauchy	0.01746 ± 0.01008

Table: Experimental quantile query errors for all quantile sketch with parameters $\varepsilon = 0.05$ and $\varepsilon\delta = 0.05 \times 0.05$. The queries were 1) linspace(1, 1000000, 50), 2) linspace(-3, 3, 50), and 3) linspace(-10, 10, 50).



Approximate histograms reconstructed for three streams in Table.

Experimental Memory Cost



Memory cost with $\log_{10}$ number of stream elements.

Thank you!

Any questions?

References

[CV20] Graham Cormode and Pavel Veselỳ. "A tight lower bound for comparison-based quantile summaries". In: *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2020, pp. 81–93.

[GK01] Michael Greenwald and Sanjeev Khanna. "Space-efficient online computation of quantile summaries". In: *ACM SIGMOD Record* 30.2 (2001), pp. 58–66.

References (cont.)

[HT10] Regant Y. S. Hung and Hingfung F. Ting. "An $\Omega(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ Space Lower Bound for Finding ϵ-Approximate Quantiles in a Data Stream". In: *Frontiers in Algorithmics*. Ed. by Der-Tsai Lee, Danny Z. Chen, and Shi Ying. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 89–100. ISBN: 978-3-642-14553-7.

[KLL16] Zohar Karnin, Kevin Lang, and Edo Liberty. "Optimal quantile approximation in streams". In: *2016 ieee 57th annual symposium on foundations of computer science (focs)*. IEEE. 2016, pp. 71–78.

[MRL99] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G Lindsay. "Random sampling techniques for space efficient online computation of order statistics of large datasets". In: *ACM SIGMOD Record* 28.2 (1999), pp. 251–262.