CS7800: Advanced Algorithms

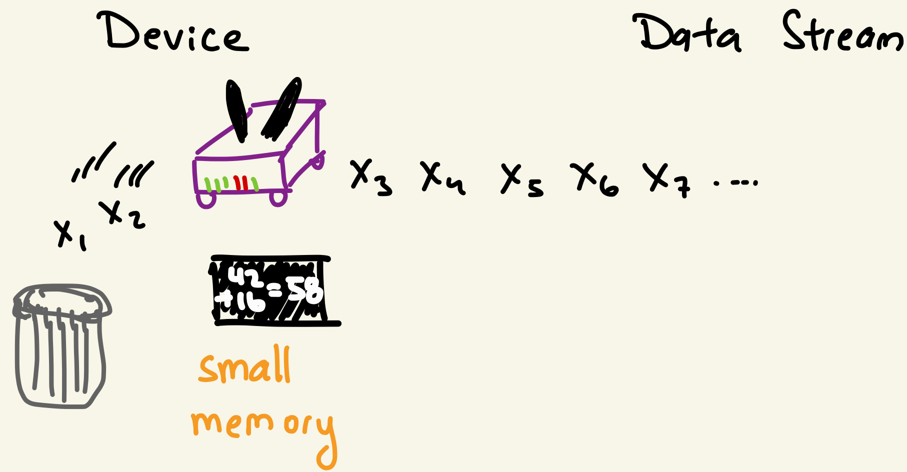
Lecture 20 : Streaming I

- Streaming Algorithms
- Distinct Elements

Jonathan Ullman

11-29-2022

Streaming Algorithms



- Stream of elements $x_1, x_2, x_3, \ldots$ from a universe $\mathcal{U}$
- Want to only store a small number of bits S at any time
- Want to <u>approximate</u> some function $f(x_1, x_2, x_3, \ldots)$

Warmup: Uniform Sampling in a Stream

Inputs: a stream of elements $x_1, x_2, x_3, \ldots$ from $\mathcal{U}$

Goal: output a uniformly random element s from the stream

If stream has length n
then $\mathbb{P}(s = x_i) = \frac{1}{n}$

If you know the stream length n:

- Pick $i \in \{1, 2, \ldots, n\}$ at random
- Wait until you see x_i
- Write down x_i

$\log_2 n$	$\log_2 n$	$\log_2 \lvert\mathcal{U}\rvert$
i	count so far	x_i

$O(\log n + \log \lvert\mathcal{U}\rvert)$

Reservoir Sampling

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \cdots \quad x_i$

$s = x_1$

$s = \frac{1}{2}x_1 + \frac{1}{2}x_2$

w.p. $\frac{1}{2}$, let $s = x_2$
w.p. $\frac{1}{2}$, do nothing

$$s = \frac{2}{3}\left(\frac{1}{2}x_1 + \frac{1}{2}x_2\right) + \frac{1}{3}x_3$$
$$= \frac{1}{3}x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3$$

w.p. $\frac{1}{3}$, let $s = x_3$
w.p. $\frac{2}{3}$, do nothing

w.p. $\frac{1}{i}$, let $s = x_i$
w.p. $1 - \frac{1}{i}$, do nothing

space is $O(\log n + \log |U|)$

Reservoir Sampling

```
for i=1,2,3,...:
    get x_i
    w.p. 1/i set s_i = x_i
    w.p. 1-1/i set s_i = s_{i-1}
return s_n
```

only store s_i

Thm: For every n, s_n is a random sample from $\{x_1, x_2, \ldots, x_n\}$

Pf: Base case is easy

Assume that s_{n-1} is random from $\{x_1 \ldots x_{n-1}\}$ $\left(\begin{array}{c}\mathbb{P}(s_{n-1} = x_i) = \frac{1}{n-1} \\ \text{for } i=1,\ldots,n-1\end{array}\right)$

$\mathbb{P}(s_n = x_n) = \frac{1}{n}$ (by construction)

$\mathbb{P}(s_n = x_i) = \left(1 - \frac{1}{n}\right) \cdot \left(\frac{1}{n-1}\right) = \frac{1}{n}$ $\square$

