Basic Prophet Inequality, Pandora Box Problem and Online Contention Resolution Schemes

Hu Fu

December 29, 2021

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 θ_i such that we take box i if and only if $v_i \geq \theta_i$. It is not hard to see that θ_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 θ such that accepting the first box with value at least θ 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 θ 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 θ 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 θ accepts a box, and that gives value at least $\frac{1}{2}\theta$ even if the value of the accepted box is just θ. On top of this, values that are strictly greater than θ contribute more. That additional contribution in expectation is

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

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 θ, 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 θ, 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 θ, let p denote the probability that any buyer purchases. Then the seller's revenue is θp. 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 θ.

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 Matroid Prophet Inequalities

Prophet inequalities for matroids were first proved by Kleinberg and Weinberg (2019). We present the same algorithm with a slightly stronger guarantee against the *ex ante* optimal. The ex ante version was proposed by Lee and Singla (2018), who then used it to give Online Contention Resolution Schcmes (OCRS, see Section 4 and Feldman et al., 2016).

As before, let $[n]$ be the set of boxes, each box i containing a value v_i independently drawn from distribution F_i. Let $\mathcal{M}$ be a matroid on $[n]$, with $\mathcal{I}$ the set of independent sets. Boxes arrive in an adversarial order which is known to the algorithm before hand. (It is important that the adversary cannot let the arrival order depend on the realized values.) Without loss of generality, assume the boxes arrive in the order $1, 2, \ldots, n$. The algorithm upon seeing the value v_i in box i when it arrives, must decide whether to take the box or not, and cannot retract later an accepted box. At any time the set of boxes accepted by the algorithm must be an independent set of $\mathcal{M}$. The algorithm aims to maximize the total value in the accepted boxes.

The natural prophet benchmark is $\mathbf{E}[\max_{S \in \mathcal{I}} \sum_{i \in S} v_i]$. Let $\mathcal{P}_{\mathcal{M}}$ be the polytope associated with the matroid M, the *ex ante* optimal is defined as:

$$\max_{\boldsymbol{x} \in \mathcal{P}_{\mathcal{M}}} \sum_i \mathbf{E}\left[v_i x_i \;\middle|\; v_i \text{ is in the top } x_i \text{ quantile of } F_i\right]. \tag{1}$$

It is easy to see that the ex ante optimal is no less than the prophet benchmark.[1] The ex ante benchmark in general can be strictly greater than the prophet, although the gap is bounded by a factor of $\frac{e}{e-1}$, and is known as the *correlation gap* (Agrawal et al., 2012).

A *threshold algorithm* computes, when each box i arrives, a threshold θ_i, so that box i is accepted if and only if $v_i \geq \theta_i$. Two remarks are in order:

- The threshold θ_i is computed using only information before box i's arrival, which includes the observed values $v_1, \ldots, v_{i-1}$ and the set of boxes that have been accepted after box $i-1$, which we denote as A_{i-1};

[1] We did not write the conditioning in (1) simply as $v_i \geq F_i^{-1}(1 - x_i)$, to avoid discussion of tie-breaking when there is a probability mass on $F_i^{-1}(1 - x_i)$.

- If $A_{i-1} \cup \{i\} \notin \mathcal{I}$, then θ_i should be set as ∞.

Theorem 2. *There is a threshold algorithm that collects a total value in expectation at least half of the ex ante optimal.*

Let's define a correlated distribution on $\boldsymbol{v}' = (v'_1, \dots, v'_n)$. These random variables are independent from $\boldsymbol{v}$ but have the same marginal distributions. Define

$$\boldsymbol{x}^* := \mathrm{argmax}_{\boldsymbol{x} \in \mathcal{P}_\mathcal{M}} \sum_i \mathbf{E}\left[v_i x_i \ \middle| \ v_i \text{ is in the top } x_i \text{ quantile of } F_i\right].$$

Since $\boldsymbol{x}^* \in \mathcal{P}_\mathcal{M}$, $\boldsymbol{x}^*$ can be expressed as a convex combination $\boldsymbol{x}^* = \sum_{S \in \mathcal{I}} \alpha_S \mathbb{1}_S$, where $\mathbb{1}_S$ is the indicator variable of set S, and $\sum_S \alpha_S = 1$, $\alpha_S \geq 0, \forall S$. $\boldsymbol{v}'$ is defined by $\boldsymbol{x}^*$: first draw $\boldsymbol{x} \in \{0,1\}^n$, with probability α_S, $\boldsymbol{x} = \mathbb{1}_S$; then for each $i \in [n]$, if $x_i = 1$, v'_i is drawn the top x^*_i quantile of F_i; otherwise v'_i is drawn from the bottom $1 - x^*_i$ quantile of F_i.

