Prophet Inequalities Over Time

BTP Report by Divyanshu Agarwal : 2020CS10343

January 4, 2024

Overview

1 Problem statement

1.1 Informal description

Suppose you have to rent a room for n days. On each of the n days, a buyer comes, offers you a price that he/she is willing to pay per day. You need to decide the number of days that you want to rent the room to the buyer. The buyer will take the room for as many days as you offer him. You cannot rent the room to buyer i if the room has already been rented on day i to some previous buyer. The distribution of the prices offered by the buyer is known. We need to construct the optimal strategy to rent the room and compare it against an adversary that already knows the values that the buyers are going to offer (the prophet).
Specifically, we are dealing with the scenario where the buyers are *iid* (indpendent and identically distributed).

1.2 Formal statement

Model the value of the i^{th} person as a random variable X_i. X_i's are assumed to be iid variables with the distribution F. The items will be observed in the order : $x_n, x_{n-1}, \ldots, x_1$. For each x_i decide the number of days d_i to choose it. The value obtained by the algorithm is thus given by :

$$ALG = \sum_{i=n}^{1}(x_i \cdot d_i) \tag{1}$$

Since we are not allowed to rent a room on day for which it has already been rented, we have the constraint that if $d_i = k$, then $d_{i-1}, d_{i-2} \ldots, d_{i-k} = 0$.
Let the payoff obtained by the prophet be denoted by OPT. Then the competitive ratio is defined as follows :

$$\text{Competitive Ratio} = \frac{\mathbb{E}[ALG]}{\mathbb{E}[OPT]} \tag{2}$$

The expectation is over the random variables $X_1, X_2, \ldots, X_n$ and over the algorithm (if it is randomised). Our goal is to come up with an algorithm that maximises the competitive ratio, over the worst case n and F.
Also note that we only deal with distributions with positive support, therefore $F(x) = 0 \ \forall x < 0$. This also allows us to use 33 to express the expected value of the random variables.

2 Optimal strategy for the prophet

Since the prophet knows the values in advance, he rents a room to a buyer till the day when a buyer with a better values arrives. This means on day i, the rent earned by the prophet is

$\max(x_n, x_{n-1}, \ldots, x_i)$ The payoff obtained by the prophet can be expressed as follows :

$$OPT = \sum_{i=n}^{i=1} \max(x_n, x_{n-1}, \ldots, x_i) \tag{3}$$

$$\mathbb{E}[OPT] = \sum_{i=1}^{i=n} \mathbb{E}[\max(x_n, x_{n-1}, \ldots, x_i)] \tag{4}$$

Since the max of n random variables with distributions F is distributed according to F^n

$$\begin{aligned}\mathbb{E}[OPT] &= \sum_{i=1}^{i=n} \int_{x=0}^{\infty} (1 - F^i(x))dx \\ &= \int_{x=0}^{\infty} \left(n - \sum_{i=1}^{n}(F^i(x))\right) dx \\ &= \int_{x=0}^{\infty} \left(n - F(x) \cdot \frac{(1 - F^n(x))}{1 - F(x)}\right) dx \end{aligned} \tag{5}$$

3 Previous Work

This problem was first introduced in this paper by Abels Et al. [1]. They gave the following important results :

1. The optimal dynamic programing algorithm
2. An upper bound on the competitive ratio of $1/\phi \approx 0.618$
3. A quantile-based algorithm, along with a lower bound of 0.598 on the *CR* of this algorithm

1 and 3 were in particular interest to us.
The dynamic programming algorithm uses distinct threshold for each day defined as follows : $\tau_0 = 0$, and $\tau_i = \frac{\mathbb{E}[\text{Payoff of algorithm from } i \text{ to } 1]}{i}$. Note that the thresholds represent the average daily payoff of the algorithm from that time step. The online decision process works as follows :

Algorithm 1: Optimal Online Decision Procedure

for $i = n, \ldots, 1$ **do**
    if $x_i >= \tau_{i-1}$ **then**
        Select x_i for all remaining i time steps ;
        Break;
    else
        Select x_i only in step i

Abels Et al. proved that the τ_i's defined in the above fashion form an increasing sequence, and hence established the optimality of the algorithm. They, however, were unable to derive a lower bound for this algorithm, nor were they able to obtain a closed form of the thersholds to work with.

4 Determining if $n \to \infty$ is the worst case

In the single choice iid prophet problem it is observed that $n \to \infty$ is the worst case, where we simulate an algorithm on n items using an algorithm on *2n* items and observing that the competitive ratio improves on n as compared to *2n*. We begin by checking if a similar guarantee holds in our scenario as well.

Notation: For $\mu \in \mathbb{R}^+$, let F_μ be the CDF of the minimum of iid random variables, each with CDF $\sqrt{F}$, conditioned on their max being μ .
Suppose we have an Algorithm : ALG_{2n} that guarantees CR_{2n} on 2n items. We assume that the algorithm behaves by either picking the items for 1 day or committing to an item for the remaining $n-i+1$ days. We will use this to construct an algorithm : ALG_n that for an instance of n random variables $X_1, X_2, , \ldots, X_n$. Say the distribution of X_i is F. Our algorithm works as follows :

Algorithm 2: ALG_n

for $i = n, \ldots, 1$ **do**
    Read X_i ;
    Sample Z_i from F_{X_i} ;
    Randomly Permute (X_i, Z_i) and call them (Y_{2i}, Y_{2i-1}) ;
    Send Y_{2i} followed by Y_{2i-1} to ALG_{2n} ;
    if ALG_{2n} *commits to* Y_{2i} *or* Y_{2i-1} **then**
        Select X_i for all i remaining steps ;
        Break ;
    else
        Select X_i for one step

