Special Topics in Theoretical Computer Science Spring 2019

Lecture 1 — 1/7/2019

Huy L. Nguyen *Scribe: Huy L. Nguyen*

In this course, we will discuss algorithmic methods for dealing with massive datasets. Below we start with a deceptively simple problem that already illustrates many important techniques: counting. Specifically, the algorithm needs to support the following three operations:

- Initialize the counter n to 0
- Increase the counter n by 1
- Report an estimate $\hat{n}$ of n

A trivial algorithm is to store n using $\lceil \log n \rceil$ bits. This turns out to be the best possible if we require that $\hat{n}$ is always equal to n. Indeed, if there is an algorithm using m bits then the algorithm can only has 2^m different states. To be able to count exactly from 1 to n, the algorithm needs to be able to return n different outputs, which requires n different states. Thus, $2^m \geq n$ i.e. $m \geq \log n$.

To overcome this lower bound, we will relax the requirements in two ways. First, the answer $\hat{n}$ only needs to approximate n up to a $1 \pm \varepsilon$ factor. Second, we allow the algorithm to be randomized and fails with a small probability δ. In other words, we want $\hat{n}$ to satisfy

$$\Pr[|\hat{n} - n| > \varepsilon n] \leq \delta$$

1 Probability review

Before delving into some randomized algorithms, we first review some basic facts about probability.

Lemma 1.1 (Linearity of expectation). *For any two random variables X and Y, we have*

$$\mathbb{E}[aX + bY] = a\,\mathbb{E}[X] + b\,\mathbb{E}[Y]$$

Lemma 1.2 (Markov's inequality). *For any nonnegative random variable X and $a > 0$, we have*

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

Lemma 1.3 (Chebyshev's inequality). *For any random variable X and $a > 0$, we have*

$$\Pr[|X - \mathbb{E}[X]| \geq a] \leq \frac{Var(X)}{a^2}$$

Lemma 1.4 (Chernoff bound). *Suppose $X_1, X_2, \ldots, X_n$ are independent random variables with $X_i \in [0, 1]$. Let $X = \sum_i X_i$ and $\mu = \mathbb{E}\,X$. If $0 < \varepsilon < 1$ then*

$$\Pr[|X - \mu| \geq \varepsilon\mu] \leq 2\exp(-\varepsilon^2\mu/3)$$

2 Morris' algorithm

In 1978, Morris gave the following algorithm for the counter problem:

- Initialize X to 0
- On each update, with probability 2^{-X} we increase X by 1. Otherwise, X remains the same.
- We report $\hat{n} = 2^X - 1$ as an estimate for n

The distribution of $\hat{n}$ is complex so we will try to partially understand it through only two statistics: the expectation and the variance. Let $X^{(i)}$ be the value of X after i updates.

Lemma 2.1.

$$\mathbb{E}[2^{X^{(n)}}] = n$$

Proof. We prove the lemma by induction. For the base case, observe that $2^{X^{(0)}} = 2^0 = 1$. For the inductive case, we assume that $\mathbb{E}[2^{X^{(n)}}] = n + 1$ and will analyze $\mathbb{E}[2^{X^{(n+1)}}]$.

$$\begin{aligned}
\mathbb{E}[2^{X^{(n+1)}}] &= \sum_{j=1}^{\infty} \Pr[2^{X^{(n)}} = j] \cdot \mathbb{E}[2^{X^{(n+1)}} | 2^{X^{(n)}} = j] \\
&= \sum_{j} \Pr[2^{X^{(n)}} = j] \cdot \left(\frac{1}{j} \cdot 2j + \left(1 - \frac{1}{j}\right) \cdot j \right) \\
&= \sum_{j} \Pr[2^{X^{(n)}} = j] \cdot (j + 1) \\
&= \sum_{j} \Pr[2^{X^{(n)}} = j] \cdot j + \sum_{j} \Pr[2^{X^{(n)}} = j] \\
&= \mathbb{E}[2^{X^{(n)}}] + 1
\end{aligned}$$

□

Lemma 2.2.

$$\mathbb{E}[2^{2X^{(n)}}] = \frac{3}{2}n^2 + \frac{3}{2}n + 1$$

Proof. We use induction as before.

$$\begin{aligned}
\mathbb{E}[2^{2X^{(n+1)}}] &= \sum_{j=1}^{\infty} \Pr[2^{X^{(n)}} = j] \cdot \mathbb{E}[2^{2X^{(n+1)}} | 2^{X^{(n)}} = j] \\
&= \sum_{j} \Pr[2^{X^{(n)}} = j] \cdot \left(\frac{1}{j} \cdot 4j^2 + \left(1 - \frac{1}{j}\right) \cdot j^2 \right) \\
&= \sum_{j} \Pr[2^{X^{(n)}} = j] \cdot (j^2 + 3j) \\
&= \sum_{j} \Pr[2^{X^{(n)}} = j] \cdot j + \sum_{j} \Pr[2^{X^{(n)}} = j] \\
&= \mathbb{E}[2^{2X^{(n)}}] + 3\,\mathbb{E}[2^{X^{(n)}}] \\
&= \frac{3}{2}(n+1)^2 + \frac{3}{2}(n+1) + 1
\end{aligned}$$

□

By calculation, we can show that $\mathrm{Var}(\hat{n}) \leq n^2/2$. Unfortunately, this variance is still too high to obtain an accurate estimate with high probability.

To reduce the variance, we use multiple independent copies of Morris' algorithm and compute the average of their outputs. Let's call this algorithm Morris+. That is, we compute independent estimator $\hat{n}_1, \ldots, \hat{n}_s$ and the output is

$$\hat{n} = \frac{1}{s} \sum_i \hat{n}_i$$

The key properties we are using is that $\mathbb{E}[\hat{n}] = \mathbb{E}[\hat{n}_1]$ but $Var(\hat{n}) = \frac{1}{s} Var(\hat{n}_1)$. Thus, for $s = 10/\varepsilon^2$, by Chebyshev's inequality, we have

$$\Pr[|\hat{n} - n| \geq \varepsilon n] \leq \frac{n^2/(2s)}{\varepsilon^2 n^2} \leq \frac{1}{20}$$

The next thing we need to fix is that the failure probability remains high at 1/20. To reduce the failure probability to δ, we use t independent copies of Morris+ and output the median of the outputs. Let Y_i be the indicator random variable of whether the ith copy is correct. As we argued before, $Y_i = 1$ with probability 19/20 and $Y_i = 0$ with probability 1/20. Our algorithm succeeds if at least $t/2$ copies succeed. Thus, by the Chernoff bound, the probability that the algorithm fails is bounded by

$$\Pr[\sum_i Y_i < t/2] \leq 2\exp(-t/3) \leq \delta$$

for $t = \Theta(\log(1/\delta))$.