Proposition 1. *The ex ante optimal is at most* $\mathbf{E}[\max_{S \in \mathcal{I}} \sum_{i \in S} v'_i]$.

Proof. Note that $\{\alpha_S\}_S$ defines a distribution over independent sets. Let's denote this distribution by α, and denote the conditional distribution of $\boldsymbol{v}'$ given S as F_S. Then

$$\mathbf{E}_{\boldsymbol{v}'}\left[\max_{S \in \mathcal{I}} \sum_{i \in S} v'_i\right] \geq \mathbf{E}_{S \sim \alpha}\left[\mathbf{E}_{\boldsymbol{v}' \sim F_S}\left[\sum_{i \in S} v'_i\right]\right] = \mathbf{E}_{S \sim \alpha}\left[\sum_{i \in S} v_i \ \middle| \ v_i \text{ is in the top } x^*_i \text{ quantile of } F_i\right].$$

By definition of α, when S is drawn from α, with probability precisely x^*_i, i is in S. Therefore the right hand side is just $\sum_i \mathbf{E}[v_i x^*_i \ | \ v_i \text{ is in the top } x_i \text{ quantile of } F_i]$.

$\square$

This shows an inequality, which suffices for our purpose. In fact one can show that equality holds, by showing that $\boldsymbol{x}^*$ can be arrived at by a continuous greedy procedure.

Notations: Let A_i denote the set of boxes accepted by a threshold algorithm after the i-th box, and $A = A_n$ the final selection; for any realization of v'_i, let B be $\mathrm{argmax}_{B \in \mathcal{I}} \sum_{i \in B} v'_i$; then there must be a partition of B into C and R such that $|C| = |A|$ and $A \cup R \in \mathcal{I}$; among all such partitions let $R(A)$ and $C(A)$ be such that $\sum_{i \in R(A)} v'_i$ is maximized. Note that A depends only on $\boldsymbol{v}$, whereas $C(A)$ and $R(A)$ depend on both $\boldsymbol{v}$ and $\boldsymbol{v}'$.

Definition 1. A threshold algorithm is said to be α-balanced if for any realizaed $\boldsymbol{v}$, for any $V \subseteq [n]$ such that $V \cup A \in \mathcal{I}$, the thresholds used by the algorithm satisfy

$$\sum_{i \in A} \theta_i \geq \frac{1}{\alpha} \mathbf{E}\left[\sum_{i \in C(A)} v'_i\right]; \tag{2}$$

$$\sum_{i \in V} \theta_i \leq \left(1 - \frac{1}{\alpha}\right) \mathbf{E}\left[\sum_{i \in R(A)} v'_i\right]. \tag{3}$$

Lemma 3. *An α-balanced threshold algorithm obtains expected value at least $\frac{1}{\alpha}$ fraction of the ex ante optimal.*

Proof. By definition of $\boldsymbol{v}'$, $R(A)$ and $C(A)$, the ex ante optimal is $\mathbf{E}[\sum_{i\in C(A)} v'_i + \sum_{i \in R(A)} v'_i]$.

By (2),

$$\mathbf{E}_{\boldsymbol{v}}\left[\sum_{i\in A}\theta_i\right] \geq \frac{1}{\alpha}\,\mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in C(A)} v'_i\right]. \tag{4}$$

With the economic interpretation (see Section 1.2), this part of the value can be seen as the revenue, and the remaining part is the utility, which we lower bound as follows.

$$\mathbf{E}_{\boldsymbol{v}}\left[\sum_{i\in A}(v_i-\theta_i)_+\right] = \mathbf{E}_{\boldsymbol{v}}\left[\sum_{i\in [n]}(v_i-\theta_i)_+\right] = \mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in [n]}(v'_i-\theta_i)_+\right] \geq \mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)}(v'_i-\theta_i)_+\right].$$

