CS5120: Probability & Computing **Jan-May 2020**

Lecture 16
The Count-Min Sketch

Lecturer: N.R.Aravind *Scribe: N.R.Aravind*

In the previous class, we saw the Misra-Gries algorithm for finding heavy hitters (elements with frequency at least εn in a stream of length n). It used $O(\frac{1}{\varepsilon}\log m)$ space where the domain of the elements is $\{1, 2, \ldots, m\}$.

We now see the Count-Min algorithm, which is due to Cormode and Muthukrishnan (2003). It gives an estimate of the frequencies of every element, and makes an error of at most εn on each estimate.

1. Choose t hash functions $h_1, \ldots, h_t : \{1, 2, \ldots, m\} \to \{1, 2, \ldots, k\}$, each from a pairwise independent hash family.
2. For $1 \le i \le t$ and $1 \le j \le k$, initialize counters $C[i, j]$ to zero. The counters $C[i, 1], \ldots, C[i, k]$ correspond to the ith hash function.
3. For each element x, do: set $i_1 = h_1(x), \ldots, i_t = h_t(x)$. Increment each of $C[1, i_1], C[2, i_2], \ldots, C[t, i_t]$ by one.
4. For each $y \in \{1, \ldots, m\}$, output $min\{C[1, h_1(x)], C[2, h_2(x)], \ldots, C[t, h_t(x)]\}$ as the estimated frequency of y.

Analysis: We will choose the values of t and k, based on the analysis.

Fix $y \in \{1, \ldots, m\}$ and $i \in \{1, 2, \ldots, t\}$. We have $C[i, y] \ge f(y)$ and $E[C[i, h(y)]] = f(y) + \frac{n - f(y)}{k} \le f(y) + \frac{n}{k}$. Thus, by applying Markov's inequality, we get:

$$Pr(C[i, y] - f(y) \ge \varepsilon n) \le \frac{1}{\varepsilon k}.$$

We now choose $k = \lceil \frac{2}{\varepsilon} \rceil$ so that the above probability is at most $1/2$. We choose $t = \lceil (1 + c) \log m \rceil$ so that we get:

$$Pr(Min(C[1, h_1(y)], \ldots, C[t, h_t(y)]) \ge f(y) + \varepsilon n) \le \frac{1}{2^t} \le \frac{1}{m^{1+c}}.$$

Finally, by applying the union bound on all m elements, we can bound the total probability of error by at most $\frac{1}{m^c}$. The choice of $c = \frac{1}{\log m} \log\left(\frac{1}{\delta}\right)$ will bring the error to at most δ.

Thus, the total space complexity is $O\left(\frac{\log n}{\varepsilon}(\log m + \log\left(\frac{1}{\delta}\right))\right)$, where $\log n$ is for the space per counter and the remaining factors are the total number of counters.