Online problems

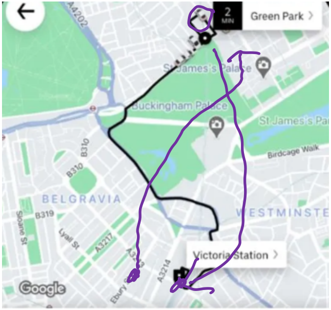
Online problems

It is a broad class of problems (you could say that living is an online problem) in which:

- The **Input** is revealed over time (not fully available initially).
- We must make decisions along the way.

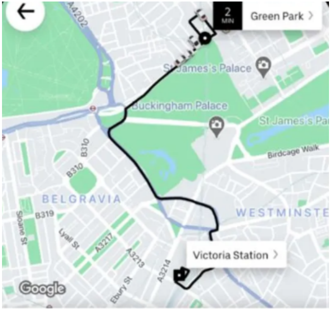
Example

Imagine you're an uber driver picking clients.



Example

Imagine you're an uber driver picking clients.



You don't know where future clients will appear, but you have to pick in real time.

How do we measure our performance?

How do we measure our performance?

$c \geq 1$

Given an online problem, a constant $c \leq 1$ and some objective function that we want to maximize, we say an online algorithm is *c*-competitive if

min

$$\mathbb{E}(\text{Output}) \geq c\text{OPT}$$

$\leq$

Where OPT denotes the maximum value that an oracle could get.

A concrete online problem: the Hiring Problem

Motivation: you interview several people for a position, when should you stop interviewing and hire someone?

Hiring Problem: Toy Instance

- Say you'll interview only 3 people;

Hiring Problem: Toy Instance

- Say you'll interview only 3 people;
- You know nothing about them until you interview them;

Hiring Problem: Toy Instance

- Say you'll interview only 3 people;
- You know nothing about them until you interview them;
- They arrive in a uniform random order;
- After an interview:
  - you can compare and rank seen candidates;
  - you must either hire the current candidate or call the next one;

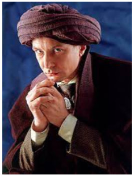
Hiring Problem: Toy Instance

- Say you'll interview only 3 people;
- You know nothing about them until you interview them;
- They arrive in a uniform random order;
- After an interview:
  - you can compare and rank seen candidates;
  - you must either hire the current candidate or call the next one;
- You want to maximize the probability of hiring the best.

3 Candidates Example

Maybe these are the 3 candidates in their arrival order:

Best

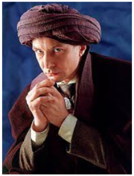


Worst

3 Candidates Example

Maybe these are the 3 candidates in their arrival order:


Best


Worst

So you should pick the first one, but...

Frame Title

You don't see the future and you have no standards to compare.

Frame Title

You don't see the future and you have no standards to compare.

Is there a strategy to pick the best with probability higher that $1/3$?

3 Candidates Example

There are 6 possible permutations:

1/2

3 Candidates Example

There are 6 possible permutations:

Strategy: always reject the first candidate and accept the second only if better than the first one.

Hiring Problem

- Now there are n candidates (n is known);
- You know nothing about them until you interview them;
- They arrive in a uniform random order;
- After an interview:
    - you can compare and rank seen candidates;
    - you must either hire the current candidate or call the next one;
- You want to maximize the probability of hiring the best.

Now what?

Can you think about a strategy? Will the probability of success converge to 0?

k

k cand.

Now what?

Can you think about a strategy? Will the probability of success converge to 0?

Consider the following strategy: reject first k candidates and then propose to the first candidate better than all previous ones.

Probability of success

Let A_i denote the event that the best candidate is the i^{th} candidate.

Total prob.

