Technische Universität Berlin

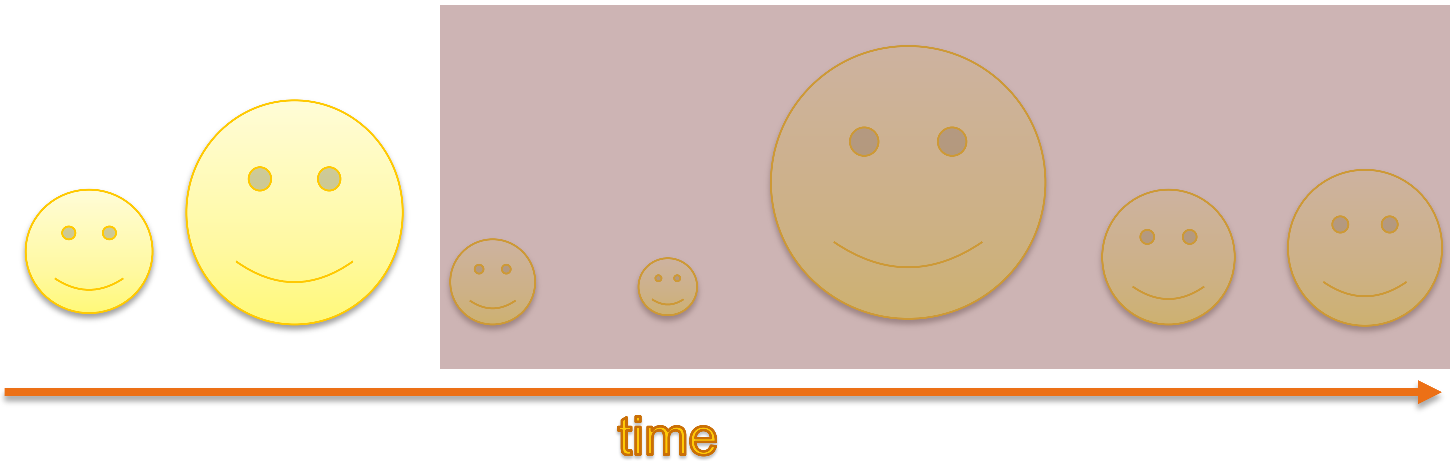
Secretary problem: Towards better bounds with ML advice

Arash Pourdamghani

Net-theory seminar
Spring 2022

Introduction to secretary problem

- A selection problem:
  - Committing to a choice before knowing all possibilites
  - Examples:
    - Finding love of you life! (ted.com/talks/hannah_fry_the_mathematics_of_love)
    - Choosing toilet at a concert! (youtube.com/watch?v=ZWib5olGbQ0)
    - Finding a student for PostDoc (vanderbei.princeton.edu/tex/PostdocProblem/PostdocProb.pdf)
    - Finding the best house: (davidwees.com/content/how-i-used-mathematics-choose-my-next-apartment/)



Introduction to secretary problem

- A selection problem:
  - Committing to a choice before knowing all possibilites
  - Examples:
    - Finding love of you life! (ted.com/talks/hannah_fry_the_mathematics_of_love)
    - Choosing toilet at a concert! (youtube.com/watch?v=ZWib5olGbQ0)
    - Finding a student for PostDoc (vanderbei.princeton.edu/tex/PostdocProblem/PostdocProb.pdf)
    - Finding the best house: (davidwees.com/content/how-i-used-mathematics-choose-my-next-apartment/)
- Basics:
  - Given n candidates with previously unknown values $v_1, \ldots, v_n \in \mathcal{R}$
  - The value of candidates is reveled in the same order
  - After seeing the i-th candidate, you either accept it or reject it

Possible goals

- Maximizing the probability of choosing the best possible candidate
  - Original Problem
- Maximizing the probability of choosing second best candidate
  - Postdoc problem
- Maximizing the expected value of chosen candidate
  - Value maximization variation
  - Any α-approximation for the classical secretary problem yields an α-approximation for the value-maximization variant.
- K-secretary problem
  - Maximizing sum, application in online auction (Kleinberg, SODA 2005)
  - Graphic matroid: select a subset of edges of maximum weight under the constraint that this subset is a forest (Kleinberg et. al., SODA 2007)

Possible arrival model

- Adversarial:
  - No deterministic algorithm better than 0-competitive
  - Randomized algorithm
    - There is $\frac{1}{n}$ −randomized algorithm (for both expected cost and best candidate)
    - No randomized algorithm can do better than $\frac{1}{n}$ (Based on Yao's principle)
- Random arrival:
  - There is an algorithm that selects the maximum with probability $\frac{1}{e}$
- Non-uniform arrival:
  - We can still approach $\frac{1}{e}$ (Kleinberg et. al., STOC 2015)

Yao's principle

- Let A be a random variable with values in class of all deterministic algorithms $\mathcal{A}'$, and let X be a random variable with values in class of all instances $\mathcal{X}'$, and g as a gain function.
- Then:

$$\min_{x \in \mathcal{X}'} \mathrm{E}[g(A, x)] \leq \max_{a \in \mathcal{A}'} E[g(a, X)]$$