The first equality is by definition of a threshold algorithm. The second equality, which is the key step that generalizes Kleinberg and Weinberg's original proof, follows from the observations that

(a) θ_i depends only on $v_1, \ldots, v_{i-1}$ but not on v_i, nor on $\boldsymbol{v}'$;

(b) $\boldsymbol{v}$ is independent from $\boldsymbol{v}'$, and therefore in particular v'_i is independent from $v_1, \ldots, v_{i-1}, \theta_i$;

(c) v'_i has the same marginal distribution as v_i.

We can now apply (3) and bound

$$\begin{aligned}\mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)}(v'_i-\theta_i)_+\right] &\geq \mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)}(v'_i-\theta_i)\right] = \mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)} v'_i\right] - \mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)}\theta_i\right] \\ &\geq \mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)} v'_i\right] - \left(1-\frac{1}{\alpha}\right)\mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)} v'_i\right] = \frac{1}{\alpha}\,\mathbf{E}_{\boldsymbol{v},\boldsymbol{v}'}\left[\sum_{i\in R(A)} v'_i\right].\end{aligned} \tag{5}$$

The lemma follows by summing up (4) and (5) $\square$

It remains to construct a 2-balanced threshold algorithm. This part is identical to Kleinberg and Weinberg's original one.

Define $f : \mathcal{I} \to \mathbb{R}_+$ as $f(S) \coloneqq \mathbf{E}[\sum_{i\in R(S)} v'_i]$.

Lemma 4. *The threshold algorithm with the following thresholds is 2-balanced: for each box i, if $A_{i-1}\cup\{i\} \notin \mathcal{I}$, let θ be ∞; otherwise $\theta_i \coloneqq \frac{1}{2}(f(A_{i-1}) - f(A_{i-1}\cup\{i\}))$.*

Proof. We check the two properties of 2-balancedness. For any $\boldsymbol{v}$, number the items in A as $a_1, \ldots, a_k$, and let a_0 be 0, then

$$\sum_{i \in A} \theta_i = \frac{1}{2} \sum_{j=1}^{k} f(A_{a_{j-1}}) - f(A_{a_j}) = \frac{1}{2} \left[f(A_0) - f(A_{a_k}) \right] = \frac{1}{2} \left[f(\emptyset) - f(A) \right] = \frac{1}{2} \mathbf{E} \left[\sum_{i \in C(A)} v'_i \right].$$

It is elementary to show that $f(\cdot)$ is submodular (or refer to Kleinberg and Weinberg, 2019 for a rigorous proof). For any $V \subseteq [n]$ such that $A \cup V \in \mathcal{I}$, let the elements in V be $a_1, \ldots, a_k$, let W_j be $A \cup \{a_1, \ldots, a_j\}$ for $j = 1, \cdots, k$, and $W_0 = A$, then

$$\sum_{i \in V} \theta_i \leq \frac{1}{2} \sum_{i \in V} (f(A) - f(A \cup \{i\})) \leq \frac{1}{2} \sum_{j=1}^{k} (f(W_{j-1}) - f(W_j)) = \frac{1}{2} f(A).$$

□

3 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.

3.1 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., $\mathbf{E}_{v_i}[(v_i - v)_+] = c_i$. There is a unique solution to this equation. Let θ_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).

3.2 Optimality of the index algorithm

Theorem 5. *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 $A_i \le I_i$:

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

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

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

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. $\square$

3.3 Generalization of Pandora Box Problem: Price of Information

It is natural to generalize the Pandora Box Problem to a variety of combinatorial optimization problems (Singla, 2018). Given a set of feasible sets $\mathcal{F} \subseteq 2^{[n]}$, distributions $F_1, \cdots, F_n$ and search costs $c_1, \ldots, c_n$, we are allowed to query any unopened box i at cost c_i and observe value $v_i \sim F_i$. At any point, we may take a subset T of opened boxes, with $T \in \mathcal{F}$. We are to maximize the expected total value of boxes we take minus the total search costs of boxes we open.

The basic Pandora Box algorithm then is the special case when $\mathcal{F}$ is the set of all singleton sets plus the empty set.