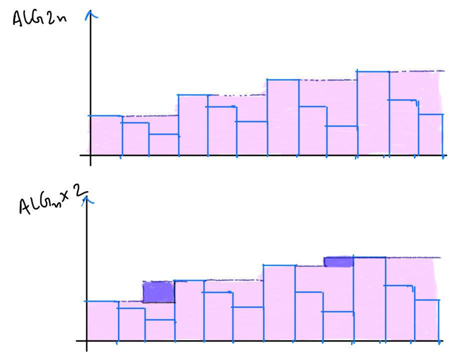
We claim that following this procedure we would have $2 * ALG_n \geqslant ALG_{2n}$.
Proof : Let u_i represents the value chosen by ALG_n on day i and v_i represents the value chosen by ALG_{2n} on day i. For any given i we can say that $u_i = max_{(}v_{2i}, v_{2i-1})$. Therefore $2*u_i \geqslant v_{2i}+v_{2i-1}$. Summing over i on both sides , we can obtain :

$$2 * ALG_n \geqslant ALG_{2n}$$

or,

$$\frac{ALG_n}{ALG_{2n}} \geqslant \frac{1}{2} \tag{6}$$

Now let us compare the payoffs of the OPT. On day i, OPT picks the prefix max. The diagram below shows the run of the Algorithm :2.



The height of the blue bars represent the realisation of the random variable. The purple dotted line records the prefix max. The shaded areas represent the payoff ALG_{2n} and $2 * ALG_n$. The purple shaded regions are where they differ. It is clear that area of the shaded purple region is $\leqslant \max(X_1, X_2, \dots, X_n)$. We can also see that these differ only when $Y_{2i} \geqslant Y_{2i-1}$ and Y_{2i} is a prefix maximum. In expectation the prefix maximum is expected to change at even-indexed samples only half of the times. Hence in expectation the area of the shaded region is $\frac{1}{2}\mathbb{E}[\max_i(X_i)]$. We can write this as following :

$$2 * \mathbb{E}[OPT_n] - \mathbb{E}[OPT_{2n}] = \frac{1}{2}\mathbb{E}[\max_i(X_i)]$$

Also observe that $\frac{n}{2}\mathbb{E}[\max_i(X_i)] \leqslant \mathbb{E}[OPT_n]$, since in expectation $\max_i(X_i)$ is conuted $\frac{n}{2}$ times in OPT_n.
So we obtain the following :

$$\begin{aligned} 2 * \mathbb{E}[OPT_n] - \mathbb{E}[OPT_{2n}] &\leqslant \frac{1}{n}\mathbb{E}[OPT_n] \\ \implies \frac{\mathbb{E}[OPT_{2n}]}{\mathbb{E}[OPT_n]} &\geqslant 2 \cdot (1 - \frac{1}{2n}) \end{aligned} \tag{7}$$

Multiplying 6 and 7 we can obtain that :

$$\frac{CR_n}{CR_{2n}} \geqslant 1 - \frac{1}{2n} \tag{8}$$

Applying repeatedly, we get the following :

$$\begin{aligned} CR_n &\geqslant CR_{2n}(1 - \frac{1}{2n}) \geqslant CR_{4n}(1 - \frac{1}{2n}) \cdot (1 - \frac{1}{4n}) \\ &\geqslant CR^*(1 - \frac{1}{2n}) \cdot (1 - \frac{1}{4n}) \cdot (1 - \frac{1}{8n}) \dots \\ &\geqslant CR^*(1 - \frac{1}{2n} - \frac{1}{4n} - \frac{1}{8n} - \dots) \\ &= CR^*(1 - \frac{1}{n}) \end{aligned} \tag{9}$$

Where $CR^* = \liminf_{m \to \infty} CR_m$
Hence we establish that, $CR_n \geqslant CR^*(1 - \frac{1}{n})$ Although this does not imply that $n \to \infty$ is the worst

case, it does motivate us to consider the case, since this ensures that the worst case n needs to be a large value.

5 Analysing $n \to \infty$

Consider the thresholds described in 4. Let the distribution of the maximum of the random variables be F. Therefore the distribution of each random variable is $F^{1/n}$. The value of the thresholds can be expressed as follows :

$$\begin{aligned}
\tau(i+1) &= \frac{1}{i+1} \cdot [\mathbb{P}[X < \tau(i)] \cdot (\mathbb{E}[X|X < \tau(i)] + i\tau(i)) + \mathbb{P}[X \geqslant \tau(i)] \cdot ((i+1)\mathbb{E}[X|X \geqslant \tau(i)])] \\
&= \frac{1}{i+1} \cdot [\mathbb{P}[X < \tau(i)] \cdot (\mathbb{E}[X|X < \tau(i)]) + \mathbb{P}[X \geqslant \tau(i)] \cdot (\mathbb{E}[X|X \geqslant \tau(i)]) \\
&\quad + i \cdot (\mathbb{P}[X < \tau(i)] \cdot \tau(i) + \mathbb{P}[X \geqslant \tau(i)] \cdot \mathbb{E}[X|X > \tau(i)])] \\
&= \frac{1}{i+1} \cdot [\mathbb{E}[X] + i \cdot (\mathbb{P}[X < \tau(i)] \cdot \tau(i) + \mathbb{P}[X \geqslant \tau(i)] \cdot \mathbb{E}[X|X > \tau(i)])]
\end{aligned} \tag{10}$$

Keep in mind that the $\mathbb{E}[X]$ term above comes from picking the item on day i, it counts the contribution of a) picking x_i when we don't commit and b) contribution of x_i on day i when we commit

$$= \frac{1}{i+1} \cdot \left[\mathbb{E}[X] + i \cdot \left(\tau(i) + \int_{x=\tau(i)}^{\infty} (1 - F^{1/n}(x))dx \right) \right] \tag{11}$$

