Prophets and Secretaries

IPCO Summer School, 2017, Waterloo: Lecture #3
Anupam Gupta, Carnegie Mellon University

Prophet inequalities and Secretary problems are two classes of problems where online decision-making meets stochasticity: in the first set of problems the inputs are random variables, whereas in the second one the inputs are worst-case but revealed to the algorithm (a.k.a. the decision-maker) in random order. Here we survey some results, proofs, and techniques, and give some pointers to the rich body of work developing around them.

1 The Prophet Inequality

The problem setting: there are n random variables $X_1, X_2, \ldots, X_n$. We know their distributions up-front, but not their realizations. These realizations are revealed one-by-one (say in the order $1, 2, \ldots, n$). We want to give a strategy (which is a stopping rule) that, upon seeing the value X_i (and all the values before it) decides either to *choose* i, in which case we get reward X_i and the process stops. Or we can *pass*, in which case we move on to the next items, and are not allowed to come back to i ever again. We want to maximize our expected reward. If $X_{\max} := \max\{X_1, X_2, \ldots, X_n\}$, it is clear that our reward cannot exceed $\mathbb{E}[X_{\max}]$. But how close can we get?[1]

In fact, we may be off by a factor of almost two against this yardstick in some cases: suppose $X_1 = 1$ surely, whereas $X_2 = 1/\varepsilon$ with probability ε, and 0 otherwise. Any strategy either picks 1 or passes on it, and hence gets expected value 1, whereas $\mathbb{E}[X_{\max}] = (2 - \varepsilon)$. Surprisingly, this is the worst case.

Theorem 1.1 (Krengel, Sucheston, and Garling) *There is a strategy with expected reward* $\frac{1}{2}\mathbb{E}[X_{\max}]$.

We now give two proofs of this theorem. For the moment, let us ignore computational considerations, and just talk about the information theoretic issues.

Proof. (Due to Ester Samuel-Cahn.) Let τ be the median of the distribution of $X_{\max}$: i.e., $\Pr[X_{\max} \geq \tau] = 1/2$. (For simplicity we assume that there is no point mass at τ, the proof is easily extended to discrete distributions too.) Now the strategy is simple: pick the first X_i which exceeds τ. Clearly, we will pick an item with probability exactly $\frac{1}{2}$, but how does the expected reward compare with $\mathbb{E}[X_{\max}]$?

$$\begin{aligned}\mathbb{E}[X_{\max}] &\leq \tau + \mathbb{E}[(X_{\max} - \tau)^+]) \\ &\leq \tau + \mathbb{E}\left[\sum_{i=1}^{n} (X_i - \tau)^+\right].\end{aligned}$$

And what does the algorithm get?

$$ALG \geq \tau \cdot \Pr[X_{\max} \geq \tau] + \sum_{i=1}^{n} \mathbb{E}[(X_i - \tau)^+] \cdot \Pr[\bigwedge_{j \leq i} (X_j < \tau)]$$

[1]If we want to find the best strategy, and we know the order in which we are shown these r.v.s, there is dynamic programming algorithm. (Exercise!)

$$\geq \tau \cdot \Pr[X_{\max} \geq \tau] + \sum_{i=1}^{n} \mathbb{E}[(X_i - \tau)^+] \cdot \Pr[X_{\max} < \tau]$$

But both these probability values equal half, and hence $ALG \geq {}^1\!/\!_2\, \mathbb{E}[X_{\max}]$. ■

While a beautiful proof, it is somewhat mysterious, and difficult to generalize. Indeed, suppose we are allowed to choose up to k variables to maximize the sum of their realizations? The above proof seems difficult to generalize, but the following LP-based one will.

Proof. (Due to Chawla, Hartline, Malec, and Sivan; Alaei.[2]) Define p_i as the probability that element X_i takes on the highest value. Hence $\sum_i p_i = 1$. Moreover, suppose τ_i is such that $\Pr[X_i \geq \tau_i] = p_i$, i.e., the p_i^{th} percentile for X_i. Define

$$v_i(p_i) := \mathbb{E}[X_i \mid X_i \geq \tau_i]$$

as the value of X_i conditioned on it lying in the top p_i^{th} percentile. Clearly, $\mathbb{E}[X_{\max}] \leq \sum_i v_i(p_i) \cdot p_i$.

A Simpler, Weaker Bound. Here's a simple algorithm that gets value ${}^1\!/\!_4 \sum_i v_i(p_i) \cdot p_i \geq {}^1\!/\!_4\, \mathbb{E}[X_{\max}]$.

