— 29 —

Approximating the Number of Distinct Elements in a Stream

"See? Genuine-sounding indignation. I programmed that myself. It's the first thing you need in a university environment: the ability to take offense at any slight, real or imagined."

Robert Sawyer, Factoring Humanity

29.1. Counting number of distinct elements

29.1.1. First order statistic

Let $X_1, \ldots, X_u$ be u random variables uniformly distributed in $[0, 1]$. Let $Y = \min(X_1, \ldots, X_u)$. The value Y is the ***first order statistic*** of $X_1, \ldots, X_u$.

For a continuous variable X, the ***probability density function*** (i.e., ***pdf***) is the "probability" of X having this value. Since this is not well defined, one looks on the ***cumulative distribution function*** $F(x) = \mathbb{P}[X \leq]$. The pdf is then the derivative of the cdf. Somewhat abusing notations, the pdf of the X_is is $\mathbb{P}[X_i = x] = 1$.

The following proof is somewhat dense, check any standard text on probability for more details.

Lemma 29.1.1. *The* probability density function *of* Y *is* $f(x) = \binom{u}{1} 1 (1-x)^{u-1}$.

Proof: Considering the pdf of X_1 being x, and all other X_is being bigger. We have that this pdf is

$$g(x) = \mathbb{P}\Big[(X_1 = x) \cap \bigcap_{i=2}^{u} (X_i > X_1)\Big] = \mathbb{P}\Big[\bigcap_{i=2}^{u} (X_i > X_1) \;\Big|\; X_1 = x\Big] \mathbb{P}[X_1 = x] = (1-x)^{u-1}.$$

Since every one of the X_i has equal probability to realize Y, we have $f(x) = ug(x)$.

Lemma 29.1.2. *We have* $\mathbb{E}[Y] = \frac{1}{u+1}$, $\mathbb{E}[Y^2] = \frac{2}{(u+1)(u+2)}$, *and* $\mathbb{V}[Y] = \frac{u}{(u+1)^2(u+2)}$.

Proof: Using integration by guessing, we have

$$\begin{aligned}\mathbb{E}[Y] &= \int_{y=0}^{1} y\,\mathbb{P}[Y = y]\,\mathrm{d}y = \int_{y=0}^{1} y \cdot \binom{u}{1} 1(1-y)^{u-1}\,\mathrm{d}y = \int_{y=0}^{1} uy(1-y)^{u-1}\,\mathrm{d}y \\ &= \left[-y(1-y)^u - \frac{(1-y)^{u+1}}{u+1}\right]_{y=0}^{1} = \frac{1}{u+1}.\end{aligned}$$

Using integration by guessing again, we have

$$\begin{aligned}\mathbb{E}[Y^2] &= \int_{y=0}^{1} y^2\,\mathbb{P}[Y = y]\,\mathrm{d}y = \int_{y=0}^{1} y^2 \cdot \binom{u}{1} 1(1-y)^{u-1}\,\mathrm{d}y = \int_{y=0}^{1} uy^2(1-y)^{u-1}\,\mathrm{d}y \\ &= \left[-y^2(1-y)^u - \frac{2y(1-y)^{u+1}}{u+1} - \frac{2(1-y)^{u+2}}{(u+1)(u+2)}\right]_{y=0}^{1} = \frac{2}{(u+1)(u+2)}.\end{aligned}$$

We conclude that

$$\mathbb{V}[Y] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \frac{2}{(u+1)(u+2)} - \frac{1}{(u+1)^2} = \frac{1}{u+1}\left(\frac{2}{u+2} - \frac{1}{u+1}\right) = \frac{u}{(u+1)^2(u+2)}$$

29.1.2. The algorithm

A single estimator. Assume that we have a perfectly random hash function h that randomly maps $N = \{1, \ldots, n\}$ to $[0, 1]$. Assume that the stream has u unique numbers in N. Then the set $\{h(s_1), \ldots, h(s_m)\}$ contains u random numbers uniformly distributed in $[0.1]$. The algorithm as such, would compute $X = \min_i h(s_i)$.

