Algorithms for streaming data

Feb 8 2022

- Limited storage space
  - data streaming past

Notation:

Denote elements of the stream

$a_1, a_2, a_3, \ldots$

$a_i \in U$

Functions:

1. sum of all elements
2. max
3. heavy-hitters
4. median
5. # unique elements

Sampling?

Can lead to incorrect answers (if not done correctly)

Example: Find "# uniques" = # elements that occur exactly once.

Sampling approach (can lead to incorrect answers)

- Sample 10% of the stream picking each element of the stream w.p. 0.1
- Count # uniques among sampled elements
- Multiply by 10

Can lead to incorrect answer. Example ...

Suppose stream length = n

$n/2$ uniques

$n/4$ appear twice

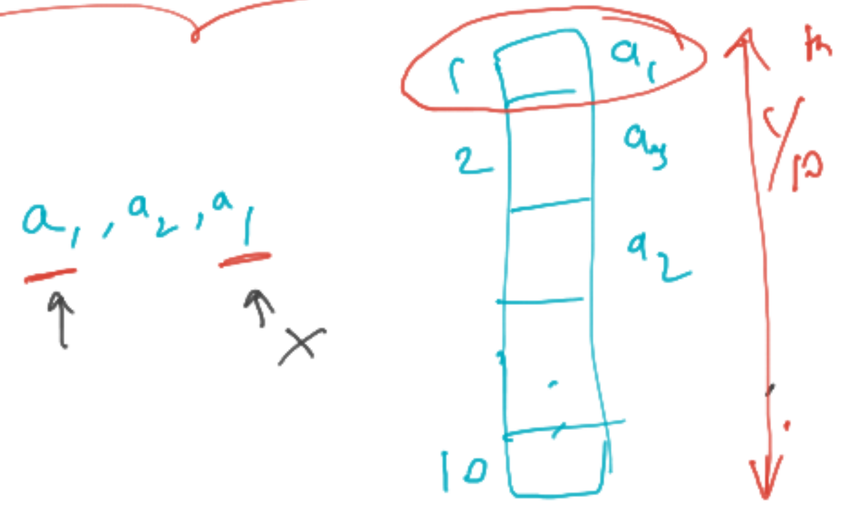
In the sampled stream,

Expected length = $\frac{n}{10}$

$$\#\text{ uniques} = \frac{n}{2} \times 0.1 + \underbrace{\frac{n}{4}\left(2 \times 0.1 - (0.1)^2\right)}$$

$$\leq \frac{n}{10}$$

Estimate $\leq \frac{n}{10} \times 10 = n$



Abstraction : view stream as a vector (in high dim. space)

Stream at time t $\quad x^t \in \mathbb{Z}^{|U|}$

$$x^t = (x_1^t, x_2^t, \ldots, x_{|U|}^t)$$

vector of length $|U|$

x_i^t = num. of times the i^{th} element of U has been seen until time t

Eg. if $a_j \in U$ appears, $x_j + 1$

- Allow both additions and deletions

E.g. $U = \{A, B, C\}$

add(A), add(B), add(A), del(B), del(A), add(C)

stream of adds and dels

$(0,0,0)$ $(1,0,0)$ $(1,1,0)$ $(2,1,0)$ $(2,0,0)$ ...

Heavy hitters: estimate "large" entries

Total num. of element seen:

distinct element = # non-zero entries

Heavy hitters:

ϵ-heavy hitters: Indices i s.t.

$$x_i > \epsilon \|x\|_1$$

$\|\cdot\|_1$ = sum of the elements

Simpler problem first: "Count-query"

At any time t, any given index i,

find an estimate of x_i^t with

error at most $\epsilon \|x^t\|_1$

i.e. $\tilde{x}_i^t \in x_i \pm \epsilon \|x\|_1$

($\tilde{x}_i$: estimate)

To the first order, we can look at i's s.t. $\tilde{x}_i^t > 0$

Hashing - based solution : Count - min Sketch

2 - step soln.

Step 1: Let $h : U \rightarrow [M]$ hash fn.

Let $A[1 \cdots M]$ array ~~for~~ storing non-negative integers

When update a_t arrives

If (a_t == (add, i))

then $A[h(a_t)]$ ++

else (i.e (del, i)

$A[h(a_t)]$ --

Estimate for x_i^t : $\tilde{x}_i^t = A[h(i)]$

$$\tilde{x}_i^t = A[h(i)]$$

$$= x_i^t + \sum_{\substack{j \in U \\ j \neq i}} x_j^t \, \mathbb{1}[h(j) = h(i)]$$

added later : (over estimate)

Collisions at index $h(i)$

$$\text{Error} = \tilde{x}_i^t - x_i^t =$$

$$\mathbb{E}[\text{error}] = \sum_{\substack{j \in U \\ j \neq i}} x_j^t \, \mathbb{E}\left[\mathbb{1}[h(j) = h(i)]\right]$$

$$= \sum_{\substack{j \in U \\ j \neq i}} x_j^t \, p[h(j) = h(i)]$$

Assume "universal hashing" $\Rightarrow \leq \sum_{j \neq i} x_j^t \cdot \frac{1}{M}$

$$\mathbb{E}[\text{error}] \leq \sum_{j \neq i} x_j^t \cdot \frac{1}{M}$$

$$\leq \frac{\|x^t\|_1}{M}$$

$$\varepsilon \|x^t\|_1, \quad \varepsilon = \frac{1}{M}$$

In expectation.

Can we give probabilistic guarantee?

$$P\left(|\hat{x}_i^t - x_i^t| < \varepsilon \|x^t\|_1 \right) \geq 1 - \delta$$

Step 2:

ℓ hash functions $h_1, h_2, \ldots h_\ell : U \to [M]$

ℓ arrays $A_1, \ldots A_\ell$

E.g.:

j^{th} element arrive: $A_k[h_k(i)]$ ++ for $k = 1, \ldots, \ell$

overall estimate $\tilde{x}_i^t = \min_{k=1}^{\ell} A_k[h_k(i)]$

For any single estimator,

$$P\left[\text{error} > \underbrace{2 \cdot \frac{\|x^t\|_1}{M}}_{\alpha}\right] \leq \frac{1}{2}$$

Markov Inequality

Non. neg. R.V. X with mean μ

$$P(X \geq \alpha) \leq \frac{\mu}{\alpha}$$

Now,

$$P\left[\text{error of } \tilde{x}_i^t > 2.\frac{\|x^t\|_1}{M}\right] = \left(\frac{1}{2}\right)^l$$

$$P\left[\left|\tilde{x}_i^t - x_i\right| \leq \varepsilon \|x^t\|_1,\right] \geq 1-\delta$$

where $M = \frac{2}{\varepsilon}$, $\delta = \left(\frac{1}{2}\right)^l$

$$\Rightarrow l = \log_2\left(\frac{1}{\delta}\right)$$

Tells us how to choose the parameters.