Multiplying by $(i+1)$ and subtracting $i \cdot \tau(i)$, we get

$$(i+1) \cdot \tau(i+1) - (i) \cdot \tau(i) = \mathbb{E}[X] + i \cdot \left(\int_{x=\tau(i)}^{\infty} (1 - F^{1/n}(x))dx \right)$$

For a large n, let t = i/n, substituing above we get :

$$(nt+1) \cdot \tau(nt+1) - (nt) \cdot \tau(nt) = \mathbb{E}[X] + nt \cdot \left(\int_{x=\tau(nt)}^{\infty} (1 - F^{1/n}(x))dx \right)$$

dividing by n on both sides, we get :

$$(t + \frac{1}{n}) \cdot \tau(n(t + \frac{1}{n})) - t \cdot \tau(nt) = \frac{\mathbb{E}[X]}{n} + t \cdot \left(\int_{x=\tau(nt)}^{\infty} (1 - F^{1/n}(x))dx \right)$$

Also note that, as n $\rightarrow \infty$:

$$1 - F^{1/n}(x) = \frac{1}{n} \cdot \ln\left(\frac{1}{F(x)}\right)$$

Let 1/n = dt

$$(t+dt)\cdot\tau(n(t+dt)) - t\cdot\tau(nt) = \mathbb{E}[X]dt + t\cdot\left(\int\limits_{x=\tau(nt)}^{\infty} \ln\left(\frac{1}{F(x)}\right) dtdx\right)$$

$$\frac{(t+dt)\cdot\tau(n(t+dt)) - t\cdot\tau(nt)}{dt} = \mathbb{E}[X] + t\cdot\left(\int\limits_{x=\tau(nt)}^{\infty} \ln\left(\frac{1}{F(x)}\right) dx\right)$$

We overload notation and let $\tau(t)$ denote $\tau(\lfloor nt \rfloor)$ for $0 \leqslant t \leqslant 1$

$$\frac{(t+dt)\cdot\tau(t+dt) - t\cdot\tau(t)}{dt} = \mathbb{E}[X] + t\cdot\left(\int\limits_{x=\tau(t)}^{\infty} \ln\left(\frac{1}{F(x)}\right) dx\right)$$

$$\frac{\mathrm{d}}{\mathrm{dt}}(t\cdot\tau(t)) = \mathbb{E}[X] + t\cdot\left(\int\limits_{x=\tau(t)}^{\infty} \ln\left(\frac{1}{F(x)}\right) dx\right)$$

Note that the expected value of X given by

$$\mathbb{E}[X] = \int\limits_{x=0}^{\infty} (1 - F^{1/n}(x))dx$$

vanishes when $n \rightarrow \infty$.

As discussed in section 5, this means that the contribution of picking the item for a single day is negligible when $n \rightarrow \infty$

Finally, we obtain :

$$t\cdot\tau'(t) + \tau(t) = t\cdot\left(\int\limits_{x=\tau(t)}^{\infty} \ln\left(\frac{1}{F(x)}\right) dx\right). \tag{12}$$

We can further differentiate, the above equation with respect to t, in order to get rid of the integral to obtain :

$$F(\tau(t)) = \exp\left(\frac{t^2\cdot\tau''(t) + t\cdot\tau'(t) - \tau(t)}{t^2\cdot\tau'(t)}\right) \tag{13}$$

We can use 5 to express the payoff of the adversary, note that we need to $F^{1/n}$ instead of F in 5 since the distribution of each item is given by $F^{1/n}$.

$$\begin{aligned}\mathbb{E}[OPT] &= \int\limits_{x=0}^{\infty} \left(n - F^{1/n}(x) \cdot \frac{(1-F(x))}{1-F^{1/n}(x)} \right) dx \\ &= \int\limits_{x=0}^{\infty} \left(n - \frac{(1-F(x))}{1/n \cdot \ln(\frac{1}{F(x)})} \right) dx \\ &= n \cdot \int\limits_{x=0}^{\infty} \left(1 - \frac{(1-F(x))}{\ln(\frac{1}{F(x)})} \right) dx\end{aligned}$$

The competitive ratio can be expressed as follows :

$$\begin{aligned}\alpha &= \frac{\mathbb{E}[ALG]}{\mathbb{E}[OPT]} = \frac{n \cdot \tau(1)}{n \cdot \int\limits_{x=0}^{\infty} \left(1 - \frac{(1-F(x))}{\ln(\frac{1}{F(x)})} \right) dx} \\ \frac{1}{\alpha} &= \frac{\cdot \int\limits_{x=0}^{\infty} \left(1 - \frac{(1-F(x))}{\ln(\frac{1}{F(x)})} \right) dx}{\tau(1)}\end{aligned}$$

substituting x by $\tau(t)$ in the numerator we obtain :

$$\frac{1}{\alpha} = \frac{\cdot \int\limits_{t=0}^{\infty} \left(1 - \frac{(1-F(\tau(t)))}{\ln(\frac{1}{F(\tau(t))})} \right) \tau'(t) dt}{\tau(1)} \tag{14}$$

Using 13, we can express $1/\alpha$ as a function of $\tau(t)$ and t alone.
Using 12 we know that an F determines a unique $\tau(t)$, and a valid $\tau(t)$ corresponds to a unique CDF F. Therefore in order to find the minimum α, we can maximise the RHS over the space of valid $\tau(t)$'s. This is functional extremisation and methods such as calculus of variation can be applied. We will obtain the worst case $\tau(t)$ corresponding to the worst case F and its competitive ratio α. I attempted that but the expressions turned out to be very complicated.

