1 An equivalent view of estimating F_2

Again, you have a data stream of elements $\sigma_1, \sigma_2, \ldots$, each element σ_j drawn from the universe $[D]$. This stream defines a frequency vector $\vec{x} \in \mathbb{R}^D$, where x_i is the number of times element i is seen. Consider the following algorithm to computing $F_2 := \|x\|_2^2 = \sum_i x_i^2$.

> Take a (suitably random) hash function $h : [D] \to \{-1, +1\}$. Maintain counter C, which starts off at zero. Every time an element i comes in, increment the counter $C \to C + h(i)$. And when queried, we reply with the value C^2.

Hence, having seen the stream that results in the frequency vector $x \in \mathbb{Z}^D_{\geq 0}$, the counter will have the value $C = \sum_i x_i\, h(i)$. Does $E[C^2]$ at least have the right expectation? It does:

$$E[C^2] = E[\sum_{i,j} h(i)h(j)x_i x_j] = \sum_i x_i^2.$$

And what about the variance? Recall that $\mathrm{Var}(C^2) = E[(C^2)^2] - E[C^2]^2$, so let us calculate

$$E[(C^2)^2] = E[\sum_{p,q,r,s} h(p)h(q)h(r)h(s)x_p x_q x_r x_s] = \sum_p x_p^4 + 6\sum_{p<q} x_p^2 x_q^2.$$

So

$$\mathrm{Var}(C^2) = \sum_p x_p^4 + 6\sum_{p<q} x_p^2 x_q^2 - (\sum_p x_p^2)^2 = 4\sum_{p<q} x_p^2 x_q^2 \leq 2E[C^2]^2.$$

What does Chebyshev say then?

$$\Pr[|C^2 - E[C^2]| > \epsilon E[C^2]] \leq \frac{\mathrm{Var}(C^2)}{(\epsilon E[C^2])^2} \leq \frac{2}{\epsilon^2}.$$

Not that hot: in fact, this is usually more than 1.

But if we take a collection of k such independent counters $C_1, C_2, \ldots, C_k$, and given a query, take their average $\overline{C} = \frac{1}{k}\sum_i C_i$, and return $\overline{C}^2$. The expectation of the average remains the same, but the variance falls by a factor of k. And we get

$$\Pr[|\overline{C}^2 - E[\overline{C}^2]| > \epsilon E[\overline{C}^2]] \leq \frac{\mathrm{Var}(\overline{C}^2)}{(\epsilon E[\overline{C}^2])^2} \leq \frac{2}{k\epsilon^2}.$$

So, our probability of error on any query is at most δ if we take $k = \frac{2}{\epsilon^2\delta}$.

1.1 Hey, those calculations look familiar...

Sure. This is just a restatement of what we did in lecture. There we took a matrix M and filled with random $\{-1, +1\}$ values—hence each row of M corresponds to a hash function from $[D]$ to $\{-1, +1\}$. And taking k rows in the matrix corresponds to the variance reduction step at the end.

1.2 Limited Independence

How much randomness do you need for the hash functions? Indeed, hash functions which are 4-wise independent suffice for the above proofs to go through. And how does one get a 4-wise independent hash function? Watch this blog (and the HWs) for details.