for $i \le n-1$

Reservoir Sampling

for 2 samples w/o replacement

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad \cdots \quad x_i$

$s = \{x_1, x_2\}$

$\frac{1}{3}(x_1, x_2)$
$+ \frac{1}{3}(x_2 x_3)$
$+ \frac{1}{3}(x_1, x_3)$

w.p. $1 - \frac{2}{i}$, hold

w.p. $\frac{2}{i}$, keep x_i and one of your samples at random

$$\frac{2\cancel{(i-1)}}{i\cancel{(i-1)}} = \frac{2}{i}$$

w.p. $\frac{1}{3}$, do nothing

w.p. $\frac{2}{3}$, pick one of s at random and replace the other with x_3

Reservoir Sampling

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad \cdots$

$0.623 \quad 0.715 \quad 0.112 \quad 0.100 \quad 0.563$

$$h(x) = md5(s \| x)$$

Counting Distinct Elements

Inputs: a stream of elements $x_1, x_2, x_3, \ldots$ from $\mathcal{U}$

Goal: the (approximate) number of distinct elements in the stream

$DE_x = |\{u \in \mathcal{U} : x_i = u \text{ for some } i\}|$

c-approximate means

$\frac{1}{c} \cdot DE \leq \widetilde{DE} \leq c \cdot DE$

with high probability

stream: 1 1 3 4 8 3 1 2 8 3

#distinct: 5

Baseline: storing all the elements you've seen so far takes

$DE_x \cdot \log |\mathcal{U}|$ bits of space

store a flag for each element takes $|\mathcal{U}|$ bits of space

A Simplification: Threshold Testing

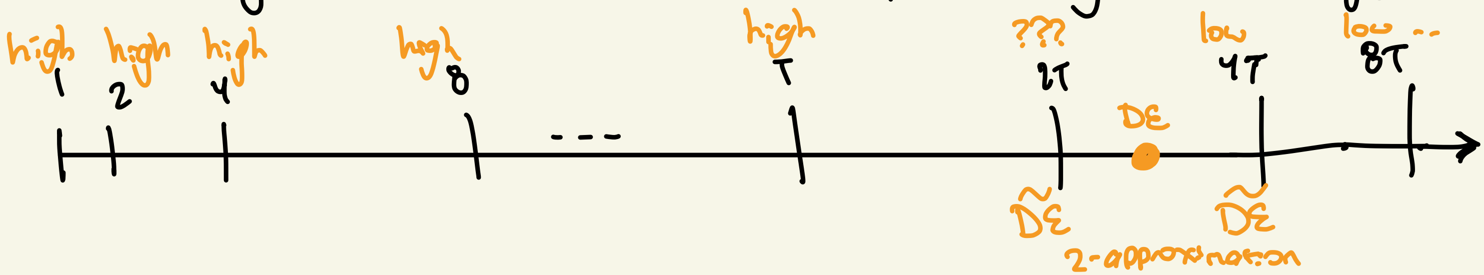
Goal is to design an algorithm A_T such that

① If $DE \le T$ then $\mathbb{P}(A_T = \text{low}) \geq 1-\delta$

② If $DE \geq 2T$ then $\mathbb{P}(A_T = \text{high}) \geq 1-\delta$

2 is arbitrary

Can run $\log_2|\mathcal{U}|$ copies of $A_1, A_2, A_4, A_8, \ldots, A_{|\mathcal{U}|}$ in parallel to get a 2-approximation with probability $\geq 1 - \delta \cdot \log_2|\mathcal{U}|$

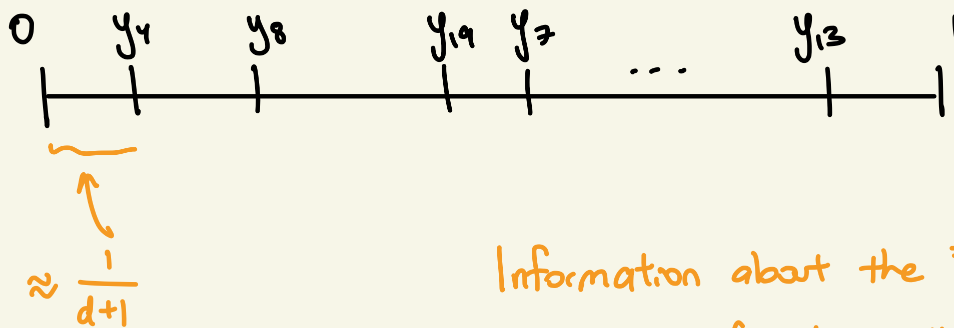


Threshold Testing Distinct Elements I

suppose the distinct elements are $y_1, \dots, y_d$

suppose I assign a random number in $[0,1]$ to each

use hashing to assign the same number to multiple copies of one elt



Information about the # of distinct elements is found in the width of these gaps

Threshold Testing Distinct Elements I

Choose a uniformly random hash function

$h : \mathcal{U} \to \{0, 1, 2, \ldots, \underbrace{T}_{\text{threshold}} - 1\}$

For each x_i in the stream:

  if $h(x_i) = 0$ output high

Output low

Suppose $DE \le T$

$\mathbb{P}(\text{low}) = \left(1 - \frac{1}{T}\right)^T \approx \frac{1}{e}$

want $\mathbb{P}(\text{low}) \ge 1 - \delta$

Suppose $DE \ge 2T$

$\mathbb{P}(\text{low}) = \left(1 - \frac{1}{T}\right)^{2T}$

$= \left(\left(1 - \frac{1}{T}\right)^T\right)^2 \approx \frac{1}{e^2}$

want $\mathbb{P}(\text{low}) \le \delta$