# Basic Prophet Inequality and Pandora Box Problem

Hu Fu

November 20, 2019

## 1 Basic Prophet Inequality

There are $n$ boxes 1, 2, $\ldots, n$. Each box $i$ contains an invisible number $v_i \geq 0$ drawn from distribution $F_i$ independently. We know the distributions beforehand. The boxes arrive in the order $1, 2, \ldots, n$. When box $i$ arrives, we observe $v_i$ and need to decide whether to take the box or not. If we take the box $i$, the game stops and our reward is $v_i$; if we let box $i$ pass, we cannot go back to retrieve it.

An algorithmic task is to design an algorithm that, given the distributions $F_1, \cdots, F_n$, maximizes the expected reward. After a moment's thought, we recognize that the optimal algorithm is a "backward induction": if we wait till the last box, the expected reward we will get is $\mathbf{E}[v_n]$; given this, when box $n-1$ comes, we should only take any value greater than $\mathbf{E}[v_n]$; this allows us to compute the expected value we get when we wait till the last two boxes, which we may use to decide the threshold to be used for box $n-3$, and so on and so forth. The backward induction needs to know the order in which the boxes arriveis. When each box $i$ arrives, the algorithm holds a threshold $\theta_i$ such that we take box $i$ if and only if $v_i \geq \theta_i$. It is not hard to see that $\theta_i$ decreases as $i$ increases.

The *prophet inequality* problem, however, asks a question that is more about the value of information. It is concerned with comparing the performance of an online algorithm with an prescient benchmark: if a prophet knows beforehand all the values in the box, then the expected performance of the prophet is $\mathbf{E}[\max v_i]$. How well can an online which only knows the distribution do in comparison to this benchmark?

**Theorem 1.** *There exists a threshold $\theta$ such that accepting the first box with value at least $\theta$ achieves expected value at least $\frac{1}{2}\mathbf{E}[\max_i v_i]$. For any $\epsilon > 0$, there is no online algorithm whose performance is guaranteed to be at least $(0.5+\epsilon)\mathbf{E}[\max v_i]$.*

### 1.1 Quantile Approach

Let the random variable $v^*$ be $\max_i v_i$. Then the cdf of $v^*$ is $F = \prod_i F_i$. Let $\theta$ be $F^{-1}(\frac{1}{2})$. Let's assume $\mathbf{Pr}[\exists i, v_i = \theta] = 0$; this is true, for example, when all distributions are atomless. We show that accepting the first box with value at least $\theta$ yields expected value at least $\frac{1}{2}\mathbf{E}[v^*]$.

We first give an upper bound on the prophet's value:

$$\begin{aligned}\mathbf{E}\left[v^*\right] &= \frac{1}{2}\mathbf{E}\left[v^* \mid v^* < \theta\right] + \frac{1}{2}\mathbf{E}\left[v^* \mid v^* \geq \theta\right] \leq \frac{1}{2}\theta + \frac{1}{2}\mathbf{E}\left[\theta + (v^* - \theta) \mid v^* \geq \theta\right] \\ &= \theta + \frac{1}{2}\mathbf{E}\left[v^* - \theta \mid v^* \geq \theta\right] = \theta + \mathbf{E}\left[(v^* - \theta)_+\right],\end{aligned}$$

where $(x)_+$ denotes $\max(x, 0)$.

With probability $\frac{1}{2}$, the algorithm with threshold $\theta$ accepts a box, and that gives value at least $\frac{1}{2}\theta$ even if the value of the accepted box is just $\theta$. On top of this, values that are strictly greater than $\theta$ contribute more. That additional contribution in expectation is

$$\sum_i \mathbf{E}\left[(v_i-\theta)_+\right]\cdot\mathbf{Pr}\left[\text{box } i \text{ is looked at}\right] \geq \sum_i \mathbf{E}\left[(v_i-\theta)_+\right]\cdot\mathbf{Pr}\left[\text{no box is taken in the end}\right]$$
$$= \frac{1}{2}\sum_i \mathbf{E}\left[(v_i-\theta)_+\right] \geq \frac{1}{2}\mathbf{E}\left[(v^*-\theta)_+\right].$$

Therefore in total the algorithm yields expected value at least $\frac{1}{2}(\theta + \mathbf{E}[(v^*-\theta)_+])$.

**Remark** In lower bounding the contribution from the part $(v_i-\theta)_+$, we use that the threshold price is accepted with probability at most $\frac{1}{2}$, whereas in lower bounding the contribution from the part $\theta$, we used that the threshold price is accepted with probability at least $\frac{1}{2}$. Therefore the atomless assumption was important. If the condition does not hold, one needs to break ties very carefully. (How?)

### 1.2 An Economic Interpretation

If we think of selling a single item to a sequence of bidders, where each bidder $i$ has value $v_i$ drawn independently from distribution $F_i$, then a threshold algorithm can be seen as the simplest selling strategy: post a price $\theta$, and sell it to the first buyer $i$ who would like to buy at this price, i.e., $v_i \geq \theta$. The problem then asks for a selling strategy so that in expectation the buyer who buys the item should have a high value. This is known as *social welfare* maximization in economics.

Under this perspective, $\theta \cdot \mathbf{Pr}[\text{some buyer buys}]$ is the seller's *revenue* (or, in other words, the seller's utility), whereas $\sum_i (v_i-\theta)_+ \cdot \mathbf{Pr}[i \text{ was considered}]$ is the sum of the buyers' *utilities*. Therefore the calculation we performed above bounds respectively the revenue and the sellers' utility, and the welfare of a transaction is just the sum of the seller's revenue and the buyers' utility!

