CS 683: Advanced Design & Analysis of Algorithms April 18, 2008

Lecture 35: Low-memory approximations

Lecturer: John Hopcroft *Scribes: Jean-Baptiste Jeannin, Chris Provan*

1 Counting the number of occurrences of an element in a sequence

1.1 One element

Given a sequence of 0's and 1's of length n, we would like to count the number of 1's in this sequence. Doing this in the obvious way (having a counter and just counting them) is done in space $\log n$. We would like to come up with something better.

Instead we keep a k such that if m is the number of 1's then $m \simeq 2^k$. k is initialized to 0, and while going through the sequence, each time we see a 1, we increase k by 1 with probability $\frac{1}{2^k}$. This should give a good approximation of m, but in space $\log \log n$.

Now we would like to ask what the error is between m and 2^k: is 2^k in the range $[m-1; m+1]$, in the range $[\frac{m}{2}; 2m]$ or in the range $[\sqrt{m}; n^2]$? What if we instead keep k such that $2^{2^k} \simeq n$? Now just $\log \log \log n$ space, but how does the error change?

Question: How much space is actually needed to flip a coin? Do we need $\log n$ space already?

1.2 Several elements

But what happens if you want the frequency of all elements? Let us suppose an alphabet size of m. Define the alphabet as $\{s_1, ..., s_m\}$. Let us introduce a randomly chosen hash function $h : \{s_1, ..., s_m\} \to \{-1; 1\}$. Let us apply h on all elements of the string and then add everything up; the number c we get, if n_i is the number of occurrences of each element is:

$$c = \sum_{i=1}^{m} n_i h(s_i),$$

and

$$E[ch(s_i)] = n_i$$

for all i, although the variance may be large.

More generally, if we have a very large alphabet, for example an alphabet $\{c_1, ..., c_{10^6}\}$, and we want to compress our counts to 10 dimensions, we can declare a randomly chosen hash function $h : \{c_1, ..., c_{10^6}\} \to \{d_1, ..., d_{10}\}$. We start with the counts for all the

d_i equal to 0, and each time we get a c_i in the sequence, we add 1 to the count for $h(c_i)$. Moreover, since $\mathbb{N}^{10}$ is isomorphic to $\mathbb{N}$, the $\{d_1, ..., d_{10}\}$ can then be reduced to a single number in $\mathbb{N}$.

2 Number of distinct elements in a data stream

Now, still with an alphabet $\{s_1, .., s_m\}$, we would like to count the number of distinct elements in a data stream and in low space. More precisely, we would like to know if this number of distinct elements is greater than t or not. To do that, we declare a randomly chosen hash function $h : \{s_1, ..., s_m\} \to \{1, ..., t\}$, and we answer yes if any symbol in the data stream was mapped to 1, i.e., if $\exists i, h(s_i) = 1$.

An alternate algorithm declares a hash function $h : \{s_1, ..., s_m\} \to \{1, ..., m\}$ chosen randomly from a 2-universal set of hash functions. Define d to be the number of distinct symbols appearing in the data stream. Then d is approximated by $\frac{m}{\min_k h(a_k)} - 1$, where $\{a_1, a_2, \ldots, a_d\}$ are the distinct elements appearing in the stream. The following lemma gives a probabilistic bound on this approximation.

Lemma 1. $P\left[\frac{d}{6} \leq \frac{m}{\min_k h(a_k)} \leq 6d\right] \geq \frac{2}{3}$.

Proof. Let $\{a_1, a_2, \ldots, a_d\}$ be the symbols appearing in the stream. For each $k \leq d$, define the indicator function

$$z_k = \begin{cases} 1 & \text{if } h(a_k) < \frac{m}{6d}; \\ 0 & \text{otherwise,} \end{cases}$$

and let $z = \sum_{k=1}^{d} z_k$. h is chosen uniformly at random from a 2-universal set of hash functions, so $h(a_k)$ is uniformly distributed over Σ. Then

$$E[z_k] = P[z_k = 1] = P\left[h(a_k) < \frac{m}{6d}\right] = \frac{1}{6d}, \text{ and}$$

$$E[z] = \sum_{k=1}^{d} E[z_k] = d \cdot \frac{1}{6d} = \frac{1}{6}.$$

Therefore we have

$$\begin{aligned} P\left[\frac{m}{\min_k h(a_k)} > 6d\right] = P\left[\min_k h(a_k) < \frac{m}{6d}\right] &= P\left[\exists k : h(a_k) < \frac{m}{6d}\right] \\ &= P\left[\exists k : z_k = 1\right] \\ &= P\left[z \geq 1\right] \\ &= P\left[z \geq 6E[z]\right] \\ &\leq \frac{1}{6}, \end{aligned}$$

where the last inequality is a direct application of Markov's inequality.

Now define the indicator function

$$y_k = \begin{cases} 0 & \text{if } h(a_k) > \frac{6m}{d}; \\ 1 & \text{otherwise,} \end{cases}$$

for each $k \leq d$, and define $y = \sum_{k=1}^{d} y_k$. Then for each k,

$$E[y_k] = P[y_k = 1] = P\left[h(a_k) \leq \frac{6m}{d}\right] = \frac{6}{d}, \text{ and}$$

$$E[y] = \sum_{k=1}^{d} E[y_k] = d \cdot \frac{6}{d} = 6.$$

We will also need a bound on the variance of y. For each k,

$$\begin{aligned} Var(y_k) &= P[y_k = 1] \cdot (1 - E[y_k])^2 + P[y_k = 0] \cdot (E[y_k])^2 \\ &= \frac{6}{d} \cdot \left(1 - \frac{6}{d}\right)^2 + \left(1 - \frac{6}{d}\right) \cdot \left(\frac{6}{d}\right)^2 \\ &= \frac{6}{d} \cdot \left(1 - \frac{6}{d}\right) \\ &< \frac{6}{d}. \end{aligned}$$

By the properties of 2-universal sets, the $h(a_k)$ are pairwise independent for all $k \neq j$, and thus $Cov(y_k, y_j) = 0$. Therefore

$$Var(y) = \sum_{k=1}^{d} Var(y_k) < d \cdot \frac{6}{d} = 6.$$

Then we have

$$\begin{aligned} P\left[\frac{m}{\min_k h(a_k)} < \frac{d}{6}\right] = P\left[\min_k h(a_k) > \frac{6m}{d}\right] &= P\left[\forall k : h(a_k) > \frac{6m}{d}\right] \\ &= P\left[\forall k : y_k = 0\right] \\ &= P\left[y = 0\right] \\ &\leq P\left[|y - 6| \geq 6\right] \\ &= P\left[|y - E[y]| \geq 6\right] \\ &\leq \frac{Var(y)}{36} \\ &< \frac{1}{6}, \end{aligned}$$

where the second to last inequality is a direct application of Chebyshev's inequality.

Combining the two intermediate results above gives

$$\begin{aligned} P\left[\frac{d}{6} \leq \frac{m}{\min_k h(a_k)} \leq 6d\right] &= 1 - P\left[\frac{m}{\min_k h(a_k)} > 6d\right] - P\left[\frac{m}{\min_k h(a_k)} < \frac{d}{6}\right] \\ &\geq 1 - \frac{1}{6} - \frac{1}{6} \\ &= \frac{2}{3}. \end{aligned}$$

□