Chapter 26

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

By Sariel Har-Peled, April 26, 2022

26.1. Counting number of distinct elements

26.1.1. 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 \textit{first order statistic} of \( X_1, \ldots, X_u \).

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

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

\textbf{Lemma 26.1.1.} \textit{The probability density function of} \( Y \) \textit{is} \( f(x) = \binom{u}{i} 1(1 - x)^{u-1} \).

\textit{Proof:} Considering the \textit{pdf} of \( X_1 \) being \( x \), and all other \( X_i \)s being bigger. We have that this \textit{pdf} is

\[
g(x) = P[(X_1 = x) \cap \bigcap_{i=2}^u (X_i > X_1)] = P\left[\bigcap_{i=2}^u (X_i > X_1) \mid X_1 = x\right] P[X_1 = x] = (1 - x)^{u-1}.
\]

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

\textbf{Lemma 26.1.2.} \textit{We have} \( E[Y] = \frac{1}{u+1} \), \( E[Y^2] = \frac{2}{(u+1)(u+2)} \), \textit{and} \( \forall [Y] = \frac{u}{(u+1)^2(u+2)} \).

\textit{Proof:} Using integration by guessing, we have

\[
E[Y] = \int_{y=0}^{1} y P[Y = y] \, dy = \int_{y=0}^{1} y \cdot \binom{u}{1} (1 - y)^{u-1} \, dy = \int_{y=0}^{1} uy(1 - y)^{u-1} \, dy
\]

\[
= \left[-y(1 - y)^u - \frac{(1 - y)^{u+1}}{u + 1}\right]_{y=0}^{1} = \frac{1}{u + 1}.
\]

\textsuperscript{a}This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit \url{http://creativecommons.org/licenses/by-nc/3.0/} or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

1
Using integration by guessing again, we have
\[
\mathbb{E}[Y^2] = \int_{y=0}^{1} y^2 \mathbb{P}[Y = y] \, dy = \int_{y=0}^{1} y^2 \cdot \left( \frac{u}{u+1} \right) (1-y)^{u-1} \, dy = \int_{y=0}^{1} uy^2 (1-y)^{u-1} \, dy
\]
\[
= \left[ -y^2 (1-y)^u - 2y (1-y)^{u+1} \right]_{y=0}^{1} = \frac{2}{(u+1)(u+2)}. \]
We conclude that
\[
\forall Y = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \frac{2}{(u+1)(u+2)} - \frac{1}{(u+1)^2} = \frac{1}{u+1} \left( \frac{2}{u+2} - \frac{1}{u+1} \right) = \frac{u}{(u+1)^2(u+2)}. \]

26.1.2. 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 \( u \), which is in turn an \( 1 \pm O(\varepsilon) \) estimator for \( u \).

Lemma 26.1.3. Let \( \varepsilon, \varphi \in (0, 1) \) be parameters. Given a stream \( S \) of items from \( \{1, \ldots, n\} \) one can return an estimate \( X \), such that \( \mathbb{P}\left[ \left( (1-\varepsilon/4) \frac{1}{u+1} \leq X \leq (1+\varepsilon/4) \frac{1}{u+1} \right) \right] \geq 1 - \varphi \), where \( u \) is the number of unique elements in \( S \). This requires \( O\left( \frac{1}{\varepsilon^2} \log \frac{1}{\varphi} \right) \) space.

Proof: The basic estimator \( Y \) has \( \mu = \mathbb{E}[Y] = \frac{1}{u+1} \) and \( v = \forall Y = \frac{u}{(u+1)^2(u+2)} \). We now plug this estimator into the mean/median framework. By Lemma 26.1.2, for some absolute constant, this requires maintaining \( M \) estimators, where \( M \) is larger than
\[
c \cdot \frac{4 \cdot 16\varphi}{\varepsilon^2 \mu^2} \log \frac{1}{\varphi} = O\left( \frac{u^2}{\varepsilon^2 \mu^2 \log \frac{1}{\varphi}} \right) = O\left( \frac{1}{\varepsilon^2} \log \frac{1}{\varphi} \right). \]

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 26.1.4. 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.
26.2. Sampling from a stream with “low quality” randomness

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

$$F_0 = F_0(S) = |\text{set}(S)|$$

is the number of distinct values in the stream $S$.