### 1.3 Balanced Price Approach

An alternative approach gives the same guarantee but is more robust (without needing to be careful with tie-breaking, for one thing, and, for another, being in fact also more robust against inaccuracy in the input distributions). We present the proof using economic terms introduced above.

Consider $\theta = \frac{1}{2}\mathbf{E}[\max_i v_i]$. With this posted price $\theta$, let $p$ denote the probability that any buyer purchases. Then the seller's revenue is $\theta p$. The buyers' utilities is

$$\sum_i \mathbf{E}\left[(v_i-\theta)_+\right]\cdot\mathbf{Pr}\left[\text{buyer } i \text{ has an opportunity to purchase}\right]$$
$$\geq \sum_i \mathbf{E}\left[(v_i-\theta)_+\right](1-p) \geq (1-p)\,\mathbf{E}\left[\max_i v_i - \theta\right] = \theta.$$

Therefore the welfare is at least $\theta$.

### 1.4 Lower bound

To see the lower bound, consider two boxes; $v_1$ is deterministically 1, and $v_2$ is $h$ with probability $1/h$ and 0 with probability $1 - 1/h$, for some arbitrarily large $h$. The prophet's performance on this instance is $2 - 1/h$. whereas any online algorithm, by either taking or not taking the first box, can get no more value than 1.

## 2 Pandora Box Problem

In the Pandora Box problem, we again have $n$ boxes, each box $i$ containing a hidden value $v_i$ drawn independently from a known distribution $F_i$. Again, we are allowed to pick only one box. Instead of having the boxes arriving in an adversary order, we can choose the order in which to open the box; also, we are allowed to retrieve a box which we opened and temporarily decided not to take immediately. On each box $i$, there is also a cost $c_i \geq 0$: so in order to open box $i$, we must first pay a cost $c_i$. The Pandora Box problem asks for an algorithm that, given this information, decides on a procedure that maximizes the expected utility, which is the value in the box we take minus all the search costs we pay along the way. We are not allowed to take a box we did not open.

### 2.1 The index algorithm

For each box $i$, with its distribution $F_i$ and search cost $c_i$, if we were also offered another box which deterministically contains a value $v$, how large should $v$ be so that we are indifferent between (a) opening box $i$ first and then deciding which one to take, and (b) skipping box $i$ and directly taking the deterministic value?

The break-even value $v$ is such that the expected *additional* value we obtain from opening box $i$ covers exactly the cost of opening it, i.e., $\mathbf{E}_{(v_i - v)_+}[=]c_i$. There is a unique solution to this equation. Let $\theta_i$ denote this solution, and call it the *index* of box $i$.

The *index algorithm* uses the following procedure: Initialize by writing on each box its index. Remove all boxes with negative indices. Among the remaining ones, open the box with the highest index, and replace the index written on it by the value contained in it. If at any point the largest number written on a box is a value (instead of an index), take that box and quit; otherwise open a box with the currently highest index (breaking ties arbitrarily).

### 2.2 Optimality of the index algorithm

**Theorem 2.** *Among all procedures, the index algorithm has the highest expected utility.*

*Proof.* Consider any algorithm. For each box $i$, let $I_i$ be the indicator variable for the event that the algorithm opens box $i$, and $A_i$ be the indicator variable for the event that the algorithm takes box $i$. Then the algorithm's expected utility is $\sum_i \mathbf{E}[A_i v_i - c_i I_i]$. Recall that $c_i = \mathbf{E}_{v_i' \sim F_i}[(v_i - \theta_i)_+]$. So

$$\mathbf{E}_{\boldsymbol{v}}\left[A_i v_i - c_i I_i\right] = \mathbf{E}_{\boldsymbol{v}}\left[A_i v_i - \mathbf{E}_{v_i' \sim F_i}\left[(v_i' - \theta_i)_+\right] \cdot I_i\right] = \mathbf{E}_{\boldsymbol{v}}\left[A_i v_i - (v_i - \theta_i)_+ \cdot I_i\right].$$

Replacing $v_i'$ by $v_i$ is crucial and subtle. They are from the distribution, but note that $I_i$ depends on $\boldsymbol{v}$. The crucial observation is that $I_i$ is independent from $v_i$ — whether an algorithm opens box $i$

cannot be affected by what is actually contained in it. Now we upper bound the expected utility of the algorithm, using the fact that $A_i \leq I_i$:

$$\begin{aligned}\sum_i \mathbf{E}\left[A_i v_i - c_i I_i\right] &= \sum_i \mathbf{E}\left[A_i v_i - (v_i - \theta_i)_+ \cdot I_i\right] \\ &\leq \sum_i \mathbf{E}\left[A_i v_i - (v_i - \theta_i)_+ \cdot A_i\right] = \sum_i \mathbf{E}\left[A_i \cdot \min(v_i, \theta_i)\right].\end{aligned}$$

Since for any realization of $\boldsymbol{v}$, we have $\sum_i A_i \leq 1$, so we have

$$\sum_i \mathbf{E}\left[\min(v_i, \theta_i)\right] \leq \mathbf{E}\left[\max_i \min(v_i, \theta_i)\right].$$

We argue that the index algorithm achieves exactly this upper bound. In this line of derivation, there are two inequalities. The first inequality would be tight if the algorithm satisfies the following property: whenever it opens a box and sees a value greater than the index written on the box, the algorithm must take it. The index algorithm does satisfy this property. The second inequality is satisfied if the box chosen by the algorithm always maximizes $\min(v_i, \theta_i)$. The index algorithm also satisfies this. □