> If we have not chosen an item among $1, \ldots, i-1$, when looking at item i, pass with probability half without even looking at X_i, else accept it if $X_i \geq \tau_i$.

Say we "reach" item i if we've not picked an item before i. The expected value of the algorithm is

$$\begin{aligned} ALG &\geq \sum_{i=1}^{n} \Pr[\text{reach item } i] \cdot {}^1\!/\!_2 \cdot \Pr[X_i \geq \tau_i] \cdot \mathbb{E}[X_i \mid X_i \geq \tau_i] \\ &= \sum_{i=1}^{n} \Pr[\text{reach item } i] \cdot {}^1\!/\!_2 \cdot p_i \cdot v_i(p_i). \end{aligned} \tag{1.1}$$

Since we pick each item with probability $\frac{1}{2}p_i$, the expected number of items we choose is half. So, by Markov's inequality, the probability we pick no item at all is at least half. Hence, the probability we reach item i is at least one half too, the above expression is ${}^1\!/\!_4 \sum_i v_i(p_i) \cdot p_i$ as claimed.

The Bound of 2. To improve this algorithm, suppose we denote the probability of reaching item i by r_i, and suppose we reject item i outright with probability $1 - q_i$. Then (1.1) really shows that

$$ALG \geq \sum_{i=1}^{n} r_i \cdot q_i \cdot p_i \cdot v_i(p_i).$$

Above, we ensured that $q_i = r_i = \frac{1}{2}$, and hence lost a factor of $\frac{1}{4}$. But if we could ensure that $r_i \cdot q_i = 1/2$, we'd get the desired result of ${}^1\!/\!_2\mathbb{E}[X_{\max}]$. For the first item $r_1 = 1$ and hence we can set $q_1 = 1/2$. What about future items? Note that since that

$$r_{i+1} = r_i(1 - q_i \cdot p_i) \tag{1.2}$$

we can just set $q_{i+1} = \frac{1}{2r_{i+1}}$. A simple induction shows that $r_{i+1} \geq \frac{1}{2}$—indeed, sum up (1.2) to get $r_{i+1} = r_1 - \sum_{j \leq i} p_i/2$—so that $q_{i+1} \in [0, 1]$ and is well-defined. ■

[2]If you know of an earlier reference, let me know.

Such a result, that gives a stopping rule whose value is comparable to the $\mathbb{E}[X_{\max}]$ is called a *prophet inequality*, the idea being that one can come close to the performance of a prophet who is clairvoyant, can see the future. The result in Theorem 1.1 was proved by Krengel, Sucheston, and Garling [KS78]; several proofs have been given since. See the survey by Hill and Kertz [HK92] for classical work, and references in, e.g., [KW12, AKW14] for more recent work. Apart from being a useful algorithmic construct, the prophet inequality naturally fits into work on algorithmic auction design: suppose you know that n potential are interested in an item with valuations $X_1, \ldots, X_n$, and you want to sell to one person: how do you make sure your revenue is close to $\mathbb{E}[X_{\max}]$?

1.1 Discussion and Extensions

1.1.1 The Computational Aspect

If the distribution of the r.v.s X_i is explicitly given, the proofs above immediately give us algorithms. However, what if the variables X_i are given via a black-box that we can only access via samples?

The first proof merely relies on finding the median: in fact, finding an "approximate median" $\widehat{\tau}$ such that $\Pr[X_{\max} < \widehat{\tau}] \in (\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon)$ gives us expected reward $\frac{1}{2+\varepsilon/2}\mathbb{E}[X_{\max}]$. To do this, sample from $X_{\max}$ $O(\varepsilon^{-2} \log \delta^{-1})$ times (each sample to $X_{\max}$ requires one sample to each of the X_is) and take $\widehat{\tau}$ to be the sample median. A Hoeffding bound shows that $\widehat{\tau}$ is an "approximate median" with probability at least $1 - \delta$.

For the second proof, there are two ways of making it algorithmic. Firstly, if the quantities are polynomially bounded, estimate p_i and v_i by sampling. Alternatively, solve the convex program

$$\max \left\{ \sum_i y_i \cdot v_i(y_i) \mid \sum_i y_i = 1 \right\}$$

and use the y_i's from its solution in lieu of p_i's in the algorithm above.

Do we need such good approximations? Since getting samples from the distributions may be expensive, how many samples can we get away with? The paper of Azar, Kleinberg, and Weinberg [AKW14] shows how to get a constant fraction of $\mathbb{E}[X_{\max}]$ via taking just one sample from each of the X_is.

