CS7800: Advanced Algorithms

Lecture 21 : Streaming II

- Finish Distinct Elements

Jonathan Ullman

12-2-2022

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 $DE \leq T$ 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

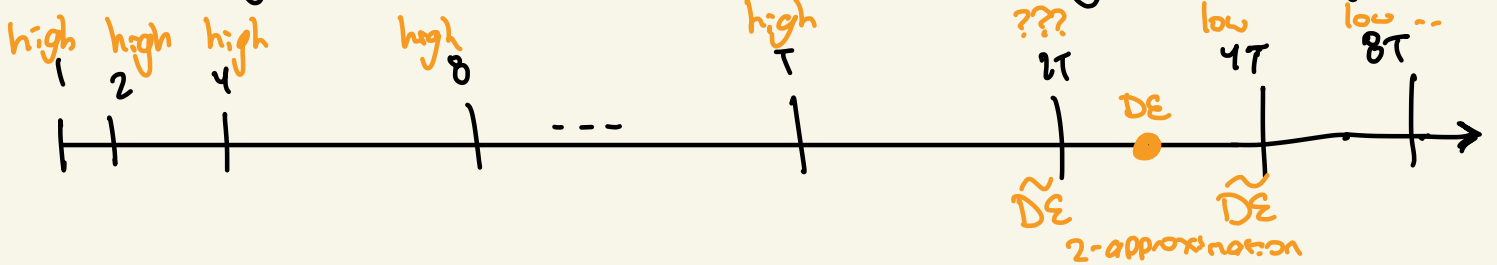
$1+\varepsilon$

$\log_{1+\varepsilon}|U|$

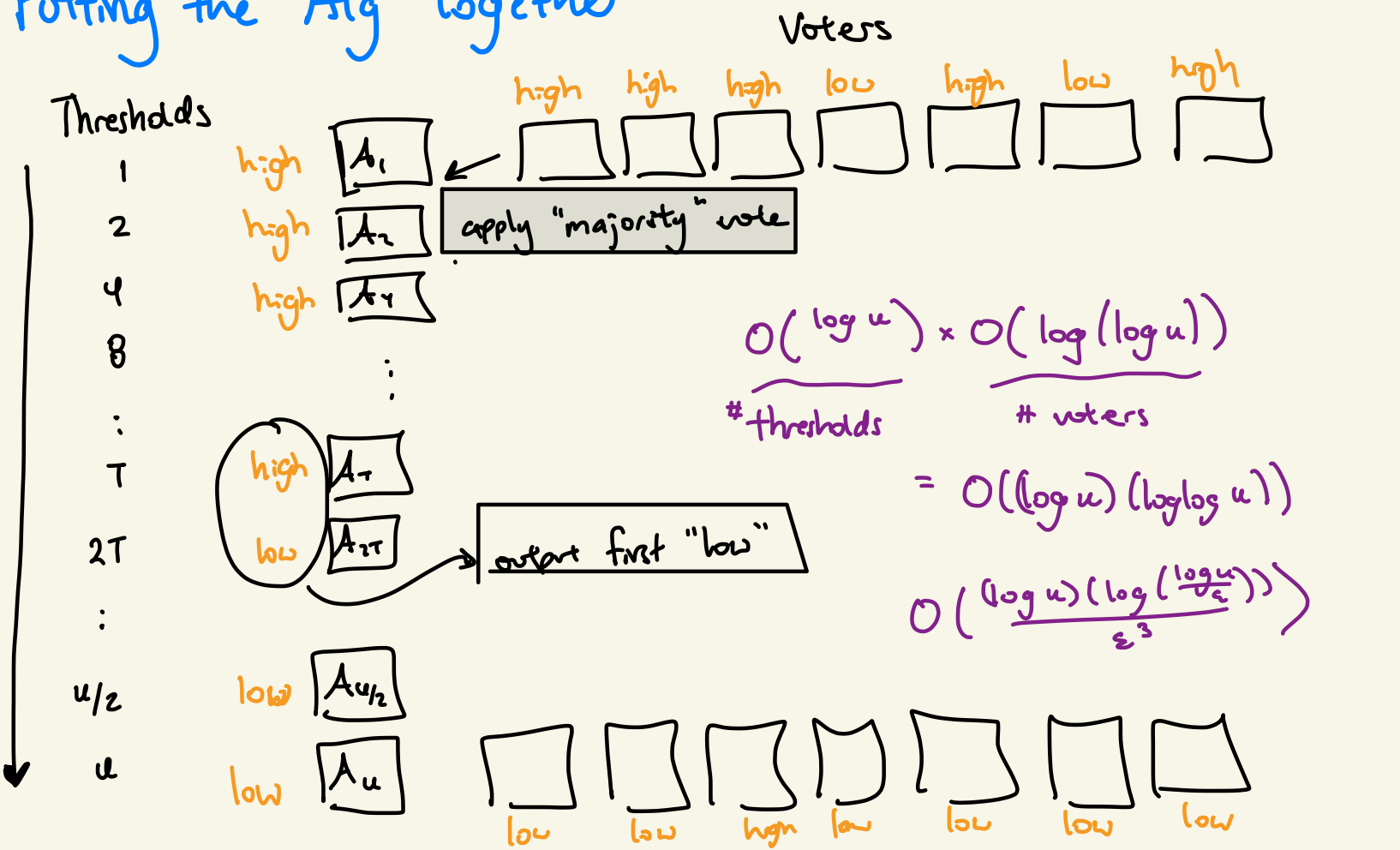
$(\log_2|U|)/(\log_2(1+\varepsilon))$

$\approx \frac{\log_2|U|}{\varepsilon}$

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



Putting the Alg Together



Threshold Testing Distinct Elements I

Choose a uniformly random hash function

$h: U \to \{0, 1, 2, \ldots, T-1\}$ (threshold)

For each x_i in the stream:

  If $h(x_i) = 0$ output high

Output low

Suppose $DE \leq T$

$$\mathbb{P}(\text{low}) = \left(1 - \frac{1}{T}\right)^{DE} \geq \frac{1}{e} \approx .36$$

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

Suppose $DE \geq 2T$

$$\mathbb{P}(\text{low}) = \left(1 - \frac{1}{T}\right)^{DE} = \left(\left(1 - \frac{1}{T}\right)^{T}\right)^{2} \leq \frac{1}{e^2} \approx .14$$

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

Amplifying Success

$\frac{1}{e} \approx .36 \quad \frac{1}{e^2} \approx .14 \quad \frac{1}{2}\left(\frac{1}{e}+\frac{1}{e^2}\right) \approx .25$

1 = low
0 = high

