Optimization Under Uncertainty

Online Optimization (Lecture 1)

Scribe: Elliot Pickens (eep58)

Omar El Housni
ORIE 6360: Optimization Under Uncertainty

Mar. 22, 2022

1 Online Algorithms

Let's start off with the basic problem we're trying to solve with online algorithms.

- We are faced with a sequence of uncertain parameters (requests) $\sigma_1, \sigma_2, ..., \sigma_n, ...$
  - Which we assume to be fully adversarial i.e. each instance $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n, ...)$ is chosen in an adversarial manner
- We are made aware of the requests **one by one** and need to satisfy each request σ_i before serving σ_{i+1}
  - This differs from the situation faced by offline algorithms where the full sequence of arrivals is known **in advance**.

Obviously, we are at a disadvantage when working in an online setting, because we are at an informational disadvantage, but how wide is the gap?

Definition 1.1. We say that an algorithm is α-competitive if and only if $\max_\sigma \dfrac{Alg(\sigma)}{OPT(\sigma)} \leq \alpha$
Where

- $Alg(\sigma) :=$ the cost of the online algorithm
- $OPT(\sigma) :=$ the cost of the offline algorithm

1.1 The Ski Rental Problem

In the ski rental problem we know we're going to head off on a ski trip, and that we can either buy skis for some price B\$ or rent skis for 1\$ per day. We do not, however, know how many days we will actually spend skiing. Put compactly,

- The number of ski days is **unknown**
- All you know is that each morning you will be told if the trip will continue

In the **offline** setting this problem is fairly straight forward, because you know the number of days you'll be skiing in advance. All you need to do is ask whether the number of ski days is less than B. If $t > B$ you buy on day 1 otherwise you rent every day. Thus,

$$OPT(t) = \min(t, B) \quad \begin{array}{l} \text{if } t \leq B \text{ you rent every day} \\ \text{if } t > B \text{ you buy on day 1} \end{array}$$

The online setting is a little different. Since we don't know t we can't fully know our strategy on day 1. Instead let's consider an instance t (# of ski days). We can use the strategy: rent until day i and then purchase the skis on the morning of day i you you're told you'll be skiing that day.

The cost of this approach is

$$Alg_i(t) = \begin{cases} t, & \text{if } t < i \\ i - 1 + B, & \text{if } t \geq i \end{cases}$$

which means the online/online cost ratio is:

$$\frac{Alg_i(t)}{OPT(t)} = \frac{\mathbb{1}_{t<i} + (i-1+B)\mathbb{1}_{t\geq i}}{\min(t,B)}$$

and if $i = B$,

$$\frac{Alg_B(t)}{OPT(t)} = \frac{\mathbb{1}_{t<B} + (2B-1)\mathbb{1}_{t\geq B}}{\min(t,B)} = \begin{cases} 1, & \text{if } t < B \\ \dfrac{2B-1}{B}, & \text{if } t \geq B \end{cases}$$

__**Theorem:**__ the competitive ratio of Alg_B is $2 - \dfrac{1}{B}$ and this is the best you can achieve with a deterministic algorithm.

Let's show that $2 - \dfrac{1}{B}$ is a tight bound. Consider an online algorithm Alg_i for some $1 \leq i \leq B$. Let $t = i$ (e.g. you break your leg the day you buy the skis). Then

$$\begin{aligned} \frac{Alg_i(i)}{OPT(i)} = \frac{i-1+B}{\min(i,B)} &= \begin{cases} \dfrac{i-1+B}{i}, & \text{if } i \leq B \left(\dfrac{2B-1}{B}\right) \\ \dfrac{i-1+B}{B}, & \text{if } i > B \left(\dfrac{2B-1}{B}\right) \end{cases} \\ &\geq 2 - \frac{1}{B} \end{aligned}$$

so Alg_b is the best you can do.

Question: Can we do better by using randomized algorithms?

Maybe, depends on the problem - so let's check for the ski rental problem.

1.1.1 Randomized Algorithms

A randomized algorithm is a distribution over a deterministic algorithm. Much like the deterministic algorithms we've already described we can define the cost of a randomized algorithm θ as $\widetilde{Alg}(\sigma)$for an instance σ, and say that $\widetilde{Alg}$ is α-competitive if

$$\max_{\sigma} \mathbb{E}\left(\frac{\widetilde{Alg}(\sigma)}{OPT(\sigma)}\right) \leq \alpha$$

where we're taking expectation with respect to the algorithm and not the instance.

So what happens if we randomize the online algorithm we described in the last section (Alg_i where you buy on the morning of day i)? To do this we can build out a table representing how different strategies fare in different instances.

Competitive Ratio Table ($B = 4$)				
Algorithm	I_1	I_2	I_3	I_4
Alg_1	4	4/2	4/3	1
Alg_2	1	5/2	5/3	5/4
Alg_3	1	1	6/3	6/4
Alg_4	1	1	1	7/4

In the table we above we can see how each algorithm Alg_i performs for each instance I_i (where $I_i \coloneqq (t = i)$) when $B = 4$. It's clear that algorithm performance varies with each begin optimal under different conditions, so what if we tried to combine them to a (weighted) average result?

To do this let's assign a probability P_i to each Alg_i. Now we can find an average competitive ratio across all strategies for each instance. With the table above, for example, we get the following:

$$\begin{aligned}
4P_1 + P_2 + P_3 + P_4 &\leq \alpha, && \text{for } I_1\\
2P_1 + \frac{5}{2}P_2 + P_3 + P_4 &\leq \alpha, && \text{for } I_2\\
\frac{4}{3}P_1 + \frac{5}{3}P_2 + 2P_3 + P_4 &\leq \alpha, && \text{for } I_3\\
P_1 + \frac{5}{4}P_2 + \frac{3}{2}P_3 + \frac{7}{4}P_4 &\leq \alpha, && \text{for } I_4\\
P_1 + P_2 + P_3 + P_4 &= 1
\end{aligned}$$

Now, we just have to minimize α such that the system of equations above is satisfied (primarily by tuning the P_i values). Then

$$\Rightarrow \alpha \leq \frac{e}{e-1} \quad \text{for any } B$$

and

$$\frac{1}{1 - (1 - 1/B)^B} \leq \frac{1}{1 - 1/p} = \frac{e}{e-1}$$

2 Conclusion

In this lecture we covered the basics on online algorithms and explored some of the difficulties we face when working in the online setting. We also touched upon how randomized algorithms might be able to help us improve performance.

In the next lecture we will continue with online algorithms by looking at online matching.