Explanation. Note, that X is *not* an estimator for u – instead, as $\mathbb{E}[X] = 1/(u+1)$, we are estimating $1/(u+1)$. The key observation is that an $1 \pm \varepsilon$ estimator for $1/(u+1)$, is $1 \pm O(\varepsilon)$ estimator for $u+1$, which is in turn an $1 \pm O(\varepsilon)$ estimator for u.

Lemma 29.1.3. *Let $\varepsilon, \varphi \in (0,1)$ be parameters. Given a stream $\mathcal{S}$ of items from $\{1, \ldots, n\}$ one can return an estimate X, such that $\mathbb{P}\Big[(1-\varepsilon/4)\frac{1}{u+1} \leq X \leq (1+\varepsilon/4)\frac{1}{u+1}$ $1 - \varphi$, where u is the number of unique elements in $\mathcal{S}$. This requires $O\Big(\frac{1}{\varepsilon^2} \log \frac{1}{\varphi}\Big)$ space.*

Proof: The basic estimator Y has $\mu = \mathbb{E}[Y] = \frac{1}{u+1}$ and $\nu = \mathbb{V}[Y] = \frac{u}{(u+1)^2(u+2)}$. We now plug this estimator into the mean/median framework. By Lemma 29.1.2, for c some absolute constant, this requires maintaining M estimators, where M is larger than

$$c \frac{4 \cdot 16\nu}{\varepsilon^2 \mu^2} \log \frac{1}{\varphi} = O\left(\frac{u^2}{\varepsilon^2 u^2} \log \frac{1}{\varphi}\right) = O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\varphi}\right).$$

Observe that if $(1-\varepsilon/4)\frac{1}{u+1} \leq X \leq (1+\varepsilon/4)\frac{1}{u+1}$ then

$$\frac{u+1}{1-\varepsilon/4} - 1 \geq \frac{1}{X} - 1 \geq \frac{u+1}{1+\varepsilon/4} - 1,$$

which implies

$$(1+\varepsilon)u \geq \frac{(1+\varepsilon/4)u}{1-\varepsilon/4} \geq \frac{u+\varepsilon/4}{1-\varepsilon/4} \geq \frac{1}{X} - 1 \geq \frac{u+1}{1+\varepsilon/4} - 1 \geq (1-\varepsilon)u.$$

Namely, $1/X - 1$ is a good estimator for the number of distinct elements.

The algorithm revisited. Compute X as above, and output the quantity $1/X - 1$.

This immediately implies the following.

Lemma 29.1.4. *Under the unreasonable assumption that we can sample perfectly random functions from $\{1, \ldots, n\}$ to $[0,1]$, and storing such a function requires $O(1)$ words, then one can estimate the number of unique elements in a stream, using $O(\varepsilon^{-2} \log \varphi^{-1})$ words.*

29.2. Sampling from a stream with "low quality" randomness

Assume that we have a stream of elements $\mathcal{S} = s_1, \ldots, s_m$, all taken from the set $\{1, \ldots, n\}$. In the following, let $\mathrm{set}(S)$ denote the set of values that appear in S. That is

$$F_0 = F_0(\mathcal{S}) = |\mathrm{set}(S)|$$

is the number of distinct values in the stream $\mathcal{S}$.

Assume that we have a random sequence of bits $\mathcal{B} \equiv B_1, \ldots, B_n$, such that $\mathbb{P}[B_i = 1] = p$, for some p. Furthermore, we can compute B_i efficiently. Assume that the bits of $\mathcal{B}$ are pairwise independent.

The sampling algorithm. When the ith arrives s_i, we compute B_{s_i}. If this bit is 1, then we insert s_i into the random sample R (if it is already in R, there is no need to store a second copy, naturally).