1.1.2 Extensions: Picking Multiple Items

What about the case where we are allowed to choose k variables from among the n? Proof #2 generalizes quite seamlessly. If p_i is the probability that X_i is among the top k values, we now have:

$$\textstyle\sum_i p_i = k. \tag{1.3}$$

The "upper bound" on our quantity of interest remains essentially unchanged:

$$\mathbb{E}[\text{sum of top } k \text{ r.v.s}] \leq \textstyle\sum_i v_i(p_i) \cdot p_i. \tag{1.4}$$

What about an algorithm to get value $1/4$ of the value in (1.4)? The same as above: reject each item outright with probability $1/2$, else pick i if $X_i \geq \tau_i$. Proof #2 goes through unchanged.

Better: For this case, we can do much better: a result of Alaei shows that one can get within $(1 - \frac{1}{\sqrt{k+3}})$ of the value in (1.4)—for $k = 1$, this nicely matches the $1/2$. One can, however, get a factor of $(1 - O(\sqrt{\frac{\log k}{k}}))$ using a simple concentration bound (as in [HKS07]).

Suppose we reduce the rejection probability to δ. Then the probability that we reach some item i without having picked k items already is lower-bounded by the probability that we pick at most k elements in the entire process. Since we reject items with probability δ, the expected number of elements we pick is $(1-\delta)k$. Hence, the probability that we pick less than k items is at least $1 - e^{-\Theta(\delta^2 k)}$, by a Hoeffding bound for sums of independent random variables. Now setting $\delta = O(\sqrt{\frac{\log k}{k}})$ ensures that the probability of reaching each item is at least $(1 - \frac{1}{k})$, and a argument similar to that in Proof #2 shows that

$$\begin{aligned} ALG &\geq \textstyle\sum_{i=1}^n \Pr[\text{reach item } i] \cdot \Pr[\text{not reject item } i] \cdot \Pr[X_i \geq \tau_i] \cdot \mathbb{E}[X_i \mid X_i \geq \tau_i] \\ &= \textstyle\sum_{i=1}^n (1 - \frac{1}{k}) \cdot (1 - O(\sqrt{\frac{\log k}{k}})) \cdot p_i \cdot v_i(p_i), \end{aligned}$$

which gives the claimed bound of $(1 - O(\sqrt{\frac{\log k}{k}}))$.

1.1.3 Extensions: Matroid Constraints

Suppose there is a matroid structure $\mathcal{M}$ with ground set $[n]$, and the set of random variables we choose must be independent in this matroid $\mathcal{M}$.[3] The value of the set is the sum of the values of items within it. (Hence, the case of at most k items above corresponds to the *uniform* matroid of rank k.) The goal is to make the expected value of the set picked by the algorithm close to the expected value of the max-weight independent set.

A result of Kleinberg and Weinberg [KW12] shows an algorithm to picks an independent set whose expected value is at least half the value of the max-weight independent set, thereby extending the original single-item prophet inequality seamlessly to all matroids. While their original proof uses a combinatorial idea, a LP-based proof was subsequently given by Feldman, Svensson, and Zenklusen [FSZ16]. The idea is again simple: find a solution y to the convex program

$$\begin{gathered} \textstyle\sum_i v_i(y_i) \cdot y_i. \\ y \in \text{the matroid polytope for } \mathcal{M} \end{gathered}$$

Now given a fractional point y in the matroid polytope, how to get an integer point (i.e., an independent set). For this they give an approach called an "online contention resolution" scheme that ensures that any item i is picked with probability at least $\Omega(y_i)$, much like in the single-item and k-item cases.

There are many other extensions to prophet inequalities: people have studied more general constraint sets, submodular valuations instead of just additive valuations, what if the order of items is not known, what if we are allowed to choose the order, etc. See papers on arXiv, or in the latest conferences for much more.

Exercises

1. Give a dynamic programming algorithm for the best strategy when we know the order in which r.v.s are revealed to us. (Footnote 1). Extend this to the case where you can pick k items.

   Open problem: is this "best strategy" problem computationally hard when we are given a general matroid constraint? Even a laminar matroid or graphical matroid?

2. If we can choose the order in which we see the items, show that we can get expected value $\geq (1 - 1/e)\mathbb{E}[X_{\max}]$. (Hint: use proof #2, but consider the elements in decreasing order of $v_i(p_i)$.)

   Open problem: can you beat $(1 - 1/e)\mathbb{E}[X_{\max}]$? A recent paper of [AEE+17] does so for i.i.d. X_is.