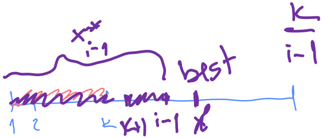
$$\mathbb{P}(\text{the best is chosen}) = \sum_{i=1}^{n} \underbrace{\mathbb{P}(\text{the best is chosen}|A_i)}_{W}\, \cancel{\mathbb{P}(A_i)}^{\,1/n}$$

for $i \le k$

$\mathbb{P}(W|A_i) = 0$

for $i > k$

$\mathbb{P}(W|A_i) = \frac{k}{i-1}$



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

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

$$\underbrace{H_{n-1} - H_{k-1}}_{} \approx \ln n - \ln k$$

$$\approx \frac{k}{n}\ln\frac{n}{k} = -\frac{k}{n}\ln\frac{k}{n}$$

Probability of success

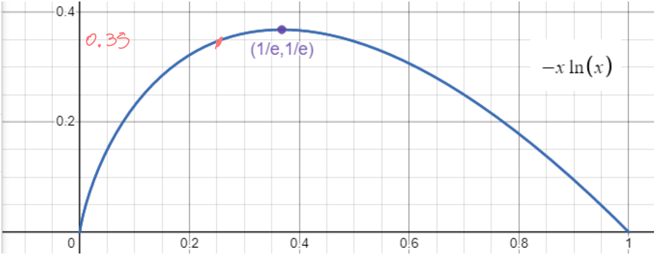
Define $x = \frac{k}{n}$, then $\mathbb{P}(\text{the best is chosen}) \approx x\ln(x^{-1}) = -x\ln(x)$.

Probability of success

Define $x = \frac{k}{n}$, then $\mathbb{P}(\text{the best is chosen}) \approx x\ln(x^{-1}) = -x\ln(x)$.
You can optimize x by taking derivative equal to 0.

Probability of success

Define $x = \frac{k}{n}$, then $\mathbb{P}(\text{the best is chosen}) \approx x \ln(x^{-1}) = -x \ln(x)$. You can optimize x by taking derivative equal to 0.



So the optimal choice is to take $k \approx n/e$ and this guarantees we choose the best candidate with probability $1/e$.

How good is this?

Our objective function was the probability of choosing the best and we got a probability of $\frac{1}{e}$.

How good is this?

Our objective function was the probability of choosing the best and we got a probability of $\frac{1}{e}$.

$$w(\text{output}) = \begin{cases} 1 & \text{if output is best} \\ 0 & \text{o/w} \end{cases}$$

An oracle on the other hand would always hire the best candidate (probability 1).

$$\mathbb{E}(w(\text{Output}))$$

Therefore, our online algorithm is $\frac{1}{e}$-competitive.

How good is this?

Theorem (Dynkin 1963)

In the Hiring Problem, the optimal strategy selects the best candidate with probability $\frac{1}{e}$.

Be careful what you wish for

In the hiring problem, there is $\frac{1}{e} \approx 37\%$ chance of not hiring anyone (why?).

Be careful what you wish for

If you reject the first 30% of the candidates (instead of a $\frac{1}{e}$ fraction), your chances of getting the best would drop to 36%, but the chances of no hiring would drop to 30%.

Try to always think about the practical and ethical consequences of your modeling choices.

Variants of the Problem

Prophet Inequalities

- Intuitively this is when you know something about the candidates beforehand (you read their CVs).
- Each candidate x_i has a value $v_i \sim X_i$, where v_i is revealed after the date, but the distributions $\{X_i\}_i$ are known beforehand.
- Objective: maximize the expected value of the chosen candidate.

Variants of the Problem

Pandora's Box

- Intuitively, here we add a cost on each interview.
- Like with prophet inequalities, the values of the candidates comes from known distributions.
- You can choose the order of the candidates, but each candidate has a price p_i you have to pay to interview them.
- You can choose any candidate x_i as long as you had payed the corresponding p_i.
- Objective: maximize expected total gain.

Variants of the Problem

You can also consider richer combinatorial constraints: instead of just choosing one candidate you may want to choose k or a feasible set of candidates given some feasibility constraints.