(CS602) Applied Algorithms 09 Jan, 2014

Lecture 2: Computing the length of the input

Lecturer: Nutan Limaye *Scribe: Nutan Limaye*

In the last class, we started with two problems:
Problem 1: Given a stream of numbers, output the length of the input stream.
Problem 2: Given a graph as a set of edges, output the maximum matching in a graph.

We saw a $O(n \log n)$ space 2-approximation algorithm for Problem 2. We gave an algorithm for Problem 1. Today we will analyze the algorithm.

2.1 Algorithm for Problem 1 and analysis

Let us start by recalling the algorithm.

$Y \leftarrow 0$;
while *there exists x_i, an input element* **do**
  $Y \leftarrow Y + 1$ w.p. $\frac{1}{2^Y}$;
  $Y \leftarrow Y$ w.p. $1 - \frac{1}{2^Y}$;
end
Output $2^Y - 1$

The algorithm increments the counter with lower and lower probability as the length of the input increases. We will first analyze the expected value and the variance of the output after i steps.

Lemma 2.1.1. $\mathbb{E}(2^{Y_i}) = i + 1$

Proof.

$$\begin{aligned}
\mathbb{E}(2^{Y_i}) &= \sum_{j=0}^{\infty} \left(\left[\mathbb{E}(2^{Y_i}) | Y_{i-1} = j \right] \Pr\left[Y_{i-1} = j \right] \right) \\
&= \sum_{j=0}^{\infty} \left(2^{j+1} \frac{1}{2^j} + 2^j (1 - \frac{1}{2^j}) \right) \Pr\left[Y_{i-1} = j \right] \\
&= \sum_{j=0}^{\infty} 2^j \Pr\left[Y_{i-1} = j \right] + \sum_{j=0}^{\infty} \Pr\left[Y_{i-1} = j \right] \\
&= \mathbb{E}(2^{Y_{i-1}}) + 1
\end{aligned}$$

Using the recurrence, we get the lemma. □

Lemma 2.1.2. $\mathbb{V}ar(2^{Y_i}) = \frac{i(i-1)}{2}$

Proof.

$$\begin{aligned}
\mathbb{E}(2^{2Y_i}) &= \sum_{j=0}^{\infty} \Big(\left[\mathbb{E}(2^{2Y_i}) | Y_{i-1} = j\right] \Pr\left[Y_{i-1} = j\right] \Big) \\
&= \sum_{j=0}^{\infty} \left(2^{2(j+1)} \frac{1}{2^j} + 2^{2j}(1 - \frac{1}{2^j}) \right) \Pr\left[Y_{i-1} = j\right] \\
&= \sum_{j=0}^{\infty} 2^{2j} \Pr\left[Y_{i-1} = j\right] + 3 \sum_{j=0}^{\infty} 2^j \Pr\left[Y_{i-1} = j\right] \\
&= \mathbb{E}(2^{2Y_{i-1}}) + 3i
\end{aligned}$$

Solving the recurrence, we get that $\mathbb{E}(2^{2Y_i}) = 1 + \sum_{k=1}^{i} 3k = 1 + \frac{3i(i+1)}{2}$. Therefore, $\mathbb{V}ar(2^{Y_i}) = 1 + \frac{3i(i+1)}{2} - (i+1)^2 = \frac{i(i-1)}{2}$ □

Lemmas 2.1.1 indicates that the expected value of the output of the algorithm is equal to the actual length of the input. This is a good sign. Lemma 2.1.2 indicates that the variance is not too large. This again is useful. Now using Chebyshev we have $\Pr\left[|2^{Y_{n+1}} - (n+1)| \geq 0.9(n+1)\right] \leq \frac{n(n-1)}{1.62(n+1)^2} < \frac{3}{4}$. This tells us that with probability at least $1/4$ the algorithm gives a 0.9 approximation.

Definition 2.1.3. *A randomized algorithm $\mathcal{A}$ computing a Boolean function f is said to be an (ε, δ) algorithm for f if for every input x,* $\Pr\left[(1-\varepsilon)f(x) \leq A(x) \leq (1+\varepsilon)f(x)\right] \geq 1 - \delta$.

In this sense, the algorithm we have is an $(0.9, 3/4)$ algorithm for computing the length of the input.

