Lecture 8: Sampling based approach for distinct elements

Lecturer: Nutan Limaye *Scribe: Nutan Limaye*

In the last class we completed the analysis of Count sketch algorithm. Today we will give a sampling based approach for estimating distinct elements.

Recall the distinct element problem deals with given a stream of data $x_1, x_2, \ldots, x_n$, where for all i $x_i \in [m]$, counting the number of distinct elements in the stream. As a first step towards solving this problem using sampling, we will look at the restricted version of the same problem and design a sampling algorithm for it. We call this version, the gap version of the problem, $\mathsf{GapDist}_k$.

Given: $\tilde{x} = x_1, x_2, \ldots, x_n$, where for all $1 \leq i \leq n$, $x_i \in [m]$, and $k \in \mathbb{N}$

Output "Yes" if the number of distinct elements in $\tilde{x}$ is $> 2^{k+2}$

"No" if the number of distinct elements in $\tilde{x}$ is $< 2^{k-2}$

8.1 Naive Sampling algorithm for $\mathsf{GapDist}_k$

We first give an algorithm which uses (in the worst case) $O(m)$ number of *independent* random bits. Later we show how one can raplace independent random bits by pairwise random bits.

Pick every element of $[m]$ into the set S with probability $\frac{1}{2^k}$;
Sum $\leftarrow$ 0;
while *there exists x, an input element* **do**
  if $x \in S$ **then**
    Sum $\leftarrow$ Sum +1;
  end
end
Output "Yes" iff Sum > 0;

Algorithm 1: Algorithm with independent random bits

We now argue the correctness the above algorithm and bound its error probability. Let $\tilde{x}$ be the given input. Let D denote the set of distinct elements in $\tilde{x}$. Let F_0 denote $|D|$.

Suppose $F_0 < 2^{k-2}$. Then the probability that the algorithm makes an error is:

$$
\begin{aligned}
\Pr[\text{Algorithm makes an error}] &= \Pr[\text{Sum } > 0] \\
&= \Pr[\exists x \in D \text{ s.t. } x \in S] \\
&\le \sum_{x \in D} \Pr[x \in S] && \text{(By union bound)} \\
&= \frac{|D|}{2^k} \\
&< \frac{1}{4} && \text{(By our assumption that } |D| < 2^{k-2}\text{)}
\end{aligned}
$$

Suppose $F_0 > 2^{k+2}$. Then the probability that the algorithm makes an error is:

$$
\begin{aligned}
\Pr[\text{Algorithm makes an error}] &= \Pr[\text{Sum } = 0] \\
&= \Pr[\forall x \in D : x \notin S] \\
&\le \prod_{x \in D} \Pr[x \notin S] && \text{(As the samples are independent)} \\
&= \left(1 - \frac{1}{2^k}\right)^{2^{k+2}} && \text{(By our assumption that } |D| > 2^{k+2}\text{)} \\
&< \left(\frac{1}{e}\right)^4 && \text{(Using } \left(1 - \frac{1}{x}\right)^x = \frac{1}{e}\text{)}
\end{aligned}
$$

By the above calculations, we get that the algorithm correctly decides $\mathsf{GapDist}_k$ with probability at least $3/4$.

Note that, in the above calculations we used the fact that our samples are independent. Let us do the calculations once again, but in such a way that the analysis will go through even if we draw samples using pairwise independence. Let X_j be a 0-1 random variable defined as follows: $X_j = 1$ if $j \in S$ and $X_j = 0$ otherwise. Let $X = \sum_{j \in D} X_j$. Note that $\Pr[X_j = 1] = \frac{1}{2^k}$ for all j. Therefore, $\mathbb{E}(X_j) = \frac{1}{2^k}$ and $\mathbb{E}(X) = \frac{|D|}{2^k}$. Suppose X_js are either purely independent or pairwise independent, we know that $\mathbb{V}ar(X) \le \mathbb{E}(X)$ (by the property of pairwise independent random variables).

Suppose $F_0 < 2^{k-2}$. Then the probability that the algorithm makes an error is:

$$
\begin{aligned}
\Pr[\text{Algorithm makes an error}] &= \Pr[\text{Sum } > 0] \\
&= \Pr[X > 0] \\
&= \Pr[X \geq 1] \\
&\leq \frac{|D|}{2^k} && \text{(By Markov's inequality)} \\
&< \frac{1}{4} && \text{(By our assumption that } |D| < 2^{k-2}\text{)}
\end{aligned}
$$

On the other hand, suppose $F_0 > 2^{k+2}$. Then the probability that the algorithm makes an error is:

$$
\begin{aligned}
\Pr[\text{Algorithm makes an error}] &= \Pr[X = 0] \\
&\leq \Pr[|X - \mathbb{E}(X)| \geq \mathbb{E}(X)] \\
&\leq \frac{\mathbb{V}ar(X)}{\mathbb{E}(X)^2} && \text{(By Chebyshev's inequality)} \\
&\leq \frac{1}{\mathbb{E}(X)} && (\mathbb{V}ar(X) \leq \mathbb{E}(X)) \\
&< \frac{1}{4} && \text{(Using } |D| > 2^{k+2} \text{ and } \mathbb{E}(X) = \frac{|D|}{2^k}\text{)}
\end{aligned}
$$

Once again, by the above calculations, we get that the algorithm correctly decides $\mathsf{GapDist}_k$ with probability at least $3/4$.