6 Trying to construct the worst case F

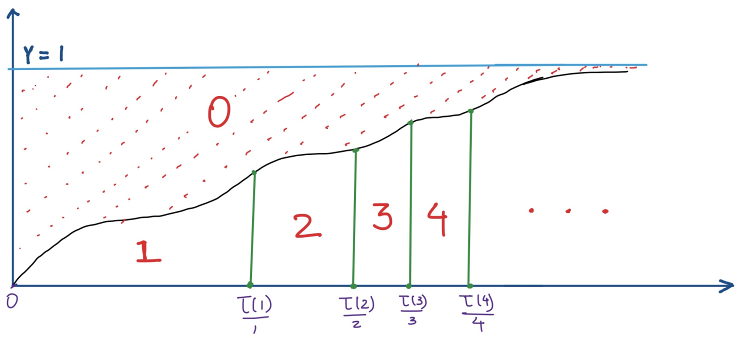
I tried another approach in which I tried to construct the worst-case distribution for a fixed n. Note the slight change in notation for $\tau(n)$ below

- Let $\tau(n)$ denote the Algorithm's payoff for n days
- Abels etal. obtained this recursive relation :

$$\tau(n) = n \cdot \mathbb{E}[x] + (n-1) \cdot \left(\frac{\tau(n-1)}{n-1} - \mathbb{E}\left[x | x \leqslant \frac{\tau(n-1)}{n-1} \right] \right)$$

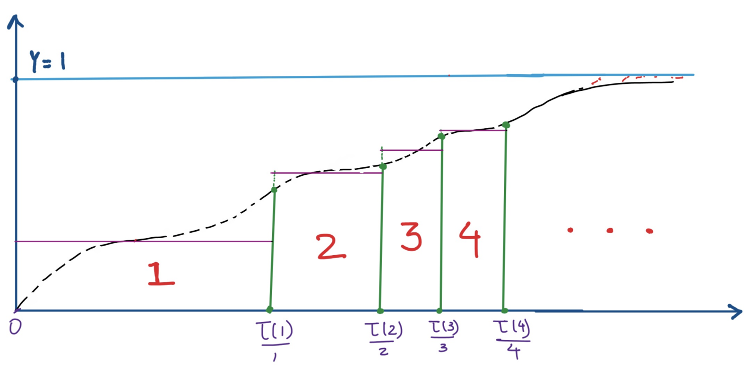
- Takeaway from this expression is that $\tau(n)$ can be recursively expressed in terms of $\tau(n-1)$

$$\tau(n) = n \cdot \mathbb{E}[x] + (n-1) \cdot \left(\frac{\tau(n-1)}{n-1} - \mathbb{E}\left[x | x \leqslant \frac{\tau(n-1)}{n-1} \right] \right)$$



- I modify the distribution F such that $\tau(1), \tau(2), \ldots, \tau(n-1)$ and the expectation quantities remain the same

- Make the distribution flat in the regions between $\frac{\tau(i)}{i}$ and $\frac{\tau(i+1)}{i+1}$. We do this because the prophet's payoff (which we want to maximise) is a concave function of F, and hence the maxima occurs when all the values are equal.



- Fixing the $\frac{\tau(i)}{i}$'s , this is the best case for the prophet

- I obtained the following relation for the height of the $i-th$ interval, denoted by h_i (the height from $\tau(i-1)$ to $\tau(i)$) :

$$h_i = \frac{\frac{i+1}{i}(\tau(i+1) - \tau(1)) - \frac{i}{i-1}(\tau(i) - \tau(1))}{\tau(i) - \tau(i-1)} \tag{15}$$

- Further applying constraints such as h_i's must be non decreasing, I could further obtain conditions on $\tau(i)$ such as $\tau(i) - \tau(i-1)$ decreases with i, and $\frac{\tau(u)-\tau(i-1)}{\tau(i-1)-\tau(i-2)}$ increase with i.
- We can see that, h_i can be in principle expressed in terms of $\tau(1)$.
- Thus we can express the payoff of the ALG and the OPT in terms of $\tau(1)$, which would be linear in $\tau(1)$, and thus obtain a relation dependent only on n, which would indicate the worst case competitive ratio for the given n.
- However, explicitly solving the recursion for h_i was hard. I realised this approach is just a discrete version of the calculus of variations approach discussed in the previous section and tried something else.

7 Review of Abels Et al.'s algorithm

In their paper Abels Et al. present the following algorithm (Note that the change of observation order from $i = n, \ldots, 1$ to $i = 1, \ldots, n$.
The algorithm's threshold in step $i = 1, \ldots, n$ is the quantile $\delta_{p(i)} = F^{-1}(p(i))$, where p(i) denotes some probability. We let $p(0) = 1, p(n+1) = p(n+2) = 0$.