[3] Recall that a matroid $\mathcal{M} = (U, \mathcal{F})$ is a set U is a collection of subsets $\mathcal{F} \subseteq 2^U$ such that (1) $\emptyset \in \mathcal{F}$, (2) if $A \in \mathcal{F}$ and $B \subseteq A$ then $B \in \mathcal{F}$, and (3) if $A, B \in \mathcal{F}$ and $|A| < |B|$ then there exists $b \in B \setminus A$ such that $A \cup \{b\} \in \mathcal{F}$. Sets in $\mathcal{F}$ are called *independent sets*.

2 Secretary Problems

The problem setting: there are n items, each having some intrinsic non-negative value. For simplicity, assume the values are distinct, but we know nothing about their ranges. We know n, and nothing else. The items are presented to us one-by-one. Upon seeing an item, we can either pick it (in which case the process ends) or we can pass (but then this item is rejected and we cannot ever pick it again). The goal is to maximize the probability of picking the item with the largest value.

If an adversary chooses the order in which the items are presented, every deterministic strategy must fail. Suppose there are just two items, the first one with value 1. If the algorithm picks it, the adversary can send a second item with value 2, else it sends one with value $\frac{1}{2}$. Randomizing our algorithm can help, but we cannot do much better than $\frac{1}{n}$.

So the secretary problem asks: *what if the items are presented in uniformly random order?* For this setting, it seems somewhat surprising at first glance that one can prove the following theorem (knowing nothing other than n, and the promise of a uniformly random order):

Theorem 2.1 *There is a strategy to pick the best item with probability at least $\frac{1}{e}$.*

Proof. Here's a simple algorithm and proof showing a probability of $\frac{1}{4}$: ignore the first $n/2$ items, and then pick the next item that is better than all the ones seen so far. Note that this algorithm succeeds if the best item is in the second half of the items (which happens w.p. $1/2$) and the second-best item is in the first half (which, conditioned on the above event, happens w.p. $\geq 1/2$). Hence $\frac{1}{4}$. Of course, rejecting the first half of the items is not optimal, and there are other cases where the algorithm succeeds that this simple analysis does not account for, so let's be more careful.

First, our algorithm should never pick an element that is not the best so far. Moreover, if we define

$$f(i) = \Pr[i^{th} \text{ item is global best } \mid i^{th} \text{ is best so far}] = \frac{\Pr[i^{th} \text{ item is global best }]}{i^{th} \text{ is best so far}} = \frac{1/n}{1/i} = \frac{i}{n}.$$

$$g(i) = \Pr[\text{picking global best using optimal strategy from item } i \text{ onwards}],$$

then $g(i)$ is a non-increasing function (else we could use the strategy for $g(i+1)$ also for $g(i)$ by just ignoring item $i+1$) and $f(i)$ is increasing. So any optimal strategy should pass at times i where $f(i) < g(i+1)$ and pick at other times if we see an element that is best so far. I.e., a strategy of the type used in the simple proof of $1/4$ is optimal (with a different threshold)!

OK, so if we reject the first τ items, then the probability we succeed in picking the global best is

$$\sum_{t=\tau+1}^{n} \Pr[t^{th} \text{ item is global best }] \cdot \Pr[\text{ best of first } t-1 \text{ items in first } \tau \text{ positions }]$$
$$= \sum_{t=\tau+1}^{n} \frac{1}{n} \cdot \frac{\tau}{t-1} = \frac{\tau}{n}(H_{n-1} - H_{\tau-1}).$$

This is $g(\tau+1)$. So the first position where $f(\tau) = \frac{\tau}{n} \geq g(\tau+1)$ is given by the smallest τ such that $1 \geq H_{n-1} - H_{\tau-1}$, i.e., $\tau \approx n/e$ for large n, and hence $g(\tau+1) \approx 1/e$. ∎

In keeping with the theme of this lecture, we now give an alternate proof that uses a convex-programming view of the process. We will write down an LP that captures some properties of any feasible solution, optimize this LP and show a strategy whose success probability is comparable to the objective of this LP! The advantage of this approach is that it then extends to adding other constraints to the problem.

Proof. (Due to Niv Buchbinder, Kamal Jain, and Mohit Singh.) Let us fix an optimal strategy. By the first proof above, we know what it is, but let us ignore that for the time being. Let us just assume w.l.o.g. that it does not pick any item that is not the best so far (since such an item cannot be the global best).

