CS 395T: Sublinear Algorithms, Fall 2020 09/15/2020

Lecture 6: Quantile Estimation by Sampling

Prof. Eric Price *Scribe: Lucas Gretta*

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

1 Overview

In the last lecture we discussed lower bounds for indexing and common concentration inequalities.

In this lecture we will discuss quantile estimation by sampling.

2 Quantile Estimation

2.1 Problem Statement

We are given a stream of distinct elements $\{y_i\}_{i=1}^n$. Let the sorted (in ascending order) version be $\{x_i\}_{i=1}^n$. We have three queries we want to answer:

- $Rank(x)$: Return r such that $x_r \leq x \leq x_{r+1}$.
- $Select(r)$: Return x_r
- $Quantile(\alpha)$: Return $x_{\alpha n}$

Performing these queries needs all elements, so we wish to approximate answers.

- $Rank(x)$: Returns r such that $x_{r-\epsilon n} \leq x \leq x_{r+\epsilon n}$.
- $Select(r)$: Returns x_i such that $r - \epsilon n \leq i \leq r + \epsilon n$
- $Quantile(\alpha)$: Returns $x_{\beta n}$ such that $\alpha - \epsilon \leq \beta \leq \alpha + \epsilon$

Note that *Select* is basically the same as *Quantile* ($Quantile(\alpha) = Select(\alpha n)$)

2.2 Random Sampling

The most obvious thing to do is randomly sample our sequence, and then perform each of the queries on the sampled sequence. This runs into a small complication: we don't know how long our stream is.

To solve this we perform **Reservoir Sampling**. We maintain a random sample S with $|S| = m$. Whenever a new element arrives, we include it with probability $\min(1, \frac{m}{i})$. Adding an element to S would make it larger than m, we eject a random element from S initially. This results in a uniform sample of size m over our stream.

Proof. We proceed by induction. The case up to $i = m$ is obviously uniform as it includes all elements. Suppose we have proved the statement for $k \in \mathbb{Z}^+, k \geq m$. Then each of the previous elements has probability $\frac{m}{k}$ of being in our sample. We select our $k+1$th element with probability $\frac{m}{k+1}$, and each previous element in our sample has probability $\frac{m}{k}(1 - \frac{1}{k+1}) = \frac{m}{k+1}$ of being chosen, so by induction the sample is uniform for all n. □

2.3 Bounding Rank's Failure Probability

Lemma 1. *The element $x_{\alpha n}$ of the true quantile α will have empirical quantile $\hat{\alpha}$ in S within $\alpha \pm \epsilon$ with probability $1 - 2e^{-\Omega(\epsilon^2 m)}$.*

Proof. It is easier if we sample with replacement (which can be done with modified reservoir sampling). We want to show that $\hat{x}_{(\alpha-\epsilon)m} \leq x_{\alpha n} \leq \hat{x}_{(\alpha+\epsilon)m}$. Rephrasing this, if it is not the case that $x_{\alpha n} > \hat{x}_{(\alpha+\epsilon)m}$ and not the case that $x_{\alpha n} < \hat{x}_{(\alpha-\epsilon)m}$, our statment holds. Let the sampled values be $\{w\}_i^m$, and $Z_i = 1$ if $w_i \leq x_{\alpha n}$. Then Z_i is a Bernoulli random variable with $p = \alpha$. Then our empirical quantile is $\frac{1}{m}\sum_{i=1}^m Z_i$. Then our chance of error is $Pr[|\sum_{i=1}^m Z_i - \alpha m| \geq \epsilon m] \leq 2e^{-2(\epsilon m)^2/m} = 2e^{-2(\epsilon)^2 m}$ (by a Chernoff bound). □

This solves *Rank* as it is highly likely that our empirical quantile is a good enough approximation for our true quantile, so we can calculate x's empirical quantile q and return nq.

2.4 Bounding Quantile's Failure Probability

We showed above that given some true quantile α, our empirical quantile $\hat{\alpha}$ is ϵ-close. Now we want to show given some empirical quantile $\hat{\alpha}$, α is ϵ-close. This is true if both true $\alpha + \epsilon$ has empirical quantile $> \alpha$, and true $\alpha - \epsilon$ has empirical quantile $< \alpha$. We have already proved this with the above lemma, and so using a union bound we get a failure probability of $\leq 4e^{-2\epsilon^2 m}$, which can be improved to $\leq 2e^{-2\epsilon^2 m}$, using one sided concentration inequalities.

2.5 Failure for Multiple Queries

If we set $m = \frac{1}{2\epsilon^2}\log(\frac{2}{\delta})$ we get δ failure for one query. But what about multiple queries?

We note that if we are accurate on $x_{\epsilon n}, x_{2\epsilon n}, \cdots, x_n$, then all x_i will be 2ϵ accurate. Therefore, we set $m = \frac{4}{2\epsilon^2}\log(\frac{4}{\epsilon\delta})$ for ϵ-accuracy on all inputs.

2.6 Next Time

We will show next time that we can get a $m = O(\frac{1}{\epsilon}\log^2(n))$ bound.