- Proof:

Technische Universität Berlin

Yao's principle

- Let A be a random variable with values in class of all deterministic algorithms $\mathcal{A}'$, and let X be a random variable with values in class of all instances $\mathcal{X}'$, and g as a gain function.
- Then:

$$\min_{x \in \mathcal{X}'} \mathrm{E}[g(A, x)] \leq \max_{a \in \mathcal{A}'} E[g(a, X)]$$

- Proof:
  - $E[g(a, X)] = \sum_{x \in \mathcal{X}'} P[X = x] g(a, x)$, $E[g(A, x)] = \sum_{a \in \mathcal{A}'} P[A = a] g(a, x)$
  - The weighted average is upper-bounded by its maximum value, and vice versa
    - $\min_{x \in \mathcal{X}'} \mathrm{E}[g(A, x)] \leq \sum_{x \in \mathcal{X}'} P[X = x] \sum_{a \in \mathcal{A}'} P[A = a]\, g(a, x)$
    - $\sum_{a \in \mathcal{A}'} P[A = a] \sum_{x \in \mathcal{X}'} P[X = x] g(a, x) \leq \max_{a \in \mathcal{A}'} E[g(a, X)]$

Choosing best candidate

- No randomized algorithm guarantees to select the best candidate with probability more than $\frac{1}{n}$.

Choosing best candidate

- No randomized algorithm guarantees to select the best candidate with probability more than $\frac{1}{n}$.
  - Define gain function as indicator random function based on selecting the best value
  - Based on Yao's principle, it is enough to show that there is a probability distribution over instances X such that $\max_{a\in\mathcal{A}'} E[g(a,X)] = \frac{1}{n}$
  - Fix an arbtriary algorithm a, and assume $x^{(t)} = (1,2,\ldots,t,0,\ldots,0)$
  - Let T be drawn uniformly at random from $1,\ldots,n$ and set $X = x^{(T)}$.

Choosing best candidate

- No randomized algorithm guarantees to select the best candidate with probability more than $\frac{1}{n}$.
  - Define gain function as indicator random function based on selecting the best value
  - Based on Yao's principle, it is enough to show that there is a probability distribution over instances X such that $\max_{a \in \mathcal{A}'} E[g(a, X)] = \frac{1}{n}$
  - Fix an arbtriary algorithm a, and assume $x^{(t)} = (1,2, \dots, t, 0, \dots, 0)$
  - Let T be drawn uniformly at random from $1, \dots, n$ and set $X = x^{(T)}$.
  - Consider an arbitrary deterministic algorithm a on sequence $x^{(n)} = (1,2 \dots, n)$, and selection s.
  - For another sequence:
    - if, $s \le t$, then the algorithm will make exactly the same decisions because sequences $x^{(t)}$ and $x^{(n)}$ look the same until position t.
    - If $s > t$, then the algorithm selects 0.

$$E[g(a, X)] = E[g(a, x^{(t)})] = \Pr(s = t) = \frac{1}{n}$$

Getting maximum expected value

- No randomized algorithm give us higher value than $\frac{1}{n}OPT$.

Getting maximum expected value

- No randomized algorithm give us higher value than $\frac{1}{n}OPT$.
  - Proof by contradiction, assume an algorithm with ratio $\frac{1}{n} + \epsilon$
  - Set $M = \frac{2}{\epsilon}$
  - Assume $x^{(t)} = (1, M, M^2, \dots, M^t, 0, \dots 0)$
  - Let v^* denote the maximum value given by this sequence, our algorithm either selects it or otherwise the cost is at most $\frac{v^*}{M}$
  - $v^*(\frac{1}{n} + \epsilon) \leq E\,[v(ALG)] \leq\ v^* \Pr[A\ selects\ maximum\ element] + \frac{v^*}{M}$
  - $\frac{1}{n} + \epsilon - \frac{\epsilon}{2} \leq \Pr[A\ selects\ maximum\ element]$ -> contradiction ☺

The old algorithm for random arrival model

- Algorithm: Say no to the first $\frac{n}{x}$ candidates, then select the one which has better value than the first $\frac{n}{x}$ candidates.

The old algorithm for random arrival model

- Algorithm: Say no to the first $\frac{n}{x}$ candidates, then select the one which has better value than the first $\frac{n}{x}$ candidates
  - Proof: Let us define probability p_i of the i-th candidate be in the first segment, and the best one is before the $(i-1)$ good ones in the second segment.