Algorithm 3: ONL

```
for i = 1, ..., n do
    if  x_i >= δ_p(i) then
        Select x_i for all n-i+1 remaining steps ;
        Break;
    else
        Select x_i random variable for one step
```

They further prove that the algorithm ONL with the threshold $\delta_{p(i)}$ where $p(i) = \frac{-ci}{n^2}$ and $c = 9.71$ gives a lower bound of 0.598 for $n \to \infty$.
To prove the above they establish the following :

$$\frac{\mathbb{E}[ONL_n]}{\mathbb{E}[OPT_n]} \geqslant \frac{\sum_{k=0}^{n} \alpha_k \mathbb{E}[x|\delta_{p(k+1)} < x \leqslant \delta_{p(k)}]}{\sum_{k=0}^{n} \alpha_k^* \mathbb{E}[x|\delta_{p(k+1)} < x \leqslant \delta_{p(k)}]} \geqslant \min_{s \in \{0,1,\ldots,n\}} \frac{\sum_{k=0}^{s} \alpha_k}{\sum_{k=0}^{s} \alpha_k^*} = CR \tag{16}$$

Where they express α_k and α_k^* in terms of $\mathbf{p}(\cdot)$ and $\mathbf{n}$ alone, independent of F. We realised that this is essentially an ASD guarantee that there algorithm is providing using the prefix sum of the coefficents α_k and α_k^*. So we explore ASDness in this regard further

8 Approximate Stochastic Dominance is equivalent to Competitiveness

Suppose we have an algorithm that guarantees the following:

$$\mathbb{E}[\text{time ALG chooses} \geqslant \text{x}] \geqslant c \cdot \mathbb{E}[\text{ time OPT chooses} \geqslant \text{x}] \qquad \forall x \geqslant 0 \tag{17}$$

We will call that this algorithm is a c-ASD algorithm. Note that the term ASD is usually used to compare distributions and has a different (but similar) definition. Whenever I use the term ASD in this text, it would mean as what is defined in eq. (17). We will prove the following :

1. A c-ASD algorithm is c - competitive
2. If there exists a c competitve algorithm then there also exists a c-ASD algorithm.

Look at the appendix B.1 and appendix B.2 for a proof of the above statements.

8.1 Benefit from ASDness

Once we have established that a c-ASD algorithm leads to a c-compettive algorithm and vice versa. This simplifies our process for constructing an algorithm. In particular we do not need to look for a worst case distribution. Instead we can work with any distribution of our choice, if we obtain an algorithm ALG that is c-competitive on F, then we can construct an algorithm ALG' for any other distribution F' , such that the thresholds used for decision making by ALG' satisfy the following property : If for the i-th item, the threshold used by ALG is τ, then ALG' uses the threshold τ' such that $F'(\tau') = F(\tau)$. We are dealing only with continuous distributions, hence jumps across $F(\tau)$ won't bother us. Although we can handle them by randomising the algorithm at jump points.

9 Quantile based algorithm

For $n \to \infty$ we construct an algorithm for any general distribution F, that takes decisions based on the quantile of the distribution the element lies in. We observed the i-th item at the time instant $t(i) = \frac{i}{n}$.Note that w view items in the order n to 1. We use a thresholding function $p(t)$, such that $p : [0, 1] \to [0, 1]$ The algorithm works as follows :

Algorithm 4: Optimal Online Decision Procedure

for $i = n, \ldots, 1$ **do**
  if $F(x_i) >= p(i/n)$ **then**
    Select x_i for the remaining $\frac{i}{n}$ time ;
    Break;
  else
    Do not pick x_i;

When $n \to \infty$, this translates to viewing the $\lfloor nt \rfloor^{\text{th}}$ element at time t, for $0 \leqslant t \leqslant 1$. We do not pick an item for duration $1/n$ in case it is below the threshold since it translates to picking

the item for an infintesimal duration which does not affect our ASDness guarantee, and hence the competitive ratio.
What remains now is to find the optimal function $p(t)$ that maximises the ASD guarantee.

9.1 Analysis of the OPT

We will calculate the expected time that the optimal holds on to items $\geqslant x$. This can be obtained as follows :

$$\mathbb{E}[\text{Time OPT picks} \geqslant \text{x}] = \int_{t=0}^{1} \mathbb{P}[\text{OPT has picked} \geqslant \text{x at time t}]dt$$

Since the OPT picks the maximum seen so far, we can write the above as

$$= \int_{t=0}^{1} \mathbb{P}[\text{max from 1 to t is} \geqslant \text{x}]dt$$

The max from 1 to t will be distributed as F^{1-t}

$$= \int_{t=0}^{1} 1 - F^{1-t}(x)dt$$

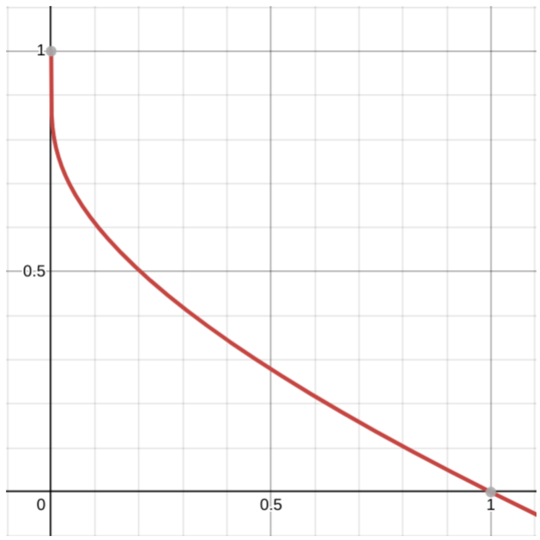
$$= \int_{t=0}^{1} 1 - F^{t}(x)dt$$

$$= 1 - \frac{1 - F(x)}{\ln(\frac{1}{F(x)})}$$

Hence we obtain :

$$\mathbb{E}[\text{Time OPT picks} \geqslant \text{x}] = 1 - \frac{1 - F(x)}{\ln(\frac{1}{F(x)})} \tag{18}$$

If we plot the curve for this with $F(x)$ on the x-axis, we obtain the following :



The y-axis above represents $\mathbb{E}[\text{Time OPT picks} \geqslant F^{-1}(x)]$

9.2 Analysis of the Algorithm

First, it must be the case that the optimal p must be increasing in nature. This is because for the optimal algorithm we must have that the threshold function matches the optimal dynamic programming thresholds obtained in 1, which had increasing thresholds. A similar argument can be used to claim that p(0)=0 for the optimal choice. We also must have p(1)=1, else we will end up commiting too early.

