COMP6207
Algorithmic Game Theory

Lecture 16 Secretary Problem

Pavel Naumov
p.naumov@soton.ac.uk
Electronics and Computer Science
University of Southampton

Learning Outcomes

- By the end of this session, the students should be able to
  - ***Describe*** the secretary problem
  - ***Compare*** its online feature with previous offline problems
  - ***Compute*** the best strategy in the classical secretary problem

Offline vs. Online

- So far we have considered settings where all participants are present, the game is single-shot, and a centralised mechanism is in place

- So we have considered
  - offline setting (vs online setting)
  - single-shot game (vs repeated games)

- What if the participants arrive at different times (i.e. we are in an online setting)?

Another marriage problem

- When someone proposes to me, should I accept or reject the current proposer if
  - I know there are in total n men
  - I do not know the future proposers and their qualities
  - My current decision is irrevocable
- What is the best strategy to maximise the probability of finding the best partner?

A strategy

- Reject the first k candidates no matter how good they are.
  - Learn how good they are
  - Because there may be better ones later.
- After this, accept the first one who is better than all the first k candidates.
- If all the rest $n - k$ are worse than the best one among the first k, then accept the last one.

A Strategy (cont.)

- I want to determine, for each k, the probability that I appoint the best candidate.
- And then maximise this probability over all k.
- Suppose I accept candidate i ($i > k$).
- S: the event that I accept the best one (amongst all n candidates).
- S_i: the event that I accept the best one (amongst all n candidates), which is i.
- $$\mathbf{Pr}\left[S\right] = \sum_{i=k+1}^{n} \mathbf{Pr}\left[S_i\right].$$

A Strategy (cont.)

- S_i: the event that I accept the best one, which is i.
- Let's compute $\mathbf{Pr}[S_i]$
- S_i: event S_i is true implies that candidate i is the best among the n people
  - probability: $1/n$. (assuming uniform distribution in the classical problem)
- Recall that my strategy is to reject the first k candidates, event S_i is true also implies that candidates $k+1, \ldots, i-1$ are all worse than the best one among $1, \ldots, k$.
  - so that candidates $k+1, \ldots, i-1$ are all rejected.
  - probability: $k/(i-1)$. (*The best one among the first* $i-1$ *appears in the first* k.)
- So, $\mathbf{Pr}[S_i] = \frac{1}{n} \cdot \frac{k}{i-1}$

- $\mathbf{Pr}\left[S_i\right] = \frac{1}{n} \cdot \frac{k}{i-1} = \frac{k}{n(i-1)}.$

n-th harmonic number

$$Hn = \sum_{i=1}^{n} \frac{1}{i} \approx \ln(n)$$

- So $\mathbf{Pr}\left[S\right] = \sum_{i=k+1}^{n} \mathbf{Pr}\left[S_i\right]$

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

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

$$\approx \frac{k}{n} \cdot \left(\ln(n-1) - \ln(k-1)\right). \qquad n \to \infty$$

- Maximise this over all $k \in \{1, \ldots, n\}$ we get

$k \approx n/e \approx 0.368 \cdot n$

  – take derivative with respect to k, and set it equal to 0.

The Secretary Problem

- **Hire the best secretary**

n	1	2	3	4	5	6	7	8	9
k+1	1	1	2	2	3	3	3	4	4
Pr	1.0	0.5	0.5	0.433	0.433	0.428	0.414	0.410	0.406

  - For small n we can find the optimal k using dynamic programming
- **Selling a house**: Given a series of offers, how many should you consider before accepting an offer?
- **Job hunting**: How many interviews should I take?
- **Ski rental**: Continuing to pay a repeating cost or paying a one-time cost which eliminates or reduces the repeating cost?

Is our solution 100% applicable in real life?

In practice,

- Number of applicants ***n*** maybe unknown
- The quality of the candidates follow some distributions (learn from statistics)
- The quality of the first k candidates is correlated with the quality of the future candidates.
- Explainable AI
- ….

Online mechanism design problems

- Limited-Supply Online Auctions
  - Selling online advertisement slots
  - Selling online digital goods
    - Books
    - Movies
    - Music
  - …

- Online matching
  - Job-machine scheduling
  - Employer-employee matching (job market)
  - Request a ride on Uber
  - …