The old algorithm for random arrival model

- Algorithm: Say no to the first $\frac{n}{x}$ candidates, then select the one which has better value than the first $\frac{n}{x}$ candidates
  - Proof: Let us define probability p_i of the i-th candidate be in the first segment, and the best one is before the $(i-1)$ good ones in the second segment.

$$p_i = p[i\ in\ the\ first\ segment].$$
$$p[\ i-1\ in\ the\ second\ segment|i\ in\ the\ first\ segment].$$
$$p[i-2\ in\ the\ second\ segment|i\ in\ first, i-1\ in\ second\ segment]\ \ldots.$$
$$p[best\ candidate\ before\ i-2th, \ldots, 2nd\ best\ candidates]$$
$$= x \cdot \left(\prod_{j=0}^{i-2} (1-x) \cdot \frac{n}{n-j-1} - \frac{j}{n-j-1}\right) \cdot \frac{1}{i-1}$$
$$\lim_{n\to\infty} p_i = x. \left(\prod_{j=0}^{i-2} (1-x)\right) \cdot \frac{1}{i-1}$$
$$p_{success} = \sum_{i=2}^{\infty} p_i \ = \sum_{i=2}^{\infty} \frac{x}{i}(1-x)^i = -x ln(x)$$

Maximizes for $x\ = \frac{1}{e}$

Technische Universität Berlin

ML model for random arrivals

- Predicting the maximum value, g^*
- λ is the confidence of the predictions
- c describes to lose in the worst case
- $\exp\{W_{-1}(-1/(ce))\}$ and $\exp\{W_0(-1/(ce))\}$ are solution to $-x ln(x) = \frac{1}{ce}$

ALGORITHM 1: Value-maximization secretary algorithm

Input : Prediction p^* for (unknown) value $\max_i v_i$; confidence parameter $0 \leq \lambda \leq p^*$ and $c \geq 1$.

Output: Element a.

Set $v' = 0$.

Phase I:

for $i = 1, \ldots, \lfloor \exp\{W_{-1}(-1/(ce))\} \cdot n \rfloor$ **do**

  Set $v' = \max\{v', v_i\}$

end

Set $t = \max\{v', p^* - \lambda\}$.

Phase II:

for $i = \lfloor \exp\{W_{-1}(-1/(ce))\} \cdot n \rfloor + 1, \ldots, \lfloor \exp\{W_0(-1/(ce))\} \cdot n \rfloor$ **do**

  if $v_i > t$ **then**

    Select element a_i and STOP.

  end

end

Set $t = \max\{v_j : j \in \{1, \ldots, \lfloor \exp(W_0(-1/(ce))) \cdot n \rfloor\}\}$.

Phase III:

for $i = \lfloor \exp\{W_0(-1/(ce))\} \cdot n \rfloor + 1, \ldots, n$ **do**

  if $v_i > t$ **then**

    Select element a_i and STOP.

  end

end

ML model for random arrivals

- Theorem, Algorithm is $g_{c,\lambda}(\eta) - competive$, where $g_{c,\lambda}(\eta)$ is:

$$g_{c,\lambda}(\eta) = \left\{ \begin{array}{ll} \max\left\{\frac{1}{ce}, \left[f(c)\left(\max\left\{1 - \frac{\lambda+\eta}{OPT}, 0\right\}\right)\right]\right\} & if\ 0 \leq \eta < \lambda \\ \frac{1}{ce} & if\ \eta \geq \lambda \end{array} \right\}$$

And $f(c)$ is:

$$f(c) = \exp\{W_0(-1/(ce))\} - \exp\{W_{-1}(-1/(ce))\}.$$

Technische Universität Berlin

ML model for random arrivals

- Theorem, Algorithm is $g_{c,\lambda}(\eta) - competive$, where $g_{c,\lambda}(\eta)$ is:

$$g_{c,\lambda}(\eta) = \left\{ \begin{array}{ll} \max\left\{\frac{1}{ce}, \left[f(c)\left(\max\left\{1 - \frac{\lambda+\eta}{OPT}, 0\right\}\right)\right]\right\} & if\ 0 \leq \eta < \lambda \\ \frac{1}{ce} & if\ \eta \geq \lambda \end{array} \right\}$$

- Proof: In the worst-case we are always $\frac{1}{ce}$-competitive:
  - $p^* - \lambda > OPT$: Never goes to step two, similar to previous proof
  - $p^* - \lambda \leq OPT$: estimation was not higher than opt, then from the fact answer, and any α-approximation for the classical secretary problem yields an α-approximation for the value-maximization variant.

Technische Universität Berlin

ML model for random arrivals

- Theorem, Algorithm is $g_{c,\lambda}(\eta) - competive$, where $g_{c,\lambda}(\eta)$ is:

$$g_{c,\lambda}(\eta) = \left\{ \begin{array}{ll} \max\left\{\frac{1}{ce}, \left[f(c)\left(\max\left\{1 - \frac{\lambda+\eta}{OPT}, 0\right\}\right)\right]\right\} & if\ 0 \leq \eta < \lambda \\ \frac{1}{ce} & if\ \eta \geq \lambda \end{array} \right\}$$

- When the error is low, then:
- $p^* > OPT$: we have $g^* - \lambda < OPT$, Since OPT appears in Phase II with probability $f(c)$, we in particular pick some element in Phase II with value at least $OP\ T - \lambda$ with probability $f(c)$.
- With probability $f(c)$ we will pick some element with value at least $OP\ T - \lambda - \eta$. To see this, note that in the worst case we would have $g^* = OP\ T - \eta$, and we could select an element with value $g^* - \lambda$, which means that the value of the selected item is OP T - λ - η.

Technische Universität Berlin

Thank you ☺