Online Algorithms and the Ski Rental Problem

1. Introduction to Online Algorithms

Online algorithms deal with the problem of decision making under partial information, where the inputs arrive as a stream of data, and at each time step, decision must be made on the fly, without knowing what will happen in the future. Examples of scenarios where online algorithms apply includes finding the shortest path for partially observable graphs (Canadian Traveler Problem), hiring high-quality applicants from interviews where individual decisions must be made right after the interview (Secretary Problem), and determining whether to rent or buy a particular resource at each time point (Ski Rental Problem).

We are interested in algorithms that guarantee a certain performance, or produce solution that is within a certain bounded ration away from the optimal solution even in the worst case. This motivates the definition of competitive ratio.

Definition 1: *The* competitive ratio *of an online algorithm A is the worst case over possible future input sequences μ of the ratio A(μ)/A*(μ), where A(μ) represents the cost of A on μ and A*(μ) is the cost of the optimal offline algorithm.*

2. The Ski Rental Problem Overview

Consider the following problem in online decision making. You came to a resort for skiing, uncertain about how many days you will ski. For each day in an ambiguous time period, you have to make decisions between renting or buying skis. Let's say that you can rent the skis at a cost of $1 per day, or buy them for $B. Once you buy the skis, you can ski rent-free for as long as you decide to stay, but before then you incur the daily rent cost. You would like to determine the "optimum" day on which you should buy skis.

We can formulate this into an online problem thus:

- Let x_d denote the input sequence where you ski for d days, and then stop
- Let A_i denote the deterministic algorithm where you rent the skis for i-1 days before buying the skis on the i-th day

Determining the optimum solution can be viewed as a game between you (the consumer) and an adversary that we will call nature who's able to determine when you would stop skiing.

- Your action set consists of $\pi_c = \{A_i : i \in [0, +\infty) \cup \{+\infty\}\}$ where $i = +\infty$ means you will always rent.
- Nature's action set consists of $\pi_n = \{ x_d : d \in [0, +\infty) \cup \{+\infty\}\}$.

You first choose an algorithm A_i, then nature will choose the worst possible x_d based on your A_i so as to maximize $A(x_d)/A^*(x_d)$.

Assume that nature sets the time you ski to be d days (x_d) and you choose to buy the skis on day i (A_i), then the cost of A_i for different i's is as follows:

$$C(A) = \begin{cases} B + i - 1 & \text{if } d \geq i \\ d & \text{if } d < i \end{cases}$$

While the cost of the optimal offline algorithm (A*), with knowledge of d, is

$$C(A^*) = \min\{B, d\}$$

Thus the optimal algorithm chooses either to rent the whole time or buy from the beginning.

We want to minimize the competitive ratio ρ:

$$\text{competitive ratio } \rho = \max_{over\ d} \frac{C(A)}{C(A*)}$$

One strategy we could choose is the "better-late-than-never" strategy, where we rent until we realize we should've bought, then buy. This means renting the ski for B days, then buy. Since nature wants to maximize the competitive ratio, it will choose to end to skiing time right after you buy skis. So:

- If $i \leq B$, then $\rho = \frac{i + B - 1}{i}$
- If $i > B$, then $\rho = \frac{i + B - 1}{B}$

We need to pick i to minimize ρ. This minimum is achieved when i = B, which gives us a competitive ratio of $\rho = 2 - \frac{1}{B}$.

To generalize this result, assume renting the ski costs \$r per day, and buying the ski costs \$b, then:

Theorem 2: *The algorithm better-late-than-never has the best possible competitive ratio of* $2\text{-}\frac{r}{p}$ *out of all the deterministic algorithms for the ski-rental problem.*

3. Randomized Algorithms for Ski Rental

Now we consider the extension where our action space contains probabilistic strategies and we make decisions based on coin tosses. Now, the adversary nature only knows our strategy but not the exact output. Now a strategy in our action space can be buying skis on day i with P(i), where $\sum_{i=0}^{\infty} P(i) = 1$. Then, if the time of our ski is d, the cost of the optimal offline solution is still just C(P*) = min {B, d}. Now for our strategy P(i), if we buy skis on day i, then we will need to pay \$(i – 1 + B) if i < d and \$d if i ≥ d. So the cost for our online algorithm becomes:

$$C(P(i)) = d \sum_{i \geq k} P(i) + \sum_{i<d} (i + B - 1) P(i) \approx d \int_d^{\infty} P(i) di + \int_0^d (i + B - 1)\, P(i) di$$

Our goal is:

Minimize ρ = C(P(i)) / C(P*) such that

$$\int_0^{\infty} P(i) di = 1$$

We have that $d \int_d^{\infty} P(i) di + \int_0^d (i + B - 1)\, P(i) di = \rho \min\{B, d\}$

- Case d ≥ B: Differentiate with respect to d on both sides (applying chain rule on the left), we have

$$-dP(d) + \int_d^{\infty} P(i) di + (d + B - 1) P(d) = 0$$

$$(B - 1) P(d) + \int_d^{\infty} P(i) di = 0$$

We can assume that B > 1 since otherwise it would just make sense to always buy before day 1, which is trivial. Under this assumption, we have B – 1 > 0 and P(i) > 0 for values of i ≥ 0. Thus,

- Case d < B: Case d ≥ B: Differentiate with respect to d on both sides (applying chain rule on the left), we have

$$-dP(d) + \int_d^{\infty} P(i) di + (d + B - 1) P(d) = \rho$$

$$(B - 1) P(d) + \int_d^{\infty} P(i) di = \rho$$

Differentiating again gives:

$$B \frac{dP(d)}{dd} - P(d) = 0$$

$$P(d) = Ce^{\frac{d}{B}}, for\ some\ constant\ C$$

To solve for C, we plug $P(d) = Ce^{\frac{d}{B}}$ into $\int_0^\infty P(i)di = 1$:

$$\int_0^\infty P(i)di = \int_0^B Ce^{\frac{d}{B}}dd = 1$$

$$C = \frac{1}{B(e-1)}$$

Thus

$$\rho = \frac{\mathrm{e}}{\mathrm{e}-1}$$

4. Multi-Shop Ski Rental Problem

Now we generalize the ski rental problem and consider the case when there are multiple shops offering rental and purchase options for skis at difference prices. Now, we (the customer) must make decisions for not only where, but also when to buy skis. We want to find the optimum online mixed strategy from the customer's perspective. For this section, we restrict our analysis to the multi-shop ski rental problem (MSR), where the customer must choose one shop immediately after arrival at the ski resort and cannot change the shop once she chooses one.

Assume there are n shops Again, we model our problem as a zero-sum game between the customer and nature.