Median in Random Order Streams

Lecture 17

March 26, 2019

Quantiles and Selection

Input: stream of numbers $x_1, x_2, \ldots, x_n$ (or elements from a total order) and integer k

Selection: (Approximate) rank k element in the input.

Quantile summary: A compact data structure that allows approximate selection queries.

Summary of previous lecture

Randomized: Pick $\Theta(\frac{1}{\epsilon}\log(1/\delta))$ elements. With probability $(1 - 1/\delta)$ will provide ϵ-approximate quantile summary

Deterministic: ϵ-approximate quantile summary using $O(\frac{1}{\epsilon}\log^2 n)$ elements and can be improved to $O(\frac{1}{\epsilon}\log n)$ elements

Exact selection: With $O(n^{1/p}\log n)$ memory and p passes. Median in 2 passes with $O(\sqrt{n}\log n)$ memory.

Random order streams

Question: Can we improve bounds/algorithms if we move beyond worst case?

Random order streams

Question: Can we improve bounds/algorithms if we move beyond worst case?

Two models:

- Elements $x_1, x_2, \ldots, x_n$ chosen iid from some probability distribution. For instance each $x_i \in [0, 1]$
- Elements $x_1, x_2, \ldots, x_n$ chosen adversarially but stream is a uniformaly random permutation of elements.

Median in random order streams

[Munro-Paterson 1980]

Theorem

Median in $O(\sqrt{n}\log n)$ memory in one pass with high probability if stream is random order.

More generally in p passes with memory $O(n^{1/2p}\log n)$

Munro-Paterson algorithm

- Given a space parameter s algorithm stores a set of s consecutive elements seen so far in the stream
- Maintains counters ℓ and h
- ℓ is number of elements seen so far that are less than $\min S$
- h is number of elements seen so far that are more than $\max S$.
- Tries to keep ℓ and h balanced

Munro-Paterson algorithm

MP-Median (s):

```
Store the first s elements of the stream in S
ℓ = h = 0
While (stream is not empty) do
    x is new element
    If (x > max S) then h = h + 1
    Else If (x < min S) then ℓ = ℓ + 1
    Else
        Insert x into S
        If h > ℓ discard min S from S and ℓ = ℓ + 1
        Else discard max S from S and h = h + 1
endWhile
If 1 ≤ n/2 − ℓ ≤ s then
    Output n/2 − ℓ ranked element from S
Else output FAIL
```

Example

$\sigma = 1, 2, 3, 4, 5, 6, 7, 9, 10$ and $s = 3$
$\sigma = 10, 19, 1, 23, 15, 11, 14, 16, 3, 7$ and $s = 3$.

Analysis

Theorem

If $s = \Omega(\sqrt{n} \log n)$ *and stream is random order then algorithm outputs median with high probability.*

Recall: Random walk on the line

- Start at origin $\mathbf{0}$. At each step move left one unit with probability $\mathbf{1/2}$ and move right with probability $\mathbf{1/2}$.
- After n steps how far from the origin?

Recall: Random walk on the line

- Start at origin $\mathbf{0}$. At each step move left one unit with probability $1/2$ and move right with probability $1/2$.
- After n steps how far from the origin?

At time i let X_i be -1 if move to left and 1 if move to right.
Y_n position at time n
$Y_n = \sum_{i=1}^{n} X_i$

$\mathbf{E}[Y_n] = 0$ and $Var(Y_n) = \sum_{i=1}^{n} Var(X_i) = n$

By Chebyshev: $\mathbf{Pr}\left[|Y_n| \geq t\sqrt{n}\right] \leq 1/t^2$

By Chernoff:

$$\mathbf{Pr}\left[|Y_n| \geq t\sqrt{n}\right] \leq 2exp(-t^2/2).$$

Analysis

Let H_i and L_i be random variables for the values of h and ℓ after seeing i items in the random stream

Let $D_i = H_i - L_i$

Analysis

Let H_i and L_i be random variables for the values of h and ℓ after seeing i items in the random stream

Let $D_i = H_i - L_i$

Observation: Algorithm fails only if $|D_n| \geq s - 1$

Analysis

Let H_i and L_i be random variables for the values of h and ℓ after seeing i items in the random stream

Let $D_i = H_i - L_i$

Observation: Algorithm fails only if $|D_n| \geq s - 1$

Will instead analyse the probability that $|D_i| \geq s - 1$ at any i

Analysis

Lemma

Suppose $D_i = H_i - L_i \geq 0$ *and* $D_i < s - 1$.
$\Pr[D_{i+1} = D_i + 1] = H_i/(H_i + s + L_i) \leq 1/2$.