Exercise 1. Give an optimal algorithm when $\mathcal{F}$ is the set of independent sets of a matroid.

Exercise 2. Given a feasibility system $\mathcal{F} \subseteq 2^{[n]}$ with weights $w_1, \ldots, w_n \ge 0$. Consider the problem of maximizing the max weight feasible set: $\max_{T \in \mathcal{F}} \sum_{i \in T} w_i$. If the Greedy algorithm guarantees $\frac{1}{\alpha}$-approximation for the problem for $\alpha \le 1$, describe an algorithm that guarantees $\frac{1}{\alpha}$-approximation for the Pandora box problem defined on $\mathcal{F}$.

The greedy algorithm initializes $T = \emptyset$, then iterates until no element can be added to T: let $i \in \operatorname{argmax}_{j: T \cup \{j\} \in \mathcal{F}} v_j$, $T \leftarrow T \cup \{i\}$.

4 Online Contention Resolution Schemes

Online Contention Resolution Schemes (OCRS) are closely related to Prophet Inequalities and Pandora Box Problem. We first give basic definitions and examples, and then describe the connections.

4.1 Definitions

Given a set $[n]$ of elements and a set of *feasible* sets $\mathcal{F} \subseteq 2^{[n]}$, we consider the convex hull of the indicator vectors of the feasible sets: $\mathcal{P}_{\mathcal{F}} := \mathrm{Conv}(\{\mathbb{1}_S\}_{S \in \mathcal{F}}) \subseteq [0,1]^{[n]}$. $\mathcal{F}$ is usually downward closed, i.e., $S \in \mathcal{F}, T \subseteq S \Rightarrow T \in \mathcal{F}$.

We are given a point $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathcal{P}_{\mathcal{F}}$. Each element $i \in [n]$ is *active* with probability x_i, independently from the other elements. Elements arrive one by one. When element i arrives, we get to observe whether it is active, and if it is, must decide immediately whether to select it into the solution. Once an element is selected, we cannot retract it. The solution set must remain feasible at all time. Our algorithm is said to be an α-selectable OCRS if each i, conditioning on its being active, is selected with probability at least α.

There are usually two arrival orders we consider. In the *random order* model, the elements arrive in an order that is uniformly at random, as in the secretary problem. OCRS in this model is often abbreviated as RCRS (R for "random"). In the *adversarial order* model, the elements arrive in an order that is prespecified, which may depend on $\boldsymbol{x}$. In the adversarial order model, it makes a difference whether the adversary knows the realization of the elements' statuses. The adversary is sometimes said to be *almighty* if he has access to this information. In these notes, we assume the adversary has no such information.

4.2 Basic Examples

In this section we consider perhaps the simplest, nontrivial feasible sets, where $\mathcal{F}$ contains all the singleton sets and the empty set. $\mathcal{P}_{\mathcal{F}}$ is simply $\Delta([n])$, the simplex on $[n]$. Note that in this setting we can accept at most one element.

Example 1. A $\frac{1}{4}$-selectable OCRS in the adversarial order setting. Start with the solution set $T = \emptyset$. Whenever we see an active element i and T is empty, select i.

Proof. Each element i, conditioning on its being active, is selected with probability half times the probability with which T is empty before we see i. Let E_j denote the event that element j is accepted. Then

$$\mathbf{Pr}\left[T = \emptyset \text{ when } i \text{ arrives}\right] \geq 1 - \mathbf{Pr}\left[\cup_{j \neq i} E_j\right] = 1 - \sum_{j \neq i} \mathbf{Pr}\left[E_j\right] \geq 1 - \sum_{j \neq i} \frac{x_j}{2} \geq \frac{1}{2}.$$

The equality is an application of the union bound; since the events $E_1, \cdots, E_n$ are disjoint, equality is attained here. The second inequality comes from the bound $\mathbf{Pr}[E_j] \leq \frac{x_j}{2}$.

Therefore with probability at least $\frac{1}{4}$, element i is accepted whenever it is active. $\square$

Example 2. A $\frac{1}{2}$-selectable OCRS in the adversarial order setting: when element i arrives and is active, if the algorithm has selected no other element, let y_i be the probability with which the algorithm has not selected anything before i arrives, and select i with probability $\frac{1}{2y_i}$.

