CS 683 Advanced Design and Analysis of Algorithms April 11, 2008

Lecture 32: Counting the Number of Distinct Elements in a Strem

Instructor: John Hopcroft *Scribe: Cristian Danescu Niculescu-Mizil*

Counting the number of occurences of a given symbol in a stream

Consider the problem of counting the number of occurrences of the symbol 1 in a stream of 0's and 1's. An exact solution requires $\log n$ bits of storage, where n is the length of the stream (in the worst case there would be n $1's$ in the stream).

If we are satisfied with finding and storing only the logarithm of the number of occurrences, then the we could achieve in $\log \log n$ space by using the following algorithm:

- set k=0
- for each 1 in the sequence add 1 to k with probability $\frac{1}{2^k}$

Prop.: The estimated number of 1's is $2^k - 1$

Dem.: We can think about the algorithm as a coin flipping process: each time we see a 1 we flip a biased coin with probability $p = \frac{1}{2^k}$ of heads; if we obtain heads we increment k. The expected number of 1's we need to see before updating k is the expected number of coin flips before heads occurs:

$$E(\#flips) = p + 2(1-p)p + 3(1-p)^2 p + \cdots = \frac{1}{p} = 2^k \tag{1}$$

because this is a geometric series. Therefore, the number of flip coins (i.e. the number of ones we see) before we reach the current value of k is: $\#1's = 1 + 2 + 4 + \cdots + 2^{k-1} = 2^k - 1$.

Counting the number of distinct elements in a stream

Consider the problem of finding the number of distinct elements that appear in a stream. If we have m possible elements $a_1, \cdots, a_m$ then for exact solution we would need m space (in the worst case all m elements can be present).

Instead we are considering an approximation which answers the question "Is the number of distinct elements greater than M?". The following algorithm will answer "yes" with probability at least 0.865 if the number of distinct elements is greater than $2M$ and will answer "yes" with probability at most 0.4 if the number of distinct elements is less than $\frac{M}{2}$:

- produce a hash $d : \{1, 2, \cdots, m\} \to \{1, 2, \cdots, M\}$, where $M > \sqrt{m}$
- compute $h(a_i)$ for each element in the sequence and say "yes" if $h(a_i) = 1$

Analysis: For an element a_i the $Prob(h(a_i) = 1) = \frac{1}{M}$. If we have d distinct elements in the stream, then the probability that for all a_i in the stream $h(a_i) \neq 1$ is $\left(1 - \frac{1}{M}\right)^d$. Therefore:

- if $d \leq \frac{M}{2}$ the probability that no element hashes to 1 is $\leq \left(1 - \frac{1}{M}\right)^{M/2} = \frac{1}{\sqrt{e}} \geq 0.6$, thus the probability that some element hashes to 1 (and that the algorithm returns "yes") is ≤ 0.4
- if $d \geq 2M$ then the probability that $h(a) \neq 1$ for all elements is $\geq \left(1 - \frac{1}{M}\right)^{2M} = \frac{1}{e^2} = 0.135$, thus the probability that some element hashes to 1 (and that the algorithm returns "yes") is ≥ 0.865

Obs: We can obtain better probabilities by running the algorithm in parallel and combining the results as discussed in the following lecture.

Next we will present an alternative method which uses $O(\log m)$ space. Let $S \subseteq \{1, 2, \cdots, m\}$ the subset of indexes of elements appearing in the stream (we consider $|S| \leq \sqrt{m}$).

If S would be selected uniformly at random from $\{1, 2, \cdots, m\}$ then we could find the size of S by finding the minimal element in S:

$$min \simeq \frac{m}{|S| + 1} \Rightarrow \frac{m}{min} - 1 \tag{2}$$

However, the elements of S are not selected uniformly at random from $\{1, 2, \cdots, m\}$. In order to correct for this we take a hash function h from $\{1, 2, \cdots, m\}$ such that $h(i)$ is selected uniformly at random from $\{1, 2, \cdots, m\}$. But, if this function is completely random, then in order to store h we need m space; we will instead use a 2-universal hash function, which is sufficient for our purpose:

Def.: The set of has functions $H = \{h | h : \{1, 2, \cdots, m\} \rightarrow \{1, 2, \cdots, m\}\}$ is 2-universal if for all x and y, $x \neq y$, in $\{1, 2, \cdots, m\}$ and for all z and w

$$Prob(h(x) = z \; and \; h(y) = w) = \frac{1}{m^2} \tag{3}$$

for a randomly chosen hash function $h \in H$.

An example of such an 2-universal hash functions that we can use is $H = \{h_a b | h_{ab}(x) = ax + b \mod m\}$. Note that for for such a hash function we only need to store two variables, a and b.

To see that H is 2-universal we observe that $h(x) = z$ and $h(y) = w$ when:

$$\begin{pmatrix} x & 1 \\ y & 0 \end{pmatrix} \begin{pmatrix} a \\ y \end{pmatrix} = \begin{pmatrix} b \\ b \end{pmatrix} \mod m \tag{4}$$

and that when $x \neq y$ the matrix is invertible and hence there is a unique solution for a and b. Therefore:

$$Prob(h(x) = z \; and \; h(y) = w) = \frac{1}{m^2} \tag{5}$$