Optimal k-Secretary with Logarithmic Memory

Mingda Qiao Wei Zhang
Massachusetts Institute of Technology

k-Secretary

K-secretary [Kleinberg 2005]:

- n numbers arrives in random order
- Choose or reject once seeing an element, up to k elements
- Decisions are irrevocable

Goal: Maximize the sum of chosen numbers

Competitive Ratio (CR): An algorithm is α-competitive if

$$\frac{\mathbb{E}[\text{sum of chosen numbers}]}{\text{sum of } k \text{ largest elements}} \geq \alpha$$

on every instance.

Prior Work

Theorem (Kleinberg 2005): There is a $1 - O\left(\frac{1}{\sqrt{k}}\right)$ competitive algorithm. (Optimal, i.e. $1 - \Omega\left(\frac{1}{\sqrt{k}}\right)$ CR is unavoidable.)

1: Read the first $B \sim \text{Bin}(n, 1/2)$ elements. Run this algorithm recursively to solve a $(k/2)$-secretary instance on those elements

2: In parallel, list the $k/2$ largest elements among them, set the $k/2$-th largest as threshold

3: Accept the rest of the string if larger than the threshold

This Work: Memory-bounded Algorithms

We study k-secretary with bounded memory usage.

Motivation:

- *Large-scale learning and optimization tasks, e.g. routing*
- *Interest in learning & decision making in streaming setting with memory constraints*
- *Potential connections with computational complexity*

Kleinberg's algorithm requires $\Omega(k)$ space in step 2.

Quantile Estimation

Quantile Estimation:

$X_1, \ldots, X_n$ are arbitrary and come in random order

Goal: Output the (approximately) k-th largest element

Error is defined as the difference between the rank and k:

$$\mathbb{E}[|rank(x) - k|]$$

Main Algorithm for QE

(Let the string be $s_{1:n}$ in random order)

$QE(n, k, a)$: Seq length n, target rank k, threshold a

(output the approximately k-th largest element among $\{s_1, s_2, \ldots, s_n\} \cap (-\infty, a)$)

- Step 0:

- If $k \leq O(m)$, run the naive algorithm

- Step 1:

- Draw $B \sim \text{Bin}(n, 1/2)$ and $B_1 \sim \text{Bin}(B, p)$

- Read the first B_1 elements, maintaining the top m elements (below a) in $M[1], M[2], \ldots, M[m]$

- Read the first B elements, maintaining the rank of $M[1], M[2], \ldots, M[m]$ among the B elements

- Step 2:

- Find i in $[m]$ such that the rank of $M[i]$ is $k/2 - k'$ for some k' in $[1, k/100]$

- If fail, give up and output the largest element

- Otherwise, return $QE(n - B, k', M[i])$

Our Results

Proposition:

For all $\alpha \in [\frac{1}{2}, 1]$, quantile estimation with memory m and error $O(k^\alpha)$ implies k-secretary with memory $m + O(1)$ and CR $1 - O(1/k^{1-\alpha})$.

Main Theorem:

- Error $O(\sqrt{k})$, with memory $O(\log k)$
- Exact selection, with memory $O(\sqrt{k})$
- ***Memory usage and error bounds are independent of n.***

Corollary: Optimal k-secretary with $O(\log k)$ memory

Proof Sketch for O(sqrt(k)) Error

Main Idea:

reducing Quantile Estimation (k) to Quantile Estimation $(k/100)$

Proof of $O(\sqrt{k})$ error:

If we could show the error for recursions decreases fast

$$E[error_k] \leq \sqrt{k} + 2E[error_{k/100}]$$

Adding them:

$$E[error_k] \leq \sqrt{k} + 2\sqrt{k/100} + 4\sqrt{k/100^2} + \ldots = O(\sqrt{k})$$