CS787: Advanced Algorithms **Homework 5 Solutions**

Q1

We run two sketching subroutines while streaming the data. The first subroutine is the "Heavy Hitters" algorithm with hash table size $r = \lceil \frac{1}{\theta} \rceil - 1$ while the second subroutine is the "Count Sketch" algorithm using the trick of "Median of Means", taking the median of q means of p 2-way independent hash functions whose hash table has size k.

We recall from lectures that if there exists a heavy hitter in the stream it is guaranteed to be one of the outputs of the "Heavy Hitters" algorithm. Therefore, after running the above algorithm we obtain an estimate $\hat{f}_x$ for every element x that could potentially be a heavy hitter. We then check the value of these estimates output those elements x with $\hat{f}_x \geq \theta m$.

Now we prove that this procedure satisfies our requirement. Denote the true count of element x by f_x. Consider the ith computation of the mean of p independent hash functions whose hash table has size k. We note that there are q such computations, i.e., $1 \leq i \leq q$. Denote the estimate on element x by $\hat{f}_x^i$. According to the discussion in class, we know that $\mathbf{Var}\left[\hat{f}_x^i\right] \leq \frac{m^2}{kp}$. Set $kp = \frac{4}{\epsilon^2\theta^2}$ and apply Chebyshev's inequality, we know

$$\mathbf{Pr}\left[|\hat{f}_x^i - f_x| \geq \epsilon\theta m\right] \leq \frac{\mathbf{Var}\left[\hat{f}_x^i\right]}{\epsilon^2\theta^2 m^2} \leq \frac{1}{4}.$$

Recall $\hat{f}_x$ is the median of $\{\hat{f}_x^i\}_{i=1}^q$. Again, according to the discussion in class, we know

$$\mathbf{Pr}\left[|\hat{f}_x - f_x| \geq \epsilon\theta m\right] \leq \frac{1}{e^{O(q)}},$$

which is no more than $\frac{\delta}{r}$ if we set $q = \log \frac{r}{\delta}$.

Since we at most output r such values, the error probability of the whole algorithm is, by union bound, no more than δ. The space requirement is $r + kpq = O\left(\frac{1}{\theta^2}\frac{1}{\epsilon^2}\left(\log\frac{1}{\theta} + \log\frac{1}{\delta}\right)\right)$.

Q2

(a) Let $p(i,k)$ be the probability that after seeing the ith element, the value of X is k. Then we have

$$\sum_{k=0}^{i} p(i,k) = 1.$$

If the value of X is k after seeing the ith element, then at the $(i-1)$th step, the counter is either k and remains unchanged, or is $k-1$ and incremented. Thus for all i we have

$$p(i,k) = \frac{1}{2^{k-1}} p(i-1,k-1) + \left(1 - \frac{1}{2^k}\right) p(i-1,k),$$

i.e.

$$2^k p(i,k) = 2^k p(i-1,k) + 2p(i-1,k-1) - p(i-1,k).$$

Let E_m denote the expectation of Y after seeing m elements. Then we have

$$\begin{aligned}
E_m &= \sum_{k=0}^{m} 2^k p(m,k) \\
&= \sum_{k=0}^{m} 2^k p(m-1,k) + 2p(m-1,k-1) - p(m-1,k) \\
&= \sum_{k=0}^{m} 2^k p(m-1,k) + \sum_{k=0}^{m} 2p(m-1,k-1) - p(m-1,k) \\
&= E_{m-1} + 2\sum_{k=0}^{m-1} p(m-1,k) - \sum_{k=0}^{m} p(m-1,k) \\
&= E_{m-1} + \sum_{k=0}^{m-1} p(m-1,k) \\
&= E_{m-1} + 1
\end{aligned}$$

Since $E_1 = 2$, we have $\mathbf{E}[Y] = m+1$.

(b) Let S_m be the expectation of Y^2 after seeing m elements. Similar to part (a), for all i we have

$$2^{2k} p(i,k) = 2^{2k} p(i-1,k) + 2^{k+1} p(i-1,k-1) - 2^k p(i-1,k).$$

Therefore, we have

$$\begin{aligned}
S_m &= \sum_{k=0}^{m} 2^{2k} p(m,k) \\
&= \sum_{k=0}^{m} 2^{2k} p(m-1,k) + 2^{k+1} p(m-1,k-1) - 2^k p(m-1,k) \\
&= S_{m-1} + 3\sum_{k=0}^{m-1} 2^{k-1} p(m-1,k-1) \\
&= S_{m-1} + 3E_{m-1} \\
&= S_{m-1} + 3m.
\end{aligned}$$

Since $S_1 = 4$, we have $S_m = \frac{3m^2+3m+2}{2}$ and thus

$$\mathbf{Var}[Y] = S_m - E_m^2 = \frac{m(m-1)}{2}.$$

(c) By part (a), $Y - 1$ is an unbiased estimator of m. To get an (ϵ, δ)-estimator within small space, we use the "median of the mean" trick discussed in class.

We first run p such estimations and return the mean $\overline{Y}$. Using Chebyshev's inequality, we know

$$\mathbf{Pr}\left[\overline{Y} \notin (1 \pm \epsilon)m\right] \leq \frac{1}{2p\epsilon^2}.$$

In order for the above bound to be no more than $\frac{1}{4}$, we need $p = \Theta(\frac{1}{\epsilon^2})$.

We then run q such improved estimations and return the median Y'. By applying the bound we learned in class, we know that Y' is an (ϵ, δ)-estimator when $q = \Theta(\log \frac{1}{\delta})$.

