Exercise III, Sublinear Algorithms for Big Data Analysis 2024-2025

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students, and solve as many problems as you can. Problems marked (*) are more difficult, but also more rewarding. These problems have been taken from various sources on the Internet, too numerous to cite individually.

1 In class we saw a constant factor approximate randomized algorithm for the distinct elements problem which used pairwise independent hash functions. Design a $(1+\epsilon)$-approximate algorithm using the same techniques. How many buckets will you hash into, and how does this affect the space complexity of the algorithm?

Solution. Similarly to the sketch for the distinct element problem presented in class, we want to distinguish between the cases of less than t and greater than $(1+\epsilon)t$ elements. Select a hash function $h : [n] \to [d]$ for some to be defined integer d uniformly at random from a pairwise independent hash family. Maintain a counter c such that $c = \sum_{i:h(i)=1} x_i$. Output YES if $c > 0$ else output NO.

Case 1 $k \leq t$ **(NO case):** We have

$$\Pr[C > 0] \leq \sum_{i=1}^{k} \Pr[h(i) = 1] = \frac{k}{d} \leq \frac{t}{d}$$

Case 2 $k \geq (1+\epsilon)t$ **(YES case):** We want to lower bound the probability of $C > 0$ in this case. This probability is non-decreasing with k as if we have more elements, it is more likely that at least one of them hashes to 1. Hence $\Pr[C > 0]$ is smallest when $k = (1+\epsilon)t$ which is the case we will consider below. We have by the inclusion-exclusion principle

$$\begin{aligned}
\Pr[C > 0] &\geq \sum_{i=1}^{k} \Pr[h(i) = 1] - \sum_{i,j \in [k]} \Pr[h(i) = 1 \text{ and } h(j) = 1] \\
&= \frac{k}{d} - \frac{k^2}{d^2} \\
&= \frac{(1+\epsilon)t}{d} - \frac{(1+\epsilon)^2 t^2}{d^2}
\end{aligned}$$

Difference in the probability of saying YES in the two cases is

$$\begin{aligned}
\Pr[\text{YES in yes case}] - \Pr[\text{YES in no case}] &\geq \frac{(1+\epsilon)t}{d} - \frac{(1+\epsilon)^2 t^2}{d^2} - \frac{t}{d} \\
&= \frac{\epsilon t}{d} - \frac{(1+\epsilon)^2 t^2}{d^2} \\
&= \epsilon^2/5 - (\epsilon + \epsilon^2)^2/25 \text{ (assuming } d = \frac{5t}{\epsilon}) \\
&\geq \epsilon^2/5 - (2\epsilon)^2/25 \text{ (assuming } \epsilon \leq 1) \\
&= \epsilon^2/25
\end{aligned}$$

To get an algorithm with failure probability bounded by δ, it suffices to repeat the experiment $O(\frac{1}{\epsilon^3}\log(1/\delta))$ times and output YES if at least $\frac{t}{d}+\epsilon^2/50 = \epsilon/5+\epsilon^2/50$ fraction of the individual runs turn up YES, and say NO otherwise. We now show that $T = O(\frac{1}{\epsilon^3}\log(1/\delta))$ repetitions suffice to ensure that failure probability is at most δ. For each $t = 1, \ldots, T$ let $Y_t = 1$ if the t'th experiment says YES and 0 otherwise. Suppose that we are in the YES case, so that $\mathbb{E}[Y_t] \geq \epsilon/5 + \epsilon^2/25$ for each $t = 1, \ldots, T$. By Chernoff bounds we have for every $\delta \in [0, 1]$

$$\Pr\left[\sum_{t=1}^{T} Y_t \leq (1-\delta)\sum_{t=1}^{T}\mathbb{E}[Y_t]\right] \leq e^{-\delta^2 \sum_{t=1}^{T}\mathbb{E}[Y_t]/3}.$$

Since

$$\Pr\left[\sum_{t=1}^{T} Y_t \leq (\epsilon/5-\epsilon^2/50)T\right] \leq \Pr\left[\sum_{t=1}^{T} Y_t \leq \frac{\epsilon/5-\epsilon^2/50}{\epsilon/5-\epsilon^2/25}\cdot\sum_{t=1}^{T}\mathbb{E}[Y_t]\right],$$

we can apply the Chernoff bound above with $1-\delta = \frac{\epsilon/5-\epsilon^2/50}{\epsilon/5-\epsilon^2/25}$. This means that

$$\delta = 1 - \frac{\epsilon/5-\epsilon^2/50}{\epsilon/5-\epsilon^2/25} = \frac{(\epsilon/5-\epsilon^2/25)-(\epsilon/5-\epsilon^2/50)}{\epsilon/5-\epsilon^2/25} = \frac{\epsilon^2/50}{\epsilon/5-\epsilon^2/25} = \frac{\epsilon}{10-2\epsilon} \geq \frac{\epsilon}{10}$$

Substituting this into the Chernoff bound above yields

$$\Pr\left[\sum_{t=1}^{T} Y_t \leq (\epsilon/5-\epsilon^2/50)T\right] \leq e^{-\epsilon^2\cdot T\cdot\mathbb{E}[Y_1]/300} \leq e^{-\epsilon^3 T/1500},$$

since $\mathbb{E}[Y_i] \geq \epsilon/5$ by setting of parameters and the assumption that we are in the YES case. We thus get that setting $T = \frac{C}{\epsilon^3}\log(1/\delta)$ for a sufficiently large constant $C > 0$ suffices to ensure that the failure probability is upper bounded by δ. The NO case analysis is analogous (apply Chernoff bounds to $1 - Y_i$).

For each threshold t storing the hash function of each run, we need $O(\log n)$ bits. But we need to run $O(\frac{1}{\epsilon^3}\log(1/\delta))$ such experiments and so total storage space would be $O(\frac{1}{\epsilon^3}\log n)$. Finally, we consider all thresholds $t = (1+\epsilon)^j, j = 0, \ldots, \log_{1+\epsilon} n = O(\frac{1}{\epsilon}\log n)$, so the total space complexity for $(1+\epsilon)$-approximate distinct elements with failure probability $O((\delta/\epsilon)\log n)$ is $O(\frac{1}{\epsilon^4}\log^2 n)$ bits. □