Sublinear Time and Space Algorithms 2022B – Problem Set 2

Robert Krauthgamer

Due: May 23, 2022

General instructions: Please keep your answers short and easy to read. You can use results, calculations or notation seen in class without repeating them, unless asked explicitly to redo them.

1. Analyze the algorithm below for the Distinct Elements problem. Compared to Algorithm FM+ seen in class, the idea here is to use only one hash function (not k), but still store k hash values.

   Algorithm Bottom k

   1. Init: $z_1 = \cdots = z_k = 1$ (or let each $z_i = i$ to make them distinct) for $k = O(1/\varepsilon^2)$ and a hash function $h : [n] \to [0, 1]$

   2. When item $i \in [n]$ is seen, update $z_1 < \cdots < z_k$ to be the k smallest distinct values among $\{z_1, \ldots, z_k, h(i)\}$

   3. Output: $X := k/z_k$

   Specifically, you should bound the algorithm's space requirement, excluding random coins, and you need to prove that with high probability it reports a $(1 + \varepsilon)$-approximation to d^*. You may assume the hash function h is completely random.

   Remark: You might need to add separate handling for the case $k \le \mathrm{d}^*$.

   Hint: The event $X > (1 + \varepsilon)\,\mathrm{d}^*$ is equivalent to $z_k < \frac{k}{(1+\varepsilon)\,\mathrm{d}^*}$, which implies that at least k hashes are smaller than some threshold. Since each of the d^* distinct hash values is below the threshold independently with probability $\frac{k}{(1+\varepsilon)\,\mathrm{d}^*}$, we expect only $\frac{k}{1+\varepsilon} \approx (1 - \varepsilon)k$ hashes to meet the threshold. Furthermore, this count has standard deviation $\le \varepsilon k$, and we can use Chebyshev's inequality.

2. Let A be a 0-1 matrix of size $(2^k - 1) \times k$ whose rows are all the possible nonzero vectors $A_i \in \{0, 1\}^k$. For a random $p \in \{0, 1\}^k$, define $h_p : [2^k - 1] \to \{0, 1\}$ by $h_p(i) := (Ap)_i = \langle A_i, p\rangle$, where all operations are performed modulo 2.

   (a) Prove that the family $H = \{h_p : p \in \{0, 1\}^k\}$ is pairwise independent. Then state explicitly the performance of this construction (number of bits needed to generate/store $n = 2^k$ pairwise independent random bits $h(1), \ldots, h(n)$).

   (b) Prove that this construction generates k-wise independent bits whenever the matrix A satisfies that every k rows are linearly independent (over the field $GF[2]$).

   (c) Show how to modify the matrix A above, by adding just one extra column, so that now every 3 rows are linearly independent.

3. Show that algorithm AMS (for estimating ℓ_2 norm) works even if the random signs $\{r_i\}$ are only 4-wise independent.

   Remark: Do no repeat the proof seen in class, just explain how this change affects the analysis.

Extra credit:

4. (This question is an attempt to implement algorithm FM seen in class using pairwise independence, however the bound obtained below is too weak.)

   Let $X_1, \ldots, X_n$ be pairwise independent random variables, each distributed uniformly in $[0, 1]$. Fix $S \subset [n]$ and let $Y_S = \min\{X_i : i \in S\}$ be the minimum of the respective variables. Show that its expectation satisfies

   $$\Omega(\tfrac{1}{|S|}) \leq \mathbb{E}[Y_S] \leq O(\tfrac{1}{\sqrt{|S|}}),$$

   and provide an upper bound on its variance.