Chernoff Bound: If $Z_1, \ldots, Z_n$ are independent $\{0,1\}$ random variables, $Z = Z_1 + \ldots + Z_n$ and $\mu = \mathbb{E}(Z)$ then

$$\mathbb{P}(Z > \mu + \delta n) \leq e^{-n\delta^2/4}$$

$$\mathbb{P}(Z < \mu - \delta n) \leq e^{-n\delta^2/4}$$

(low DE)

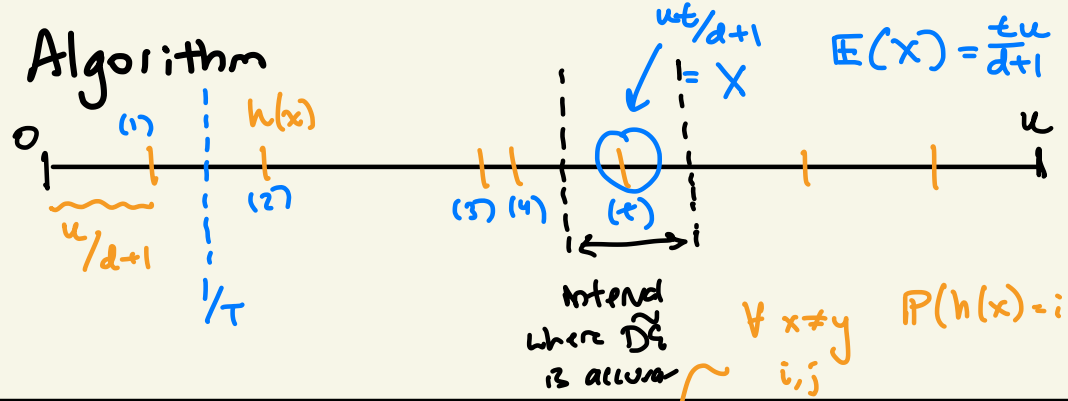
$\mu = .36n$

$\mathbb{P}(Z < .25n)$

$= \mathbb{P}(Z < \mu - .11n) \leq e^{-n(.11)^2/4} \approx e^{-\frac{n}{400}}$ (want) $\leq \beta$

$n \geq 400 \cdot \ln(1/\beta) = O(\ln(1/\beta))$

A Better Algorithm



- Let $\mathcal{H} \subseteq \{ h : [u] \to [u] \}$ be pairwise independent
- Choose h from $\mathcal{H}$ randomly
- Let $t = \frac{1000}{\varepsilon^2}$ and store the t smallest distinct hashes $(x_i, h(x_i))$ (If you don't see t then just count exactly)
- Let X be the t^{th} smallest $h(x_i)$ you've seen
- Return $\tilde{D\varepsilon} = \frac{t \cdot u}{X} \left(\approx d+1 \right)$

Analysis

Thm: $\mathbb{P}\left(|\widetilde{D\varepsilon} - D\varepsilon| > \varepsilon D\varepsilon\right) \leq \frac{1}{100}$

- Let $\mathcal{H} \subseteq \{h : [u] \to [w]\}$ be pairwise independent
- Choose h from $\mathcal{H}$ randomly
- Let $t = \frac{1000}{\varepsilon^2}$ and store the t smallest distinct hashes $(x_i, h(x_i))$
- Let X be the t^{th} smallest $h(x_i)$ you've seen
- Return $\widetilde{D\varepsilon} = \left[\; \frac{t \cdot w}{X} \; \right]$