Lemma

Suppose $D_i = H_i - L_i < 0$ *and* $|D_i| < s - 1$.
$\Pr[D_{i+1} = D_i - 1] = L_i/(H_i + s + L_i) \leq 1/2$.

Thus, process behaves better than random walk on the line (formal proof is technical) and with high probability $|D_i| \leq c\sqrt{n}\log n$ for all i. Thus if $s > c\sqrt{n}\log n$ then algorithm succeeds with high probability.

Other results on selection in random order streams

[Munro-Paterson] extend analysis for $p = 1$ and show that $\Theta(n^{1/2p} \log n)$ memory sufficient for p passes (with high probability). Note that for adversarial stream one needs $\Theta(n^{1/p})$ memory

[Guha-MacGregor] show that $O(\log \log n)$-passes sufficient for exact selection in random order streams

Part I

Secretary Problem

Secretary Problem

- Stream of numbers $x_1, x_2, \ldots, x_n$ (value/ranking of items/people)
- Want to select the largest number
- Easy if we can store the maximum number
- **Online setting:** have to make a single irrevocable decision when number seen.

Secretary Problem

- Stream of numbers $x_1, x_2, \ldots, x_n$ (value/ranking of items/people)
- Want to select the largest number
- Easy if we can store the maximum number
- **Online setting:** have to make a single irrevocable decision when number seen.

Extensively studied with applications to auction design etc.

Secretary Problem

- Stream of numbers $x_1, x_2, \ldots, x_n$ (value/ranking of items/people)
- Want to select the largest number
- Easy if we can store the maximum number
- **Online setting:** have to make a single irrevocable decision when number seen.

Extensively studied with applications to auction design etc.

In the worst case no guarantees possible. What about random arrival order?

Algorithm

Assume n is known.

LearnAndPick (θ):

```
    Let y be max number seen in the first θn numbers
    Pick z the first number larger than y in the remaining stream
```

Algorithm

Assume n is known.

```
LearnAndPick (θ):
    Let y be max number seen in the first θn numbers
    Pick z the first number larger than y in the remaining stream
```

Question: Assume numbers are in random order. What is a lower bound on the probability that algorithm will pick the largest element?

Algorithm

Assume n is known.

LearnAndPick (θ):

```
    Let y be max number seen in the first θn numbers
    Pick z the first number larger than y in the remaining stream
```

Question: Assume numbers are in random order. What is a lower bound on the probability that algorithm will pick the largest element?

Observation: Let a be largest and b the second largest. Algorithm will pick a if b is in the first θn numbers and a is the residual stream.

Algorithm

Assume n is known.

LearnAndPick (θ):

```
    Let y be max number seen in the first θn numbers
    Pick z the first number larger than y in the remaining stream
```

Question: Assume numbers are in random order. What is a lower bound on the probability that algorithm will pick the largest element?

Observation: Let a be largest and b the second largest. Algorithm will pick a if b is in the first θn numbers and a is the residual stream.

If $\theta = 1/2$ then each will occur with probability roughly $1/2$ and hence $1/4$ probability.

Algorithm

Assume n is known.

LearnAndPick (θ):

```
    Let y be max number seen in the first θn numbers
    Pick z the first number larger than y in the remaining stream
```

Question: Assume numbers are in random order. What is a lower bound on the probability that algorithm will pick the largest element?

Observation: Let a be largest and b the second largest. Algorithm will pick a if b is in the first θn numbers and a is the residual stream.

If $\theta = 1/2$ then each will occur with probability roughly $1/2$ and hence $1/4$ probability.

Optimal strategy: $\theta = 1/e$ and probability of picking largest number is $1/e$. A more careful calculation.