By using standard Chernoff argument, we can bring down the error probability down to δ using at most $O(\log \frac{1}{\delta})$ bits.

Now, we change the algorithm so that independent samples can now be changed by pairwise independent samples.

```
Pick h from a family of pairwise independent random functions
F = {h : [m] → {0,1}^k}.;
Sum ← 0;
while there exists x, an input element do
    if h(x) = 0^k then
        Sum ← Sum +1;
    end
end
Output "Yes" iff Sum > 0;
```

Algorithm 2: Algorithm with pairwise independent random variables

For the analysis, we define $X_j = 1$ iff $h(j) = 0^k$ and $X = \sum_{j \in D} X_j$ as before. The analysis of the algorithm is the same as our second analysis.

Let $\mathcal{A}_\delta^k$ denote this randomized algorithm for $\mathsf{GapDist}_k$ with error at most δ. In the next section we use this algorithm to approximate F_0.

8.2 Approximating F_0 using $\mathcal{A}_\delta^k$

In this section we will use $\mathcal{A}_\delta^1, \mathcal{A}_\delta^2, \ldots, \mathcal{A}_\delta^{\lceil \log m \rceil}$ to get an 8-approximation for F_0. In the exercise, you are asked to improve it to $(1+\varepsilon)$-approximation.

Let $\mathcal{A}_{\delta'}$ be the following algorithm:

```
for i = ⌈log m⌉ downto 1 do
    if A_δ^i outputs 0 then
        next i;
    end
    else
        Output 2^i;
    end
end
```

Suppose on some fixed input $\mathcal{A}_\delta^i$ outputs 0 for all $i \geq j$ but $\mathcal{A}_\delta^{j-1}$ outputs 1 and suppose also that all the answers are correct. Then this tells us that the answer must be certainly smaller than 2^{j+2} and definitely more than 2^{j-3}. Therefore, if the algorithm outputs 2^j then it will be 8-approximation. But unfortunately, not all answers may be correct. $\Pr\left[A_{\delta'} \text{ makes an error}\right] \leq \Pr\left[\exists A_\delta^i \text{ makes an error}\right] \leq \lceil \log m \rceil \cdot \delta$. By making $\delta = \frac{\delta'}{\lceil \log m \rceil}$, we can make the error bounded by δ'.

8.3 Space analysis of $\mathcal{A}_\delta^i$ and $\mathcal{A}_{\delta'}$

To pick a random function from a family of pairwise independent functions, we need $O(k \cdot \log m)$ bits and to store 'Sum' we need $O(\log n)$ bits. To bring down the overall error to δ, we need to run $O(\log(\frac{1}{\delta}))$ copies of Algorithm 2. Therefore, total number of bits stored by $\mathcal{A}_\delta^i$ is $O(\log(\frac{1}{\delta}) \cdot (k \cdot \log m + \log n))$. Say $s = O(\log(\frac{1}{\delta}) \cdot (k \cdot \log m + \log n))$.

Now, the algorithm $\mathcal{A}_{\delta'}$ simultaeously runs $\lceil \log m \rceil$ copies of $\mathcal{A}_\delta^i$, one for every $1 \leq i \leq \lceil \log m \rceil$. This takes space $O(\lceil \log m \rceil \cdot s)$. Finally, for the error to be bounded by δ', we need to set $\delta = \frac{\delta'}{\lceil \log m \rceil}$. Putting it together, we get that the space used by $\mathcal{A}_{\delta'}$ can be bounded by $O\left(\lceil \log m \rceil \cdot \log(\frac{\lceil \log m \rceil}{\delta'}) \cdot (k \cdot \log m + \log n)\right)$. This gives us an 8-approximation for F_0 with probability $1 - \delta'$.

8.4 Exercises

Exercise 1. *Modify Algorithm 2, $\mathcal{A}_\delta^i$ and $\mathcal{A}_{\delta'}$ to obtain for every $\varepsilon > 0$, $(1+\varepsilon)$-approximation algorithm for F_0. Analyze the space used by your algorithm.*