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

**The sampling algorithm.** When the $i$th 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 26.2.1.** For the above random sample $R$, let $X = |R|$. We have that $\mathbb{E}[X] = pv$ and $\forall[X] = pv - p^2v$, where $v = F_0(S)$ is the number of distinct elements in $S$.

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

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i \in S} B_i\right] = \sum_{i \in S} \mathbb{E}[B_i] = pv.$$

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

$$\mathbb{E}[X^2] = \mathbb{E}\left[(\sum_{i \in S} B_i)^2\right] = \sum_{i \in S} \mathbb{E}[B_i^2] + 2 \sum_{i,j \in S, i < j} \mathbb{E}[B_iB_j] = pv + 2 \sum_{i,j \in S, i < j} \mathbb{E}[B_i] \mathbb{E}[B_j] = pv + 2p^2\left(\frac{v}{2}\right).$$

As such, we have

$$\forall[X] = \forall[|R|] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = pv + 2p^2\left(\frac{v}{2}\right) - p^2v^2 = pv + 2p^2\frac{v(v-1)}{2} - p^2v^2$$

$$= pv + p^2v(v-1) - p^2v^2 = pv - p^2v. \quad \blacksquare$$

**Lemma 26.2.2.** Let $\varepsilon \in (0, 1/4)$. Given $O(1/\varepsilon^2)$ space, and a parameter $N$. Consider the task of estimating the size of $F_0 = |\text{set}(S)|$, where $F_0 > N/4$. Then, the algorithm described below outputs one of the following:

(A) $F_0 > 2N$.

(B) Output a number $\rho$ 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 $R$. 

3
To bound the failure probability, consider first the case that \( N/4 < |\text{set}(S)| \). In this case, we have by the above, that
\[
P[|X - \mathbb{E}[X]| > \varepsilon \mathbb{E}[X]] \leq P\left[|X - \mathbb{E}[X]| > \varepsilon \frac{\mathbb{E}[X]}{\sqrt{\mathbb{V}[X]}} \right] \leq \varepsilon^2 \frac{\mathbb{V}[X]}{(\mathbb{E}[X])^2} \leq \frac{1}{8},
\]
if \( \frac{\mathbb{V}[X]}{\varepsilon^2 (\mathbb{E}[X])^2} \leq \frac{1}{8} \). For \( v = F_0 \geq N/4 \), this happens if \( \frac{PV}{\varepsilon^2 pV} \leq \frac{1}{8} \). This in turn is equivalent to \( 8/\varepsilon^2 \leq pv \). 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 26.2.3.** 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 \( S \), 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 \( \rho \) 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 26.2.2. 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 26.2.4.** Let \( \varepsilon \in (0, 1/4) \). Given \( O(\varepsilon^{-2}\log^2 n) \) space, one can read the stream \( S \) once, and output a number \( \rho \), 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 \( i = 1, \ldots, M = \lfloor \log n \rfloor \). Run \( M \) copies of Lemma 26.2.3, 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.

### 26.3. Bibliographical notes

### 26.4. From previous lectures

**Theorem 26.4.1.** Let \( \mathcal{D} \) be a non-negative distribution with \( \mu = \mathbb{E}[\mathcal{D}] \) and \( v = \mathbb{V}[\mathcal{D}] \), and let \( \varepsilon, \varphi \in (0, 1) \) be parameters. For some absolute constant \( c > 0 \), let \( M \geq 24\left(\frac{4v}{\varepsilon^2 \mu^2}\right)\ln\frac{1}{\varphi} \), 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

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

**Theorem 26.4.2 (Chebyshev’s inequality).** Let $X$ be a real random variable, with $\mu_X = \mathbb{E}[X]$, and $\sigma_X = \sqrt{\mathbb{V}[X]}$. Then, for any $t > 0$, we have

$$\mathbb{P}[|X - \mu_X| \geq t\sigma_X] \leq 1/t^2.$$  

**Lemma 26.4.3.** 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 $X = \sum_{i=1}^n X_i$, and $\mu = \mathbb{E}[X] = \sum_i p_i$. For $\delta \in (0, 4)$, we have

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

**Theorem 26.4.4.** 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 $g(x) = \sum_{i=0}^{k-1} b_i x^i \mod p$. 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**