Let us compute the expected time that we hold on to an item $\geqslant x$. Before that we will define another quantity of interest, the distribution of the time at which we commit : $q(t)$, where $q(t)$ is defined as follows : $q(t) = \mathbb{P}[\text{ALG has not committed to an item in the time } (1...t)\ ]$. q and p are related as follows :

$$\begin{aligned} q(t) - q(t-dt) &= \mathbb{P}[\text{Commit to an item in time } (t-dt \text{ to } t)] \\ &= \mathbb{P}[\text{Not committed to something in (1,t) }] \cdot \mathbb{P}[\text{Commit to } x_t] \\ &= q(t) \cdot \left(1 - p(t)^{1/n}\right) \\ &= q(t) \cdot \left(\frac{1}{n} \ln\left(\frac{1}{p(t)}\right)\right) \end{aligned}$$

Note that we had chosen $dt = 1/n$ as $n \to \infty$, and hence the above can be written as :

$$= q(t) \cdot \ln\left(\frac{1}{p(t)}\right) dt$$

We obtain the following relation between q and p

$$\frac{q'(t)}{q(t)} = \ln\left(\frac{1}{p(t)}\right) \tag{19}$$

Now to calculate $\mathbb{E}[\text{time ALG holds on to} \geqslant x]$. Let t be the time when the threshold to commit is x. So F(x)=p(t).If we commit to an item in the time $(1, t)$, then the threshold at that time would have been higher which guarantees we would have committed to a value greater than x .

$$\begin{aligned}
\mathbb{E}[\text{time ALG holds on to} \geqslant x] &= \sum_{i=\lfloor nt \rfloor}^{n} \frac{i}{n} \cdot \mathbb{P}[\text{Commit to i-th item}] \\
&\quad + \sum_{i=0}^{\lfloor nt \rfloor} \frac{i}{n} \cdot \mathbb{P}[\text{Not commit to items } (i \dots n)] \cdot \mathbb{P}[\text{item i is} \geqslant x] \\
&= \sum_{i=\lfloor nt \rfloor}^{n} \frac{i}{n} \cdot \mathbb{P}[\text{Commit at time } \frac{i}{n}] + \sum_{i=0}^{\lfloor nt \rfloor} \frac{i}{n} \cdot q(\frac{i}{n}) \cdot \mathbb{P}[\text{item i is} \geqslant x]
\end{aligned}$$

Now we have $\mathbb{P}[\text{Commit at time } \frac{i}{n}] = \frac{1}{n} \cdot q'(\frac{i}{n})$

$$\begin{aligned}
&= \sum_{i=\lfloor nt \rfloor}^{n} \frac{i}{n} \cdot \frac{1}{n} \cdot q'(\frac{i}{n}) + \sum_{i=0}^{\lfloor nt \rfloor} \frac{i}{n} \cdot q(\frac{i}{n}) \cdot \left(1 - F^{\frac{1}{n}}\left(F^{-1}\left(p(t)\right)\right)\right) \\
&= \sum_{i=\lfloor nt \rfloor}^{n} \frac{i}{n} \cdot \frac{1}{n} \cdot q'(\frac{i}{n}) + \sum_{i=0}^{\lfloor nt \rfloor} \frac{i}{n} \cdot q(\frac{i}{n}) \cdot \left(1 - p(t)^{\frac{1}{n}}\right) \\
&= \sum_{i=\lfloor nt \rfloor}^{n} \frac{i}{n} \cdot \frac{1}{n} \cdot q'(\frac{i}{n}) + \sum_{i=0}^{\lfloor nt \rfloor} \frac{i}{n} \cdot q(\frac{i}{n}) \cdot \left(\frac{1}{n} \ln\left(\frac{1}{p(t)}\right)\right)
\end{aligned}$$

We let n $\to \infty$ and let $\frac{1}{n} = du$, we get :

$$= \int_{u=t}^{1} u \cdot q'(u) du + \int_{u=0}^{t} u \cdot q(u) \cdot \ln\left(\frac{1}{p(t)}\right) du \tag{20}$$

Applying integration by parts on the first equation and using q(1)=1, we get :

$$\mathbb{E}[\text{time ALG holds on to} \geqslant F^{-1}(p(t))] = 1 - t \cdot q(t) - \int_{u=t}^{1} q(u) du + \ln\left(\frac{1}{p(t)}\right) \cdot \int_{u=0}^{t} u \cdot q(u) du \tag{21}$$

From 18 we obtained that :

$$\mathbb{E}[\text{Time OPT picks} \geqslant \text{x}] = 1 - \frac{1 - F(x)}{\ln(\frac{1}{F(x)})}$$

Since F(x)=p(t), we get that :

$$\mathbb{E}[\text{Time OPT picks} \geqslant F^{-1}(p(t))] = 1 - \frac{1-p(t)}{\ln(\frac{1}{p(t)})} \tag{22}$$

Note that equation 21 is a function of both *p(t)* and *q(t)*, we instead substitute the following :

$$\begin{aligned} v(t) &= \ln(q(t)) \\ \implies v'(t) &= \frac{q'(t)}{q(t)} = -\ln(p(t)) \\ \implies p(t) &= -e^{-v'(t)} \end{aligned} \tag{23}$$

