CS369G: Algorithmic Techniques for Big Data Spring 2015-2016

Lecture 3: Estimating F_k norms for $k \geq 2$

Prof. Moses Charikar *Scribes: Spencer Yee, Michael Xie*

1 Overview

In this lecture, we discuss frequency moment estimators, how to implement efficient hash functions for the AMS second frequency moment estimator. linear sketches, the Johnson-Lindenstrauss lemma, and computing frequency moments for $k > 2$.

2 Second Frequency Moments

In the previous lecture, we saw an elegant estimator for the second frequency moment for a stream. We will review the details of this estimator. We require a hash function $h : [n] \to \{\pm 1\}$ which we assume to be random, and we define the frequency f_i as the number of copies of element i in a stream, so the second frequency moment F_2 is Σf_i^2. Start with a counter Y initialized to 0, and add $h(i)$ to Y for each element i in the stream. In the end, $Y = \Sigma f_i h(i)$, and Y^2 is our estimator for F_2.

For now, think of $h(i)$ as a random variable x_i. Then $Y = \Sigma f_i x_i$, and as shown in the previous lecture, $E[Y^2] = F_2$ and $\text{Var}[Y^2] \leq 2F_2^2$, the desired properties for our estimator. The key observations that yielded those results were $E[x_i^2] = 1$, $E[x_i] = 0$, $E[x_i x_j] = 0$, and what the value of $E[x_{i_1} x_{i_2} x_{i_3} x_{i_4}]$ is, depending on the number of distinct x_i. Ignoring the problem of how to implement h, Y requires very little space, has the right expectation, and a small variance, so we can apply median of means. This means that in $O\left(\frac{1}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right)$ space, we can estimate F_2 within ϵ error with probability at least $1 - \delta$.

Now we address the issue of how to implement the hash function. We've assumed that the hash function is random, but in order to implement a random hash function, we essentially need to store the hash value for every element in the set, which is n bits. So we can't use a completely random hash function. Looking back at our key observations, though, the properties of randomness that we require can be satisfied with a 4-wise independent hash function, where

$$\Pr[h(x_1) = a_1, h(x_2) = a_2, h(x_3) = a_3, h(x_4) = a_4] = \prod_{i=1}^{4} \Pr[h(x_i) = a_i].$$

There are 4-wise independent hash functions that only require $O(\log n)$ space and $O(\log n)$ evaluation time, so we can indeed estimate F_2 with a small amount of space.

2.1 Linear Sketches

This is an example of a linear sketch. We can think of the input stream as a vector $x \in \mathbb{R}^n$ where the i-th element of x is f_i, where each element in the stream updates x. Then we can represent that d copies of Y that we are storing as Ax, where $A \in \mathbb{R}^{d \times n}$. Each row of A represents a different 4-wise independent hash function, and the i-th entry in a row representing hash function h is simply $h(i)$. We don't actually store the entire matrix A; instead, we have an implicit representation that allows us to compete each coordinate when needed. Every time an element in the stream appears, we update x with Δx, a vector with 1 in one coordinate and 0s in the others. The sketch is incremented by $A\Delta x$, which is equivalent to updating our copies of Y with the appropriate hash values of the new element. The fact that this is a linear sketch is useful, since we can use operations such as addition and subtraction. Combining linear sketches is very simple, since $Ax + Ay = A(x + y)$.

We can also think of the input as being updates to coordinates rather than elements. In the turnstile model, the input is x, and updates are changes to one coordinate of x, which need not be $+1$. Linear sketches support this. We can also have negative updates, which are also supposrted by linear sketches.

3 Johnson-Lindenstrauss Lemma

The fact that we can estimate the second frequency moment very well with a small amount of space is perhaps less surprising given the Johnson-Lindenstrauss Lemma. The statement is as follows.

Lemma 1. *(Johnson-Lindenstrauss) Let $G \subset (\mathbb{R}^d, l_2)$ be a set of n points. Then for any $0 < \epsilon < \frac{1}{2}$ and $k = O(\log(n)/\epsilon^2)$, there exists a mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $v_i, v_j \in G$,*

$$(1 - \epsilon)\|v_i - v_j\|_2 \leq \|f(v_i) - f(v_j)\|_2 \leq (1 + \epsilon)\|v_i - v_j\|_2$$

For the l_2 norm, it can be seen that a set of n points can be mapped to a $(n-1)$-dimensional space without distance distortion. By Johnson-Lindenstrauss, we can reduce this to $O(\log(n)/\epsilon^2)$ with small distortion. The existence of such a mapping can be proven using a linear mapping in the form

$$f(v) = \frac{Mv}{\sqrt{k}}$$

where $M \in \mathbb{R}^{k \times d}$ and $M_{ij} = \mathcal{N}(0, 1)$, a matrix with elements that are drawn i.i.d. from the unit normal distribution. Note that M is data-oblivious; we do not use information from the n points to define its elements.

We use the following tail bound on chi-squared distributions to show the JL lemma.

Lemma 2. *Let $Z_1, \ldots, Z_k$ be i.i.d. unit normal random variables. Let $Y = \sum_i Z_i^2$. Then*

$$\Pr\left[(1 - \epsilon)^2 k \leq Y \leq (1 + \epsilon)^2 k\right] \geq 1 - 2e^{-c\epsilon^2 k}$$

for some suitable constant c.