2.2 Improving approximation guarantee

We now describe the standard trick used to increase the approximation gurantee. Let us call the algorithm designed in Section 2.1 as A_1. Given an $\varepsilon > 0$ and algorithm A_1, we give another algorithm A_2 such that $\Pr\left[(1-\varepsilon)f(x) \leq A_2(x) \leq (1+\varepsilon)f(x)\right] \geq 2/3$.

for $j = 1$ *to* t **do**
  $Y^{(j)} \leftarrow 0$;
end
while *there exists x_i, an input element* **do**
  for $j = 1$ *to* t **do**
    $Y^{(j)} \leftarrow Y^{(j)} + 1$ w.p. $\frac{1}{2^{Y^{(j)}}}$;
    $Y^{(j)} \leftarrow Y^{(j)}$ w.p. $1 - \frac{1}{2^{Y^{(j)}}}$;
  end
end
Output $\frac{\sum_{i=1}^{t} 2^{Y^{(i)}} - 1}{t}$

Here, t is a parameter, which we will fix shortly. Let Z_n denote the output of A_2 for inputs of length n. It is easy to see that $\mathbb{E}(Z_n) = n$ and $\mathbb{V}ar(Z_n) = \frac{n(n-1)}{2t}$. Therefore, by applying Chebyshev's inequality we get $\Pr[|Z_n - n| \geq \varepsilon n] \leq \frac{n(n-1)}{2t\varepsilon^2 n^2} < \frac{1}{2t\varepsilon^2}$. By setting $t = \lceil \frac{3}{2\varepsilon^2} \rceil$ we get that $\Pr[|Z_n - n| \geq \varepsilon n] < 1/3$. That is, we get for every input x $\Pr[(1-\varepsilon)f(x) \leq A_2(x) \leq (1+\varepsilon)f(x)] \geq 2/3$. Suppose $s(n)$ is the space used by A_1 for inputs of length n, then as $t = O(1/\varepsilon^2)$, the space used by A_2 is $ts(n) = O(s(n)/\varepsilon^2)$.

Remark 2.2.1. *Observe that the error probability that we obtained here could be made small enough by setting t appropriately. Suppose we needed the error probability to be δ, the space used by the algorithm would take an additional blow of $O(1/\delta)$. In the next section, we show how to reduce the error by increasing the space by only $O(\log(\frac{1}{\delta}))$.*

2.3 Decreasing the error probability

Consider the following modified algorithm:

for $j = 1$ *to* t **do**
  for $\ell = 1$ *to* k **do**
    $Y^{(j,\ell)} \leftarrow 0$;
  end
end
while *there exists* x_i*, an input element* **do**
  for $j = 1$ *to* t **do**
    for $\ell = 1$ *to* k **do**
      $Y^{(j,\ell)} \leftarrow Y^{(j,\ell)} + 1$ w.p. $\frac{1}{2^{Y^{(j,\ell)}}}$;
      $Y^{(j,\ell)} \leftarrow Y^{(j,\ell)}$ w.p. $1 - \frac{1}{2^{Y^{(j,\ell)}}}$;
    end
  end
end
Output Median of $\left(\frac{\sum_{j=1}^{t} 2^{Y^{(j,1)}} - 1}{t}, \frac{\sum_{j=1}^{t} 2^{Y^{(j,2)}} - 1}{t}, \ldots, \frac{\sum_{j=1}^{t} 2^{Y^{(j,k)}} - 1}{t} \right)$

Let us call this algorithm A_3. Here, let t be as fixed in Section 2.2, i.e. $t = \frac{3}{2\varepsilon^2}$. Let us define $Z_\ell = \frac{\sum_{j=1}^{t} 2^{Y^{(j,\ell)}} - 1}{t}$ for $1 \leq \ell \leq k$. And let Y_ℓ be a 0-1 random variable which is set to 1 if $(1+\varepsilon)n \leq Y_\ell \leq (1+\varepsilon)n$ for $1 \leq \ell \leq k$. Then we know that for $1 \leq \ell \leq k$, $\mathbb{E}(Y_\ell) = 2/3$. That is, in expectation, about 2/3rd of the Y_ℓs are in the correct range of values. The algorithm A_3 outputs the medial of these Y_ℓs. If more than half of the Y_ℓs have the values in the right range, the median will be in the right range. Therefore, to bound the error of the algorithm we need to bound the probability of the event that strictly less than half of the Y_ℓs are in the right range. Let $Y = \sum_{\ell=1}^{k} Y_\ell$. We wish to bound the probability that $Y < k/2$. Note that $\mathbb{E}(Y) = 2k/3$ by linearity of expectations. $\Pr[Y > k/2] = \Pr[|Y - \mathbb{E}(Y)| \geq k/6] = \Pr\left[|Y - \mathbb{E}(Y)| \geq \frac{\mathbb{E}(Y)}{4}\right]$. This can be bounded by

using the Chernoff bound as follows: $\Pr\left[|Y - \mathbb{E}(Y)| \geq \frac{\mathbb{E}(Y)}{4}\right] \leq 2.e^{-\frac{2k}{16 \cdot 3}}$. To make this smaller than $\delta > 0$, we need to $k = O(\log(\frac{1}{\delta}))$.

To summarize, we have designed an algorithm A_3 which runs for $O\left(\frac{1}{\varepsilon^2} \log\left(\frac{1}{\delta}\right)\right)$ iterations and has the following guarantee: $\Pr\left[(1-\varepsilon)n \leq A_3(x) \leq (1+\varepsilon)n\right] \geq 1 - \delta$.