Sublinear Time and Space Algorithms 2016B – Lecture 4 Precision Sampling and High Frequency Moments*

Robert Krauthgamer

1 Precision Sampling

Sum Estimation: Suppose the input is $a_1, \ldots, a_n \in [0,1]$, and we want to estimate its sum $S = \sum_i a_i$ using only a "partial reading" of the a_i's.

The Subsampling Model: Read only a random subset $J \subset [n]$ of size $|J| = m$, and output $\tilde{S} = \frac{n}{m} \sum_{j \in J} a_j$.

We analyze instead sampling elements from $[n]$ with replacement, i.e., J is a multiset. Then $\mathbb{E}[\tilde{S}] = S$ and

$$\mathrm{Var}(\tilde{S}) \leq \frac{n^2}{m^2} \sum_{j \in J} 1 = \frac{n^2}{m}.$$

(In fact, this is just like averaging of m copies of a basic estimator, which samples one element and scales it by n, with standard deviation n.) By Chebyshev's inequality $\Pr[\tilde{S} \in S \pm 2n/\sqrt{m}] \geq 3/4$. For example, to achieve additive error $O(1)$ we need $m = \Omega(n)$.

Exer: Prove similar bounds for subsampling m elements without replacement, and also for subampling each element independently with probability m/n.

Exer: Show that $\Omega(n)$ samples are really needed, even if we allow both additive error 10 and multiplicative error 1.1.

Hint: Consider S with $O(1)$ nonzeros.

The "Precision" Model: The algorithm gets "noisy readings" $\hat{a}_i$ for every a_i. The algorithm chooses in advance (non-adaptively) some precisions u_i and then it is guaranteed additive approximation $|\hat{a}_i - a_i| \leq u_i$. The algorithm's cost is the "total precision" $\frac{1}{n} \sum_i \frac{1}{u_i}$.

Comparison with subsampling explains the scaling by $\frac{1}{n}$: no information about item i means $u_i = 1$ and costs $\frac{1}{nu_i} = 1/n \approx 0$, and nearly-full information means $u_i = 1/n$ and costs $\frac{1}{n} \cdot n = 1$.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

Idea: Choose the u_i's at random (iid).

Precision Sampling Lemma [Andoni, Krauthgamer and Onak, 2011]:

Fix an integer $n \geq 2$, and consider iid $u_1, \ldots, u_n \sim \text{Exp}(1)$ (called precisions). Then for every $a_1, \ldots, a_n \in [0,1]$, and estimates $\hat{a}_1, \ldots, \hat{a}_n \in [0,1]$ that satisfy $|\hat{a}_i - a_i| \leq u_i$, the estimator $\hat{S} = \max_i \hat{a}_i / u_i$ satisfies

$$\Pr_{u_i}\left[\tfrac{1}{4}S - 1 \leq \hat{S} \leq 4\,S + 1\right] \geq 3/4.$$

Moreover, with high probability, the PSL estimator has total cost $O(\log n)$.

Remarks: Exp(1) is the continuous distribution with pdf e^{-x} on $(0, \infty)$. Intuition: its discrete analogue is the geometric distribution; indeed, both are memoryless.

Proof: Was seen in class, using the fact that the exponential distribution is min-stable.

Exer: Can you improve the multiplicative error to $1 + \varepsilon$? How would it increase the estimator cost? Can you guarantee additive error ε by changing the requirement from $\hat{a}_i$?

Hint: Use independent repetitions.

2 High Frequency Moments

Let $x \in \mathbb{R}^n$ be the frequency vector of the input stream.

Theorem [Indyk and Woodruff, 2005]: For every $p \in (2, \infty)$, one can estimate $normx_p^p$ within factor $1 + \varepsilon$ [with high constant probability] using a linear sketch of size (dimension) $s = O(n^{1-2/p}(\frac{1}{\varepsilon}\log n)^{O(1)})$. It implies a streaming algorithm using $O(s \log n)$ bits of storage.

We will see a different algorithm that relies on Precision Sampling, due to [Andoni, Krauthgamer and Onak, 2011]. We will see in class a simplified version, due to Andoni, that achieves only $O(1)$ approximation, and omits discussion of randomness (how to replace full independence with limited independence).

Algorithm PSLsketch:

1. Init: set $w = O(n^{1-2/p} \log^{O(1)} n)$ and pick a random hash function $h : [n] \to [w]$
2. pick independent signs $r_1, \ldots, r_n \in \{\pm 1\}$ and random $u_1, \ldots, u_n \sim \text{Exp}(1)$
3. Update: maintain vector $S = [S_1, \ldots, S_w]$ where $S_j = \sum_{i:h(i)=j} r_i x_i / u_i^{1/p}$.
4. Output: to estimate $\|x\|_p^p$ report $\max_{j \in [w]} |S_j|^p$

The sketch S is linear, hence can be updated easily.

Storage requirement: $O(w \log n)$ bits, not counting storing the randomness.

Correctness:

To use the PSL, let $a_i = |x_i|^p$, then $\sum_i a_i = \|x\|_p^p$, and let $\hat{a}_i = |S_{h(i)}|^p \cdot u_i$.

If we show that WHP for every $i \in [n]$,

$$|\frac{\hat{a}_i}{u_i} - \frac{a_i}{u_i}| \le \varepsilon \|x\|_p^p,$$

then we can use the PSL (the range $a_i \in [0,1]$ needs to be scaled by $\|x\|_p^p$, which is equivalent to dividing all a_i's by $\|x\|_p^p$, but the algorithm need not know this quantity.)

The additive error is further scaled by factor ε, hence by the PSL, WHP the algorithm's estimate is

$$\max_{j \in [w]} |S_j|^p = \max_i \frac{\hat{a}_i}{u_i} \in \max_i \frac{a_i}{u_i} \pm \varepsilon \|x\|_p^p \subseteq [1/4, 4] \sum_i a_i \pm \varepsilon \|x\|_p^p = [1/4 - \varepsilon, 4 + \varepsilon] \|x\|_p^p.$$

We saw in class the following weaker bound.

Lemma: For every $i \in [n]$, WHP

$$\left| S_{h(i)} - r_i x_i / u_i^{1/p} \right|^p \le \varepsilon \|x\|_p^p.$$

Proof of lemma: Was seen in class. It uses the norm-comparison inequality $\|x\|_2 \le n^{1/2-1/p} \|x\|_p$, which follows from Holder's inequality.

Remark: Holder's inequality actually asserts that for all $p, q \in [1, \infty]$ satisfying $1/p + 1/q = 1$,

$$\forall a, b \in \mathbb{R}^n, \qquad \langle a, b \rangle \le \|a\|_p \|b\|_q.$$

Notice that it generalizes the Cauchy-Schwartz inequality.