CSE 632: Analysis of Algorithms II **Fall 2017**

Lecture 24 (11/17/2017): Sketching and Streaming (II)

Lecturer: Shi Li *Scribe: Xiangyu Guo*

24.1 Computing F_2

Problem description: Given data stream $\langle i_1, \ldots, i_n \rangle$, where $i_t \in [m] \, \forall \, t \in [n]$; and let $\mathcal{X} = (x_1, x_2, \ldots, x_m)$ s.t. x_i denote the number of times that i appears in the stream. The goal is to compute F_2 — the ℓ_2-norm of $\mathcal{X}$, i.e., $f = \|\mathcal{X}\|_2^2 = \sum_{i \in [m]} x_i^2$, using only $\mathrm{poly}(1/\epsilon, 1/\delta, \log n, \log m)$ storage.

Analysis: We choose a function h uniformly at random from $\mathcal{H} = \{h : [m] \mapsto \{-1, +1\}\}$, where $\mathcal{H}$ is the set of all function mapping from $[m]$ to $\{-1, +1\}$. And our estimator for f is as follows:

Algorithm 1 Estimator for f

```
1: Z ← 0;
2: for every i_t comes do
3:     Z ← Z + h(i_t);
4: Output f̂ = Z²;
```

The first observation is:

$$Z = \sum_{i=1}^{m} x_i h(i)$$

And its expectation (w.r.t. the choice of h) is

$$\mathbb{E}[Z^2] = \mathbb{E}[\sum_{i=1}^{m} \sum_{j=1}^{m} x_i x_j h(i) h(j)] = \sum_{i=1}^{m} \sum_{j=1}^{m} x_i x_j \mathbb{E}[h(i)h(j)] = \sum_{i=1}^{m} x_i^2 = f$$

i.e., Z^2 is an unbiased estimator for f. Therefore, if Z^2 has a small variance, we can give an accurate estimate for F_2. By definition, the variance of Z^2 is $\mathrm{Var}[Z^2] = \mathbb{E}[Z^4] - \mathbb{E}^2[Z^2]$, where

$$\begin{aligned}
\mathbb{E}[Z^4] &= \mathbb{E}\left[\sum_{i,j,k,l \in [m]} x_i x_j x_k x_l h(i)h(j)h(k)h(l)\right] \\
&= \sum_{i,j,k,l \in [m]} x_i x_j x_k x_l \mathbb{E}\left[h(i)h(j)h(k)h(l)\right] \\
&= \sum_{i \in [m]} x_i^4 + \binom{4}{2} \sum_{i,j \in [m]} x_i^2 x_j^2 \qquad (24.1)
\end{aligned}$$

The third equality is because $\mathbb{E}\left[h(i)h(j)h(k)h(l)\right]$ will be zero except for the case $\mathbb{E}\left[h(i)^4\right]$ or $\mathbb{E}\left[h(i)^2h(j)^2\right]$. Thus,

$$\begin{aligned}\mathrm{Var}[Z^2] &= \sum_{i\in[m]} x_i^4 + 6\sum_{i,j\in[m]} x_i^2x_j^2 - (\sum_{i\in[m]} x_i^2)^2 \\ &= 4\sum_{i,j\in[m]} x_i^2x_j^2 \\ &\le 2(\sum_{i\in[m]} x_i^2)^2 = 2f^2\end{aligned}$$

To clear notations, let $Y = Z^2$, then $\mathbb{E}[Y] = f, \mathrm{Var}[Y] \le 2f^2$. Now similar to the last lecture, we call each run of Algorithm 1 as a "single experiment", and run $k = \lceil \frac{2}{\delta\epsilon^2} \rceil$ experiments independently. Denote the output of i-th run as Y_i, and define $Y_0 = \frac{1}{k}(Y_1 + Y_2 + \cdots + Y_k)$, then we have

$$\mathbb{E}[Y_0] = f, \quad \mathrm{Var}[Y_0] \le \frac{2f^2}{k}$$

By Chebyshev's inequality,

$$\Pr[|Y_0 - \mathbb{E}[Y_0]| \ge \epsilon f] \le \frac{\mathrm{Var}[Y_0]}{\epsilon^2 f^2} = \frac{2}{k\epsilon^2} \le \delta$$

So we get a (ϵ, δ)-approximation with $O\left(\frac{1}{\delta\epsilon^2}\right)$ memory cost. Actually the $O(1/\delta)$ factor can be further reduced by then so-called "median trick": replace each single experiment with the mean output of $O(1/\epsilon^2)$ experiments, and make $k = O(\log(1/\delta))$ such means (i.e., run $O\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$ experiments in total). Then we output the median of these $O(1/\epsilon^2)$ means as the final result. Specifically, let $\bar{Y}^t$ be the mean of the t-th $6/\epsilon^2$ experiment repetitions, and consider the random variables $W^t = \mathbb{I}\{|\bar{Y}^t - f| > \epsilon f\}(t = 1, \ldots, k)$: by Chebyshev's inequality, each W^t is a Bernoulli random variable with $\Pr[W^t = 1] \le 1/3$. So if the median of all $\bar{Y}^t$s is at least ϵf far away from f, then at least half of all W^ts are 1, of which the probability, by a standard Chernoff bound, decreases exponentially in k.

There's only one problem left: how do we sample the random function h? The function set $\mathcal{H}$ is of size 2^m, sampling uniformly random from $\mathcal{H}$ will require $O(m)$ random bits, which is unacceptable. But from the derivation of variance (24.1), we can see that h only needs to be 4-wise independent:

Definition 24.1 (4-wise independent function) *Let $\mathcal{H}$ be a family of functions from A to B, $\mathcal{H}$ is 4-wise independent if $\forall$ 4 distinct elements $a, b, c, d \in A$, and 4 values $v_a, v_b, v_c, v_d \in B$, we have*

$$\begin{aligned}&\Pr_{h\in\mathcal{H}}[h(a) = v_a, h(b) = v_b, h(c) = v_c, h(d) = v_d] = \\ &\qquad \Pr_{h\in\mathcal{H}}[h(a) = v_a] \Pr_{h\in\mathcal{H}}[h(b) = v_b] \Pr_{h\in\mathcal{H}}[h(c) = v_c] \Pr_{h\in\mathcal{H}}[h(d) = v_d]\end{aligned}$$

Example 24.2 (4-wise independent function) *Let q be a prime, and define $g_{a,b,c,d} : [m] \mapsto \mathbb{F}_{2^q}$ as follows:*

$$g_{a,b,c,d}(x) = ax^3 + bx^2 + cx + d$$

here $a, b, c, d \in \mathbb{F}_{2^q}$. Now let $h_{a,b,c,d}(x) = 2(g_{a,b,c,d}(x) \mod 2) - 1$, then $\mathcal{H} = \{h_{a,b,c,d} : a, b, c, d \in \mathbb{F}_{2^q}\}$ is a 4-wise indenpend function family.

24.2 Summary

From these two lectures we conclude that F_0 and F_2 can be estimated with $\mathrm{poly}(1/\epsilon, \log(1/\delta), \log n, \log m)$ memory space. And the method we give can even work with stream in more general forms: $\langle (i_1, \Delta_1), (i_2, \Delta_2), \ldots, (i_t, \Delta_t) \rangle$, such that $x_i = \sum_{t: i_t = i} \Delta_t$. Actually for any F_p with $p \in [0, 2]$, we can obtain a (ϵ, δ)-approximation with $\mathrm{poly}(1/\epsilon, \log(1/\delta), \log n, \log m)$ space. But for F_p where $p > 2$, we always need $n^{\Omega(1)}$ space.