Substituting in 21 and 22 we get the following :

$$\mathbb{E}[\text{time ALG holds on to} \geqslant x] = 1 - te^{v(t)} - \int_{u=t}^{1} e^{v(u)}du + v'(t)\int_{u=0}^{t} ue^{v(u)}du$$

and

$$\mathbb{E}[\text{Time OPT picks} \geqslant \text{x}] = 1 - \frac{1-e^{-v'(t)}}{v'(t)} \tag{24}$$

Along with this we have the following boundary conditions on the optimal $v(t)$:

$$\begin{aligned} v(1) &= \ln(q(1)) = 0 \\ v'(1) &= -\ln(p(1)) = 0 \end{aligned} \tag{25}$$

The optimal function $v(t)$ also has the following properties that it will be :

1. $v(t)$ is an non-decreasing function since $q(t)$ is non-decreasing
2. $v'(t)$ is a non-increasing function since $p(t)$ is non-decreasing

Now say the optimal $p(t)$ guarantees a k-ASD guarantee. This would imply that $\forall 0 \leqslant x$:

$$\mathbb{E}[\text{time ALG holds on to} \geqslant x] = k \cdot \mathbb{E}[\text{Time OPT picks} \geqslant \text{x}]$$

ie. $\nexists\ 0 \ \leqslant x \leqslant 1$ such that

$$\mathbb{E}[\text{time ALG holds on to} \geqslant x] > k \cdot \mathbb{E}[\text{Time OPT picks} \geqslant \text{x}]$$

Because, if there were such an x that had $> k$ then we could suitably modify our threshold function in order to obtain an ASD guarantee $> k$. Equivalently for all t = [0,1] we have that

$$1 - te^{v(t)} - \int_{u=t}^{1} e^{v(u)}du + v'(t)\int_{u=0}^{t} ue^{v(u)}du = \mathbf{k} \cdot \left(1 - \frac{1-e^{-v'(t)}}{v'(t)}\right) \tag{26}$$

Since it holds for all t = [0,1]. We can differentiate it in order to obtain :

$$\int_{u=0}^{t} ue^{v(u)}du = \mathbf{k} \cdot \frac{1 - e^{-v'(t)} - v'(t)e^{-v'(t)}}{v'(t)^2} \tag{27}$$

In order to get rid of the integrals and obtain a differential equation we differentiate it again and obtain :

$$\mathbf{v''(t)} \cdot \mathbf{k}(v'(t)^2 e^{-v'(t)} - 2 + 2e^{-v'(t)} + 2v'(t)e^{-v'(t)}) - \mathbf{v'(t)^3}(te^{v(t)}) = 0 \tag{28}$$

This is a second order differential equation in $v(t)$, we have our 2 boundary conditions in 25. So we can solve this equation for a given k, and use the boundary conditions to obtain a valid $p(t)$. The largest k for which we can satisfy the coundary conditions will be the ASD guarantee and equivalently the competitive ratio.

A Expectation Formula

The expected value of a random variable with distribution F can be expressed as follows :

$$\mathbb{E}[X] = \int\limits_{x=0}^{\infty} (1 - F(x))dx - \int\limits_{x=-\infty}^{0} F(x)dx \tag{29}$$

This can be proved as follows :

$$\mathbb{E}[X] = \int\limits_{x=-\infty}^{0} xf(x)dx + \int\limits_{x=0}^{\infty} xf(x)dx$$

Computing each term separately and applying integration by-parts, we get:

$$\begin{aligned}\int\limits_{x=-\infty}^{0} xf(x)dx &= [xF(x)]_{-\infty}^{0} - \int\limits_{x=-\infty}^{0} F(x)\mathrm{d}x \\ &= - \int\limits_{x=-\infty}^{0} F(x)\mathrm{d}x \end{aligned} \tag{30}$$

and

$$\begin{aligned}\int\limits_{x=0}^{\infty} xf(x)dx &= [x(F(x)-1)]_{0}^{\infty} - \int\limits_{x=0}^{\infty} (F(x)-1)\,dx \\ &= \int\limits_{x=0}^{\infty} (1-F(x))\,dx \end{aligned} \tag{31}$$

Adding eq. (30) and eq. (31), we get eq. (29)

$$\tag{32}$$

Moreover, if the random variable X has a positive support ie. $F(x) = 0 \; \forall \; x < 0$, then eq. (29) can be simplified to

$$\mathbb{E}[X] = \int\limits_{x=0}^{\infty} (1 - F(x))dx \tag{33}$$

B ASDness equivalence to Competitiveness

B.1 A c-ASD algorithm is also c-Competitive

A c-ASD algorithm guarantees the following :

$$\mathbb{E}[\text{time ALG chooses} \geqslant \text{x}] \geqslant c \cdot \mathbb{E}[\text{ time OPT chooses} \geqslant \text{x}]$$

Let $G(x) = \mathbb{E}[\text{time ALG chooses} \geqslant \text{x}]$, then $1 - G(x) = \mathbb{E}[\text{time ALG chooses} < \text{x}]$. If we assume that there are no points masses in our distribution, then $1-G(x) = \mathbb{E}[\text{time ALG chooses} < \text{x}]$, this in turn represents the CDF of a random variable that denotes the time-distribution for which we hold on to an item with value x. The expected value of this random variable is the weighted sum of items value and the time for which we hold on to the item, which represents the pay off of the algorithm. From eq. (33) we know that this can be represented as :

$$\int_{x=0}^{\infty} (1-(1-G(x)))dx = \int_{x=0}^{\infty} G(x)dx$$
$$\implies \mathbb{E}[ALG] = \int_{x=0}^{\infty} \mathbb{E}[\text{time ALG chooses} \geqslant \text{x}]dx$$
$$\geqslant c \cdot \int_{x=0}^{\infty} \mathbb{E}[\text{time OPT chooses} \geqslant \text{x}]dx$$

With a similar argument as above we can also show that

$$\mathbb{E}[OPT] = \int_{x=0}^{\infty} \mathbb{E}[\text{time OPT chooses} \geqslant \text{x}]dx$$

And therefore a c-ASD algorithm guarantees that

$$\mathbb{E}[\text{ALG}] \geqslant c \cdot \mathbb{E}[\text{OPT}] \tag{34}$$

and hence it is c-Competitive.