Proof. It suffices to show $y_i \geq \frac{1}{2}$ for every i. We show this by an induction. Without loss of generality, assume the elements arrive in the order $1, 2, \ldots, n$. When element 1 arrives, $y_1 = 1$. Now suppose $y_1, \ldots, y_{i-1} \geq \frac{1}{2}$. Let E_j denote the event that element j is active and selected. Then $\mathbf{Pr}[E_j] = x_j/2$.

$$y_i = 1 - \sum_{j=1}^{i-1} \mathbf{Pr}\left[E_j\right] = 1 - \sum_{j=1}^{i-1} \frac{x_j}{2} \geq \frac{1}{2}.$$

The second equality is again an application of the union bound on disjoint events. $\square$

Example 3 (Lee and Singla, 2018)**.** A $(1 - \frac{1}{e})$-selectable RCRS: one way of realizing a uniformly random order is to let each element i independently draw a time t_i from $[0, 1]$ uniformly at random, and then let element i arrive at time t_i. Our algorithm, when seeing element i active, selects i with probability $e^{-t_i x_i}$ if it has not taken an element.

Proof. We again lower bound the probability with which the algorithm can still accept element i when it arrives at time t_i:

$$\begin{aligned} &\mathbf{Pr}\left[\text{no element has been selected when element } i \text{ arrives at time } t_i\right] \\ \geq &\prod_{j \neq i} \left(1 - x_j \int_0^{t_i} e^{-s x_j} \, \mathrm{d}s\right) = \prod_{j \neq i} \left(1 - (1 - e^{-t_i x_j})\right) = e^{-\sum_{j \neq i} t_i x_j}. \end{aligned}$$

Therefore, when active, element i is selected with probability at least

$$\int_0^1 e^{-t x_i} \cdot e^{-t \sum_{j \neq i} x_j} \, \mathrm{d}t \geq \int_0^1 e^{-t} \, \mathrm{d}t = 1 - \frac{1}{e}.$$

$\square$

Exercise 3. Show that, for any $\epsilon > 0$, for large enough n there exists no $(1 - \frac{1}{e} + \epsilon)$-selectable OCRS in random arrival order.

4.3 Connections with Prophet Inequalities

The prophet inequality problem is naturally defined on a general feasbility system: given $\mathcal{F} \subseteq 2^{[n]}$, and distributions $F_1, \cdots, F_n$, as box i arrives, $v_i \sim F_i$ is revealed, and we must decide immediately whether to select the box to the solution. At all time, the set of selected boxes must be feasible. Let T denote the set of boxes selected by the algorithm. An algorithm is α-competitive if

$$\mathbf{E}\left[\sum_{i \in T} v_i\right] \geq \alpha \, \mathbf{E}\left[\max_{S \in \mathcal{F}} \sum_{i \in S} v_i\right].$$

Theorem 6. *If for a feasibility system $\mathcal{F}$ there is an α-selectable OCRS against a certain arrival model, then there is a prophet inequality algorithm that is α-competitive in the same arrival model.*

Proof. For every $\boldsymbol{v}$, fix an $\mathrm{OPT}(\boldsymbol{v}) \in \mathrm{argmax}_{T\in\mathcal{F}} \sum_{i\in T} v_i$. Let x_i^* be $\mathbf{Pr}_{\boldsymbol{v}}[i \in \mathrm{OPT}(\boldsymbol{v})]$. Then $\boldsymbol{x}^* \in \mathcal{P}_{\mathcal{F}}$. Feed $\boldsymbol{x}^*$ as the input to the given OCRS algorithm. Run the prophet inequality problem as follows. When item i arrives, we observe v_i, and then for every box $j \neq i$, resample $v'_j \sim F_j$ independently. If $i \in \mathrm{OPT}(v_i, \boldsymbol{v'}_{-i})$, then tell the OCRS algorithm that element i is active; otherwise tell it is not active. Select box i if and only if the OCRS algorithm selects element i.