This defines a natural random sample

$$R = \{i \mid B_i = 1 \text{ and } i \in S\} \subseteq S.$$

Lemma 29.2.1. *For the above random sample R, let $X = |R|$. We have that $\mathbb{E}[X] = p\nu$ and $\mathbb{V}[X] = p\nu - p^2\nu$, where $\nu = F_0(\mathcal{S})$ is the number of district elements in S.*

Proof: Let $X = |R|$, and we have

$$\mathbb{E}[X] = \mathbb{E}\Bigl[\sum_{i \in S} B_i\Bigr] = \sum_{i \in S} \mathbb{E}[B_i] = p\nu.$$

As for the $\mathbb{E}[X^2]$, we have

$$\mathbb{E}[X^2] = \mathbb{E}\Bigl[\bigl(\sum_{i \in S} B_i\bigr)^2\Bigr] = \sum_{i \in S} \mathbb{E}[B_i^2] + 2 \sum_{i,j \in S,\, i<j} \mathbb{E}\bigl[B_i B_j\bigr] = p\nu + 2 \sum_{i,j \in S,\, i<j} \mathbb{E}[B_i]\,\mathbb{E}[B_j] = p\nu + 2p^2 \binom{\nu}{2}$$

As such, we have

$$\mathbb{V}[X] = \mathbb{V}[|R|] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = p\nu + 2p^2\binom{\nu}{2} - p^2\nu^2 = p\nu + 2p^2\frac{\nu(\nu-1)}{2} - p^2\nu^2$$
$$= p\nu + p^2\nu(\nu-1) - p^2\nu^2 = p\nu - p^2\nu.$$

Lemma 29.2.2. *Let $\varepsilon \in (0, 1/4)$. Given $O(1/\varepsilon^2)$ space, and a parameter N. Consider the task of estimating the size of $F_0 = |\mathrm{set}(\mathcal{S})|$, where $F_0 > N/4$. Then, the algorithm described below outputs one of the following:*
(A) $F_0 > 2N$.
(B) Output a number ρ such that $(1-\varepsilon)F_0 \leq \rho \leq (1+\varepsilon)F_0$.
(Note, that the two options are not disjoint.) The output of this algorithm is correct, with probability $\geq 7/8$.

Proof: We set $p = \frac{c}{N\varepsilon^2}$, where c is a constant to be determined shortly. Let $T = pN = O(1/\varepsilon^2)$. We sample a random sample R from S, by scanning the elements of S, and adding $i \in S$ to R if $B_i = 1$, If the random sample is larger than $8T$, at any point, then the algorithm outputs that $|S| > 2N$.

In all other cases, the algorithm outputs $|R|/p$ as the estimate for the size of S, together with R.

To bound the failure probability, consider first the case that $N/4 < |\mathrm{set}(\mathcal{S})|$. In this case, we have by the above, that

$$\mathbb{P}[|X - \mathbb{E}[X]| > \varepsilon\,\mathbb{E}[X]] \leq \mathbb{P}\left[|X - \mathbb{E}[X]| > \varepsilon\frac{\mathbb{E}[X]}{\sqrt{\mathbb{V}[X]}}\sqrt{\mathbb{V}[X]}\right] \leq \varepsilon^2\frac{\mathbb{V}[X]}{(\mathbb{E}[X])^2} \leq \frac{1}{8},$$

if $\frac{\mathbb{V}[X]}{\varepsilon^2(\mathbb{E}[X])^2} \leq \frac{1}{8}$, For $\nu = F_0 \geq N/4$, this happens if $\frac{p\nu}{\varepsilon^2 p^2 \nu^2} \leq \frac{1}{8}$. This in turn is equivalent to $8/\varepsilon^2 \leq p\nu$. This is in turn happens if

$$\frac{c}{N\varepsilon^2}\cdot\frac{N}{4} \geq \frac{8}{\varepsilon^2},$$

which implies that this holds for $c = 32$. Namely, the algorithm in this case would output a $(1 \pm \varepsilon)$-estimate for $|S|$.

If the sample get bigger than $8T$, then the above readily implies that with probability at least $7/8$, the size of S is at least $(1-\varepsilon)8T/p > 2N$, Namely, the output of the algorithm is correct in this case.

Lemma 29.2.3. *Let $\varepsilon \in (0, 1/4)$ and $\varphi \in (0, 1)$. Given $O(\varepsilon^{-2}\log\varphi^{-1})$ space, and a parameter N, and the task is to estimate F_0 of $\mathcal{S}$, given that $F_0 > N/4$. Then, there is an algorithm that would output one of the following:*
(A) $F_0 > 2N$.
(B) Output a number ρ such that $(1-\varepsilon)F_0 \leq \rho \leq (1+\varepsilon)F_0$.