B.2 A c-competitive algorithm allows for a c-ASD algorithm

We will prove this for distributions that have a finite support. We will prove that the contrapositive holds true, ie. if there does not exist an algorithm that is c-ASD, then there does not exist an algorithm that is c-Competitive. Let the support of the distribution F be the finite set $a_1, \dots, a_m$. Any deterministic algorithm merely sets a threshold $\tau(i)$ for each item i, these thresholds could depend on the past realisations seen by the algorithm, however note that all thresholds $\tau(i)$ lying in the range $[0, a_1]$ or $(a_{j-1}, a_j]$ behave identically to any other threshold within the same range. Thus there are only a finite set of distinct deterministic Algorithms call them $A_1, A_2, \dots, A_N$ possible. A general randomised algorithm can be seen as behaving as A_i with probability $\lambda_i \geqslant 0$ for $i = 1, \dots, N$. Suppose no Algorithm is c-ASD then we have the following guarantee :

$$\max_{\lambda_1,\dots,\lambda_N} \min_{a_1,a_2,\dots,a_m} \frac{\mathbb{E}[\text{days ALG picks} \geqslant a_j]}{\mathbb{E}[\text{days OPT picks} \geqslant a_j]} < c$$
$$\implies \max_{\lambda_1,\dots,\lambda_N} \min_{a_1,a_2,\dots,a_m} \sum_{i=1}^{N} \frac{\mathbb{E}[\text{days } A_i \text{ picks} \geqslant \text{a}_j]}{\mathbb{E}[\text{days OPT picks} \geqslant a_j]} < c$$

This can be written in the form of an LP as follows :

$$\text{maximise } \alpha, \quad \text{subject to :}$$
$$\mathbb{E}[\text{days OPT picks} \geqslant a_j] \cdot \alpha - \sum_{i=1}^{N} \lambda \cdot \mathbb{E}[\text{days } A_i \text{ picks } \geqslant \text{a}_j] \leqslant 0 \quad \forall j = 1, \ldots, m$$
$$\sum_{i=1}^{N} \lambda_i = 1$$
$$\lambda_i \geqslant 0 \quad \forall i = 1, \ldots, N$$

The Dual of the above LP is the following :

$$\text{minimise } z, \quad \text{subject to :}$$
$$\sum_{j=1}^{m} y_j \cdot \mathbb{E}[\text{days OPT picks} \geqslant a_j] = 1$$
$$z - \sum_{j=1}^{m} y_j \cdot \mathbb{E}[\text{days } A_i \text{ picks } \geqslant \text{a}_j] \geqslant 0 \quad \forall\ i = 1, 2, \ldots, N$$
$$y_j \geqslant 0\ \forall\ j = 1, 2, \ldots, m$$

say z attains the minimum z^* at $y_1^*, y_2^*, \ldots, y_m^*$, then we have that :

$$\sum_{j=1}^{m} y_j^* \cdot \mathbb{E}[\text{days OPT picks} \geqslant a_j] = 1 \tag{35}$$
$$z^* - \sum_{j=1}^{m} y_j^* \cdot \mathbb{E}[\text{days } A_i \text{ picks } \geqslant \text{a}_j] \geqslant 0 \quad \forall\ i = 1, 2, \ldots, N$$
$$y_j^* \geqslant 0\ \forall\ j = 1, 2, \ldots, m$$

and

$$, z^* < c \tag{36}$$

The proof after this is slightly rushed and casual.

$$\text{Define } \Phi : \{0, a_1, \ldots, a_m\} \rightarrow \{0, y_1^*, y_1^* + y_2^* \ldots, y_1^* + y_2^*, + \ldots + y_m^*\}$$
$$\text{as}\Phi(0) = 0, \Phi(a_i) = y_1^* + \ldots + y_i^*$$

Since $y_i^*{}'sare \geqslant 0$, Φ is monotone.
Define $X_i' = \Phi(X_i)$,
then we we have that :

$$OPT = \sum_{i=1}^{n} \max_i(\Phi(x_i))$$
$$= \sum_{i=1}^{n} \Phi(\max_i((x_i)))$$

Hence we have that :

$$\mathbb{E}[OPT(\Phi(X))] = \sum_{i=1}^{n} \mathbb{E}[\Phi(\max_i((x_i)))]$$

Consider any deterministic algorithm A' on $\{X_1', \ldots, X_n'\}$, without loss of generality it uses the threshold from the support of $X_i's$,ie the image of Φ.

$$\begin{aligned}
\mathbb{E}[\text{Payoff of A'}] &= \sum_{i=1}^{m} y_i^* \mathbb{E}[\text{days A' picks } \geqslant y_1^* + \ldots + y_i^*] \\
&= \sum_{i=1}^{m}{}_{i:y_i*>0}\ y_i^* \mathbb{E}[\text{days ALG* picks } \geqslant y_1^* + \ldots + y_i^*] && (37) \\
&\leqslant z^* \leqslant c \cdot \mathbb{E}[OPT(\Phi(X))] && (38)
\end{aligned}$$

Which establishes that no determinstic algorithm on finite support can have a competitve ratio more than c.

References

[1] Andreas Abels, Elias Pitschmann, and Daniel Schmand. Prophet-inequalities over time, 2022.