Secretary Problem and its Extensions

Peter Borosan and Mudassir Shabbir

Rutgers University

May 4, 2009

Classical Secretary Problem

- There are n applicants arriving (online) one by one.
- After interviewing an applicant i, we can decide its value v_i relative to all $1 \dots i-1$.
- After interviewing applicant i, we have to decide whether to hire i or not.
- Select the best applicant.

Classical Secretary Problem

- There are n applicants arriving (online) one by one.
- After interviewing an applicant i, we can decide its value v_i relative to all $1 \ldots i-1$.
- After interviewing applicant i, we have to decide whether to hire i or not.
- Select the best applicant.

Algorithm

- Randomly choose some element j.

Classical Secretary Problem

- There are n applicants arriving (online) one by one.
- After interviewing an applicant i, we can decide its value v_i relative to all $1 \ldots i-1$.
- After interviewing applicant i, we have to decide whether to hire i or not.
- Select the best applicant.

Algorithm

- Randomly choose some element j.

Gives n-competitive ratio!

Classical Secretary Problem

- There are n applicants arriving (online) one by one.
- After interviewing an applicant i, we can decide its value v_i relative to all $1 \ldots i-1$.
- After interviewing applicant i, we have to decide whether to hire i or not.
- Select the best applicant.

Algorithm

- Randomly choose some element j.

Gives n-competitive ratio!

Can we do any better in adversarial model?

Random Permutation Model

- Values and ranks of applicants can be decided by adversary.
- But we know that order in which applicants arrive is uniformly random.

Random Permutation Model

- Values and ranks of applicants can be decided by adversary.
- But we know that order in which applicants arrive is uniformly random.

We can achieve n-competitive ratio by randomly picking, but we can also do better with little work. . .

$\frac{1}{3}$-competitive ratio Algorithm

- Interview first $\frac{n}{2}$ applicants and don't hire anyone.
- Let max_v be the value of best element in first half.
- Chose the first element in second half with $value > max_v$.

$\frac{1}{3}$-competitive ratio Algorithm

- Interview first $\frac{n}{2}$ applicants and don't hire anyone.
- Let max_v be the value of best element in first half.
- Chose the first element in second half with $value > max_v$.

Gives $\frac{1}{3}$-competitive ratio.

Optimal Algorithm

Algorithm

- Interview first k applicants and don't hire anyone.
- Let max_v be value of best element in first k.
- Select the first element after k elements with $value > max_v$.

Optimal Algorithm

Algorithm

- Interview first k applicants and don't hire anyone.
- Let max_v be value of best element in first k.
- Select the first element after k elements with $value > max_v$.

Optimal algorithm is obtained by putting $k = \frac{n}{e}$, with probability $\frac{1}{e}$!

Optimal Algorithm

Algorithm

- Interview first k applicants and don't hire anyone.
- Let max_v be value of best element in first k.
- Select the first element after k elements with $value > max_v$.

Optimal algorithm is obtained by putting $k = \frac{n}{e}$, with probability $\frac{1}{e}$! We can do a little better by putting $k = [(n - \frac{1}{2})e^{-1} + \frac{1}{2}]$ although the difference is very small[2].

Proof of Optimality

Proof.

$$Pr\{P\} = Pr\{A\} and Pr\{B\}$$

$$Pr\{P\} = \sum_{i=k+1}^{n} \frac{k}{n.(i-1)}$$

$$= \frac{k}{n} \sum_{i=k+1}^{n} \frac{1}{(i-1)}$$

$$= \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i}$$

$\ln k = (\ln n - 1) = \ln(n/e) \Rightarrow k = n/e$. Thus probability is maximized by setting $k = n/e$.[1] □

Motivations and Examples

- Hiring Secretaries.
- Dynamic auction market.
- Variation of House Selling Problem.
- The One-Armed Bandit Problem.[4]
- Maximizing the average and Bayes Sequential Statistical Decision Problems [Wald47].
- Detecting a change-point [Shiryaev63].
- The Burglar Problem [Haggstorm1966].

k-Secretary Problem and Different Objectives

- We have to select k secretaries instead of just 1.
- Random Permutation model.

k-Secretary Problem and Different Objectives

- We have to select k secretaries instead of just 1.
- Random Permutation model.

Different Objectives

- Maximize the probability of selecting best k.
- Minimize the expected sum of ranks of selected k.
- Maximize(or minimize) the expected sum of values of selected k.

Minimizing Expected Rank

Chow et al. showed in 1966 that optimal expected rank of selection is

$$\prod_{j=1}^{\infty}\left(\frac{j+2}{j}\right)^{\left(\frac{1}{j+1}\right)} \cong 3.8695$$

as $n \longrightarrow \infty$.

Now we have an algorithm that gives us optimal probability of selecting the best candidate, and an algorithm that minimizes the expected rank of selected candidate.

One question to be asked is:

Does there exist an algorithm that achieves both of these goals?

Trade-Off between Expected Rank and Probability

Theorem

Let p_0 be the maximum probability of selecting best object.
$\exists c > 0$, so that $\forall \epsilon > 0$, if A selects best object with $p_0 - \epsilon$, then expected rank of selected object is at least c/ϵ

Proof.

Assume contradiction that $\exists A$ that selects best object with $p_A \geq p_0 - \epsilon$, and expected rank of selected object is $< c/\epsilon$, then we can construct an algorithm R which selects the best object with $p_R > p_0$ by modifying A [2]. □

A Multiple-Choice Secretary Algorithm

- Select(n,k)
- If $k = 1$ then use classical algorithm.
- Otherwise
- $l = \lfloor \frac{k}{2} \rfloor$
- $m =$random sample from $B(n, 1/2)$
- $Select(m, l)$
- For $i = m + 1$ to n, Select ith item if its $value > y_l$ where y_l is lth largest value in first m unless we have already selected k elements.

A Multiple-Choice Secretary Algorithm

- Select(n,k)
- If $k = 1$ then use classical algorithm.
- Otherwise
- $l = \lfloor \frac{k}{2} \rfloor$
- $m =$random sample from $B(n, 1/2)$
- $Select(m, l)$
- For $i = m + 1$ to n, Select ith item if its $value > y_l$ where y_l is lth largest value in first m unless we have already selected k elements.

Gives competitiveratio of $1 - O(\sqrt{1/k})$[3]!

TH Cormen, CE Leiserson, RL Rivest, and C Stein.
Introduction to Algorithms.
MIT-press, 2001.

Ajtai M., Megiddo N., and Waarts O.
Improved algorithms and analysis for secretary problems and generalizations.
1995.

Kleinberg R.
A multiple-choice secretary algorithm with applications to online auctions.
2005.

Richard S. Sutton and Andrew G. Barto.
Reinforcement learning an introduction.
1998.