(Note, that the two options are not disjoint.) The output of this algorithm is correct, with probability $\geq 1 - \varphi$.

Proof: We run $O(\log \varphi^{-1})$ copies of the of Lemma 29.2.2. If half of them returns that $F_0 > 2N$, then the algorithm returns that $F_0 > 2N$. Otherwise, the algorithm returns the median of the estimates returned, and return it as the desired estimated. The correctness readily follows by a repeated application of Chernoff's inequality.

Lemma 29.2.4. *Let $\varepsilon \in (0, 1/4)$. Given $O(\varepsilon^{-2} \log^2 n)$ space, one can read the stream $\mathcal{S}$ once, and output a number ρ, such that $(1-\varepsilon)F_0 \leq \rho \leq (1+\varepsilon)F_0$. The estimate is correct with high probability (i.e., $\geq 1-1/n^{O(1)}$).*

Proof: Let $N_i = 2^i$, for $i = 1, \ldots, M = \lceil \lg n \rceil$. Run M copies of Lemma 29.2.3, for each value of N_i, with $\varphi = 1/n^{O(1)}$. Let $Y_1, \ldots, Y_M$ be the outputs of these algorithms for the stream. A prefix of these outputs, are going to be "$F_0 > 2N_i$", Let j be the first Y_j that is a number. Return this number as the desired estimate. The correctness is easy – the first estimate that is a number, is a correct estimate with high probability. Since $N_M \geq n$, it also follows that Y_M must be a number. As such, there is a first number in the sequence, and the algorithm would output an estimate.

More precisely, there is an index i, such that $N_i/4 \leq F_0 \leq 2F_0$, and Y_i is a good estimate, with high probability. If any of the Y_j, for $j < i$, is an estimate, then it is correct (again) with high probability.

29.3. Bibliographical notes

29.4. From previous lectures

Theorem 29.4.1. *Let $\mathcal{D}$ be a non-negative distribution with $\mu = \mathbb{E}[\mathcal{D}]$ and $\nu = \mathbb{V}[\mathcal{D}]$, and let $\varepsilon, \varphi \in (0,1)$ be parameters. For some absolute constant $c > 0$, let $M \geq 24\left\lceil \frac{4\nu}{\varepsilon^2\mu^2} \right\rceil \ln \frac{1}{\varphi}$, and consider sampling variables $X_1, \ldots, X_M \sim \mathcal{D}$. One can compute, in, $O(M)$ time, a quantity Z from the sampled variables, such that*

$$\mathbb{P}\Big[(1-\varepsilon)\mu \leq Z \leq (1+\varepsilon)\mu\Big] \geq 1 - \varphi.$$

Theorem 29.4.2 (Chebyshev's inequality). *Let X be a real random variable, with $\mu_X = \mathbb{E}[X]$, and $\sigma_X = \sqrt{\mathbb{V}[X]}$, assuming $\sigma_X > 0$. Then, for any $t > 0$, we have $\mathbb{P}\big[|X - \mu_X| \geq t\sigma_X\big] \leq 1/t^2$.*

Lemma 29.4.3. *Let $X_1, \ldots, X_n$ be n independent Bernoulli trials, where $\mathbb{P}[X_i = 1] = p_i$, and $\mathbb{P}[X_i = 0] = 1 - p_i$, for $i = 1, \ldots, n$. Let $X = \sum_{i=1}^{b} X_i$, and $\mu = \mathbb{E}[X] = \sum_i p_i$. For $\delta \in (0, 4)$, we have*

$$\mathbb{P}\big[X > (1+\delta)\mu\big] < \exp(-\mu\delta^2/4),$$

Theorem 29.4.4. *let p be a prime number, and pick independently and uniformly k values $b_0.b_1, \ldots, b_{k-1} \in \mathbb{Z}_p$, and let $g(x) = \sum_{i=0}^{k-1} b_i x^i \bmod p$. Then the random variables*

$$Y_0 = g(0), \ldots, Y_{p-1} = g(p-1).$$

are uniformly distributed in $\mathbb{Z}_p$ and are k-wise independent.

References

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge, UK: Cambridge University Press, 1995.