Consider the case where the input vector $v \in \mathbb{R}^d$ is a unit vector. Then for row i of M, we have

$$(Mv)_i = M_i^T v = \sum_j^d M_{ij} v_j = \left(\sum_j^d v_j \right)^{\frac{1}{2}} Y = Y$$

where Y is unit normal. Therefore in the case where the input is a unit vector, the output coordinates are distributed as i.i.d unit Gaussians. Then we can conclude that

$$\Pr\left[(1-\epsilon) \leq \left\| \frac{Mv}{\sqrt{k}} \right\|_2 \leq (1+\epsilon) \right] \geq 1 - 2e^{-c\epsilon^2 k}$$

or equivalently,

$$\Pr\left[(1-\epsilon)^2 k \leq \|Mv\|_2^2 \leq (1+\epsilon)^2 k \right] \geq 1 - 2e^{-c\epsilon^2 k}$$

since $\|Mv\|_2^2 = \sum_i^k (Mv)_i^2$ is a sum of squared unit normal distributions, on which we can invoke Lemma 2.

For general inputs v, we can write $\tilde{v} = v/\|v\|_2$ so that, using the result above,

$$\Pr\left[(1-\epsilon) \leq \left\| \frac{M\tilde{v}}{\sqrt{k}} \right\|_2 \leq (1+\epsilon) \right] \geq 1 - 2e^{-c\epsilon^2 k}$$

Then it follows that

$$\Pr\left[(1-\epsilon)\|v\|_2 \leq \left\| \frac{Mv}{\sqrt{k}} \right\|_2 \leq (1+\epsilon)\|v\|_2 \right] \geq 1 - 2e^{-c\epsilon^2 k}.$$

Taking the inputs v to be the $\binom{n}{2}$ differences between pairs of points in the original space, we use the union bound to show that we can make the failure probability small. The failure probability for a particular difference $v_{ij} = v_i - v_j$ is at most $2e^{-c\epsilon^2 k}$. Choosing $k = (c' \log(n))/\epsilon^2$ for some suitable constant c', we have that the failure probability is at most $2e^{-cc' \log n}$. Choosing sufficiently large constants so that this failure probability is less than $(1/n^3)$, by union bound over the $O(n^2)$ pairs, the failure probability of the random scheme for finding the linear mapping M is bounded by $(1/n)$.

In terms of improvements to this basic scheme, the fast JL transform constructs M with structure such that fast multiplication ($O(d + \log n)$) can be done. Another note is that the F_2 estimator in some sense also preserves the l_2 distances between data streams using a linear mapping; using the Johnson-Lindenstrauss argument with the relaxation that each pair of distances have low distortion with high probability, we can find that we can reduce to a dimension $k = O\left(\frac{1}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right)$, similar to the space requirement in the F_2 estimator. For dimension reduction schemes where the norm in the original and reduced space is the same, it is not possible to prove results similar to Johnson-Lindenstrauss with l_∞ and l_1.

4 Frequency Moments for $k > 2$

For $k > 2$, we need $\Omega\left(n^{1-\frac{2}{k}}\right)$ space to compute an estimator for F_k with the required properties. We will discuss a suboptimal construction which uses $\tilde{O}\left(n^{1-\frac{1}{k}}\right)$ space.

Suppose we have a stream of length m, and we choose a random element a_t (the t-th element in the stream). We want to compute:

$$R_t = \text{ the number of elements } a_j \text{ such that } a_j = a_t \text{ and } t \leq j \leq m.$$

In other words, we want to know how many symbols in the stream, starting from the chosen symbol, have the same symbol?

Reservoir Sampling We can randomly select an element from the stream in pass using *reservoir sampling*. In this scheme, we keep the first element in memory, and when the n-th element arrives, we evict our stored element and store the new item with probability $\frac{1}{n}$. We can also compute R during this pass by keeping a counter which we increment upon seeing matching symbols and reset to 1 when we replace our stored element (even if we replace the stored element with an identical symbol). We will use R to compute an estimator for F_K.

Lemma 3. *For any function $g : \mathbb{R} \to \mathbb{R}$ such that $g(0) = 0$, let R as constructed as above an unbiased estimator for $F_g = \sum_{i=1}^{n} g(f_i)$ is given by $Y = m(g(R) - g(R-1))$.*

Proof. Let i be an element in $[n]$. The probability of sampling a copy of i is $\frac{f_i}{m}$. If we select the r-th copy of i, then $R = f_i - r + 1$. Each copy is equally likely to be picked, so

$$\begin{aligned}
E[Y] &= \sum_{i \in [n]} \frac{f_i}{m}(m)(g(R) - g(R-1)) \\
&= \sum_{i \in [n]} f_i \left(\sum_{r=1}^{f_i} \frac{1}{f_i}(g(f_i - r + 1) - g(f_i - r)) \right) \\
&= \sum_{i \in [n]} ((g(f_i) - g(f_i - 1)) + (g(f_i - 1) - g(f_i - 2)) + \cdots + (g(1) - g(0))) \\
&= \sum_{i \in [n]} g(f_i).
\end{aligned}$$

□

Thus, if we use $g = x^k$, $E[Y] = F_k$. We will show in the next lecture that $\text{Var}[Y] \leq k n^{1-\frac{1}{k}} (F_k)^2$, proving the space constraint for our estimator for F_k.