For each counter, we know that $\mathbf{E}[X] = \mathbf{E}[\log Y] \leq \log \mathbf{E}[Y] = \log m$ and we only need to keep $\log X$ bits for counting purposes. Therefore, the total space requirement is $O\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta} \log \log m\right)$ in expectation.

Q3

(a) Since H is a subgraph of G, $d_G(u, v) \leq d_H(u, v)$ holds for any u, v. Now consider the other part with two vertices u and v. We can assume that u and v are connected in G (and H), otherwise the inequality trivially holds.

Suppose $P = (x_0, x_1, x_2, \cdots, x_{d_G(u,v)})$ is a shortest path of length $d_G(u, v)$ between u and v where $x_0 = u, x_{d_G(u,v)} = v$. Then for each edge between x_i and x_{i+1}, we have $d_H(x_i, x_{i+1}) \leq t$. Therefore, we have

$$d_H(u, v) \leq \sum_{i=0}^{d_G(u,v)-1} d_H(x_i, x_{i+1}) \leq t d_G(u, v).$$

(b) Consider an arbitrary cycle in H. Denote its length by k. Let $\{u, v\}$ be the last edge added by the algorithm in this cycle, then we know that before adding it we have $t < d_H(u, v) < k - 1$, i.e., $k \geq t + 2$.

(c) We first show the regular case where each vertex has degree $d = \frac{2m}{n}$. Consider an arbitrary vertex v and produce a BFS tree rooted at v. Then in this BFS tree there is no edge between any two vertex at the first $\frac{g}{2}$ levels, since otherwise a cycle of length at most g can be found. Therefore, all vertices in the first $\frac{g}{2}$ levels are distinct. Counting the number of nodes in the BFS tree in the first $\frac{g}{2}$ levels, we have

$$1 + d + d(d-1) + \cdots + d(d-1)^{g/2} \leq n,$$

which gives us

$$m \leq \frac{1}{2} n(n+1)^{g/2} + n = n^{1+O(1/g)}.$$

In the general case, we iteratively delete vertices v (and their incident edges) with degree $d(v) < \frac{m}{n} =: d'$ from G. This way we delete less than $n \cdot \frac{m}{n} = m$ edges in total, so we are left with a nonempty graph where each vertex has degree at least d'. Note that such deletions does not increase the girth of the graph. Denote the number of vertices in the resulting graph by $n' \leq n$. Using an argument similar to the regular case discussed above, we have

$$1 + d' + d'(d'-1) + \cdots + d'(d'-1)^{g/2} \leq n',$$

which gives us

$$m \leq \frac{1}{2} n'(n'+1)^{g/2} + n' \leq n^{1+O(1/g)}.$$

Q4

(a) We give an upper bound on the probability. Since the graph has minimum cut of size $|C^*|$, we know the degree $d(v) \geq |C^*|$ for each vertex $v \in V$. Therefore, at iteration i, the graph has $n-i$ vertices remaining, and the total number of edges remaining is at least $(n-i)d(v)/2 \geq (n-i)|C^*|/2$. Given that none of the $\alpha|C^*|$ edges in C has been chosen up to iteration i, the probability that the algorithm picks some edge in C is at most $\frac{\alpha|C^*|}{(n-i)|C^*|/2} = \frac{2\alpha}{n-i}$.

(b) Let E_i be the even that the cut C survived iteration i. The algorithm stops at the iteration $n-2\alpha$ when there are 2α vertices remaining and uniformly randomly picks a cut (out of the $2^{2\alpha}-1$ remaining possible cuts) in the remaining graph. Therefore, the probability that C is picked by the algorithm is at least $\frac{1}{2^{2\alpha}-1}\mathbf{Pr}[E_{n-2\alpha}]$ where

$$\mathbf{Pr}[E_{n-2\alpha}] = E_1 \prod_{i=1}^{n-2\alpha-1} \mathbf{Pr}[E_{i+1}|E_i] \geq \prod_{i=0}^{n-2\alpha-1} \left(1 - \frac{2\alpha}{n-i}\right) = \binom{n}{2\alpha}^{-1}.$$

(c) Suppose the total number of cuts of size smaller than $\alpha|C^*|$ by s. From the calculation in part (b), we know that the probability that any of the s small cuts is output is at least $\frac{1}{2^{2\alpha}-1}\binom{n}{2\alpha}^{-1}$. On the other hand, the total probability is of course no more than 1. Therefore, we have

$$s \leq \left(2^{2\alpha}-1\right)\binom{n}{2\alpha}.$$

Q5

We independently color each vertex with one of the k colors in $S(v)$. Let $E_{\{u,v\},c}$ be the "bad" event that both sides of the edge $\{u,v\}$ are colored $c \in S(u) \cap S(v)$. Then we know

$$\mathbf{Pr}\left[E_{\{u,v\},c}\right] = \frac{1}{k^2}.$$

Since we color the vertices independently, the bad event $E_{\{u,v\},c}$ is independent from any other bad event $E_{\{u',v'\},c}$ if $\{u,v\}$ and $\{u',v'\}$ share no endpoints. Therefore, each bad event $E_{\{u,v\},c}$ is dependent on at most $2k \cdot \frac{k}{10} = \frac{k^2}{5}$ other bad events.

Applying the LLL, we know that the probability of avoiding all bad events is nonzero since $epd < 1$ where $p = \frac{1}{k^2}$ and $d = \frac{k^2}{5}$.