The feasibility of the set of selected boxes is guaranteed by the feasibility of the OCRS algorithm's output. The contribution of box i to the prophet inequality algorithm is

$$\begin{aligned}
&\mathbf{E}_{v_i}\left[v_i\,\mathbf{Pr}_{\boldsymbol{v'}_{-i}}\left[i \text{ is selected } \mid i \text{ is active}\right]\cdot \mathbf{Pr}_{\boldsymbol{v'}_{-i}}\left[i \text{ is active}\right]\right]\\
\geq&\alpha\,\mathbf{E}_{v_i}\left[v_i\,\mathbf{Pr}_{\boldsymbol{v'}_{-i}}\left[i \in \mathrm{OPT}(v_i, \boldsymbol{v'}_{-i})\right]\right]\\
=&\alpha\,\mathbf{E}_{v_i}\left[v_i\,\mathbf{Pr}_{\boldsymbol{v}_{-i}}\left[i \in \mathrm{OPT}(\boldsymbol{v})\right]\right],
\end{aligned}$$

The claim is proved by summing over the boxes. □

Exercise 4. Show that an α-selectable OCRS in fact implies a prophet inequality algorithm that is α-competitive against the ex ante optimal,

In fact, the reverse of the exercise is true. We omit the proof here.

Theorem 7 (Lee and Singla, 2018)**.** *If for a feasibility system $\mathcal{F}$ there is an prophet inequality algorithm that is α-competitive against the ex ante optimal in a certain arrival model, then there is an OCRS algorithm that is α-selectable in the same arrival model.*

Corollary 1. *For any matroid, there is a $\frac{1}{2}$-selectable OCRS algorithm in adversarial arrival order.*

Proof of Theorem 7. Any valid, deterministic online contention resolution scheme is fully described by its behavior for every i and $S \subseteq [n]$: when element i arrives and turns out active, given that S is the set of elements already selected, does the algorithm select i or not? Therefore the set of deterministic schemes is finite, and we index them by λ. Given $\boldsymbol{x}$, denote by a_i^λ the probability with which the algorithm indexed by λ selects element i. Any ORCS is a randomization over deterministic OCRS. Let $(y_\lambda)_\lambda$ denote a distribution over deterministic OCRS; then the problem of finding an optimal OCRS is solved by the following linear program:

$$\begin{aligned}
&\max_{\gamma,\mathbf{y}} \gamma\\
\text{s.t. } &\sum_\lambda y_\lambda a_i^\lambda \geq x_i\gamma, \quad \forall i \in [n];\\
&\sum_\lambda y_\lambda = 1;\\
&y_\lambda \geq 0, \quad \forall\lambda.
\end{aligned}$$

The dual of the program is:

$$\begin{aligned} & \min_{\beta,\mathbf{z}} \beta \\ \text{s.t. } & \sum_i z_i a_i^\lambda \le \beta, \quad \forall \lambda; \\ & \sum_i x_i z_i = 1; \\ & z_i \ge 0, \quad \forall i. \end{aligned}$$

For any $z_1, \dots, z_n \ge 0$, think of each item i as a box in a prophet inequality problem, with value v_i that is z_i with probaiblity x_i, and 0 otherwise. The constraint $\sum_i x_i z_i = 1$ now normalizes to 1 the ex ante optimal of this prophet inequality problem. A key observation is that any deterministic prophet inequality algorithm in this context is translated to an OCRS: Element i is active if and only if v_i is z_i; without loss of generality, a prophet inequality algorithm only takes a box i when $v_i = z_i$, in which case we say the algorithm, as an OCRS, selects element i. Therefore the set of deterministic prophet inequality algorithms is the same as the set of OCRS. If there exists an prophet inequality that is α-competitive against the ex ante optimal, the objective of the dual program is at least α. By the strong duality theorem, the objective of the primal program is also at least α, and hence there is a α-selectable OCRS. □

Remark 1. This proof only shows the implication of good deterministic ex-ante prophet inequalities to avoid discussing strong duality in the presence of infinitely many variables/constraints. The argument generalizes to randomized algorithms; we omit the technical details here.

If an ex-ante prophet inequality algorithm runs in polynomial time, it can be used as a separation oracle to compute an $\alpha - \epsilon$-selectable OCRS in time polynomial in n and $\frac{1}{\epsilon}$. We also omit the details here.

5 OCRS and Pandora Box problem on Bipartite Matchings

We now consider one of the simplest feasibility system beyond matroids: bipartite matchings. Given a bipartite graph $G = (U, V, E)$, the universe is now E, and $\mathcal{F}$ is the set of all matchings in G.

5.1 OCRS on Bipartite Matchings

Theorem 8 (Ezra et al., 2020)**.** *There is a $\frac{1}{3}$-selectable OCRS algorithm for bipartite matchings, in the adversarial arrival order.*

Proof. We are given $\boldsymbol{x}$ in the matching polytope. The analysis resembles that of Example 2. When edge $e = (u, v)$ arrives, let y_e denote the probability with which neither u nor v has been matched by then. If e is active and both u and v are unmatched, the algorithm selects e with probability $1/3y_e$. We only need to show $y_e \ge 1/3$ for all e, as then the algorithm is well defined; each edge, when active, is then selected with probability precisely $1/3$.

Again we show this by induction. For the first edge, this is obviously true. Now consider an arbitrary edge $e = (u, v)$. Let $p(u, e)$ denote the set of edges incident to u that arrive before e;

and $p(v, e)$ the set of edges incident to v that arrive before e. By induction hypothesis, any edge $f \in p(u, e) \cup p(v, e)$ is selected with probability precisely $x_f/3$. Therefore by the union bound,

$$y_e \geq 1 - \sum_{f \in p(u,e)} \frac{x_f}{3} - \sum_{f \in p(v,e)} \frac{x_f}{3} \geq 1 - \frac{1}{3} - \frac{1}{3} = \frac{1}{3}.$$

The second inequality uses the fact that $\boldsymbol{x}$ is in the matching polytope. □

Corollary 2 (Gravin and Wang, 2019)**.** *There is a $\frac{1}{3}$-competitive prophet inequality algorithm for bipartite matchings, in the adversarial arrival order.*

In fact, better OCRS algorithms exist, even for general, non-bipartite graphs. Interested readers are referred to Ezra et al. (2020).

Theorem 9 (Ezra et al., 2020)**.** *There is a* 0.337*-selectable OCRS on matchings of a general graph (not necessarily bipartite) in the adversarial arrival order.*

5.2 Pandora Box Problem on Bipartite Matchings

From Exercise 2, we obtain a 2-approximation algorithm for the Pandora box problem for matchings in general graphs. To obtain better than 2-approximations, more non-trivial ideas are needed. We here describe a $\frac{e}{e-1}$-approximation due to Gamlath et al. (2019).

Theorem 10. *For arbitrary $\epsilon > 0$, there is a $\frac{e+\epsilon}{e-1}$-approximation algorithm, running in time* $\mathrm{poly}(|E|, \frac{1}{\epsilon})$*, for the Pandora box problem on bipartite matchings.*

References

Agrawal, S., Ding, Y., Saberi, A., and Ye, Y. (2012). Price of correlations in stochastic optimization. *Operations Research*, 60(1):150–162.

Ezra, T., Feldman, M., Gravin, N., and Tang, Z. G. (2020). Online stochastic max-weight matching: prophet inequality for vertex and edge arrival models. In *Proceedings of the 2020 ACM Conference on Economics and Computation, EC 2020, Budapest, Hungary, July 13-17, 2020*. ACM.

Feldman, M., Svensson, O., and Zenklusen, R. (2016). Online contention resolution schemes. In Krauthgamer, R., editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1014–1033. SIAM.

Gamlath, B., Kale, S., and Svensson, O. (2019). Beating greedy for stochastic bipartite matching. In Chan, T. M., editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2841–2854. SIAM.

Gravin, N. and Wang, H. (2019). Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In Karlin, A., Immorlica, N., and Johari, R., editors, *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 93–109. ACM.

Kleinberg, R. and Weinberg, S. M. (2019). Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games Econ. Behav.*, 113:97–115.

Lee, E. and Singla, S. (2018). Optimal online contention resolution schemes via ex-ante prophet inequalities. In Azar, Y., Bast, H., and Herman, G., editors, *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPIcs*, pages 57:1–57:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Singla, S. (2018). The price of information in combinatorial optimization. In Czumaj, A., editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2523–2532. SIAM.