Let p_i be the probability that this strategy picks an item at position i. Let q_i be the probability that we pick an item at position i, *conditioned on it being the best so far.* So $q_i = \frac{p_i}{1/i} = i \cdot p_i$.

Now, the probability of picking the best item is

$$\begin{aligned} &\sum_i \Pr[i^{th} \text{ position is global best and we pick it }] \\ &= \sum_i \Pr[i^{th} \text{ position is global best }] \cdot q_i = \sum_i \frac{1}{n} q_i = \sum_i \frac{i}{n} p_i. \end{aligned} \tag{2.5}$$

What are the constraints? Clearly $p_i \in [0,1]$. But also

$$\begin{aligned} p_i &= \Pr[\text{ pick item } i \mid i \text{ best so far}] \cdot \Pr[i \text{ best so far}] \\ &\leq \Pr[\text{ did not pick } 1, \ldots, i-1 \mid i \text{ best so far}] \cdot (1/i) \end{aligned} \tag{2.6}$$

But not picking the first $i-1$ items is independent of i being the best so far, so we get

$$p_i \leq \frac{1}{i}(1 - \sum_{j<i} p_j).$$

Hence, the success probability of any strategy (and hence of the optimal strategy) is upper-bounded by the following LP in variables p_i:

$$\begin{aligned} \max \sum_i \frac{i}{n} \cdot p_i & \\ i \cdot p_i &\leq 1 - \sum_{j<i} p_j \\ p_i &\in [0,1]. \end{aligned}$$

Now it can be checked that the solution $p_i = 0$ for $i \leq \tau$ and $p_i \frac{\tau}{n}(\frac{1}{i-1} - \frac{1}{i})$ for $\tau \leq i \leq n$ is a feasible solution, where τ is defined by the smallest value such that $H_{n-1} - H_{\tau-1} \leq 1$. (By duality, we can also show it is optimal!)

Finally we can get a stopping strategy whose success probability matches that of the LP. Indeed, solve the LP. Now, for the i^{th} position if we've not picked an item already and if this item is the best so far, pick it with probability $\frac{ip_i}{1-\sum_{j<i} p_j}$. By the LP constraint, this probability $\in [0,1]$. Moreover, removing the conditioning shows we pick an item at location i with probability p_i, and a calculation similar to the one above shows that our algorithm's success probability is $\sum_i ip_i/n$, the same as the LP. ■

2.1 Discussion and Extensions

2.1.1 Extension: Game-Theoretic Issues

Note that in the optimal strategy, we don't pick any items in the first n/e timesteps, and then we pick items with quite varying probabilities. If the items are people interviewing for a job, this gives them an incentive to not come early in the order. Suppose we insist that for each position i, the probability of picking the item at position i is the same. What can we do then?

Let's fix any such strategy, and write an LP capturing the success probabilities of this strategy with uniformity condition as a constraint. Suppose $p \leq 1/n$ is this uniform probability (over the randomness of the input sequence). Again, let q_i be the probability of picking an item at position i, conditioned on it being the best so far. Note that we may pick items even if they are not the best so far, just to satisfy the uniformity condition; hence instead of $q_i = i \cdot p$ as before, we have

$$q_i \leq ip.$$

Moreover, by the same argument as (2.6), we know that

$$q_i \leq 1 - (i-1)p.$$

And the strategy's success probability is again $\sum q_i/n$ using (2.5). So we can now solve the LP

$$\begin{aligned} \max \textstyle\sum_i \frac{1}{n} \cdot q_i \\ q_i &\leq 1 - (i-1) \cdot p \\ q_i &\leq i \cdot p \\ q_i &\in [0,1], p \geq 0 \end{aligned}$$

Now the Buchbinder et al. [BJS14] paper shows the optimal value of this LP is at least $1 - 1/\sqrt{2} \approx 0.29$; they also give a slightly more involved algorithm that achieves this success probability.

2.1.2 Extension: Multiple Items

Now back to having no restrictions on the item values. Suppose we want to pick k items, and want to maximize the expected sum of these k values. Suppose the set of the k largest values is $S^\star \subseteq [n]$, and their total value is $V^\star = \sum_{i \in S} v_i$. It is easy to get an algorithm with expected value $\Omega(V^\star)$. E.g., split the n items into k groups of n/k items, and run the single-item algorithm separately on each of these. (Why?) Or ignore the first half of the elements, look at the value $\hat{v}$ of the $(1-\epsilon)k/2^{th}$ highest value item in this set, and pick all items in the second half with values greater than $\hat{v}$. And indeed, ignoring half the items must lose a constant factor in expectation.

But here's an algorithm that gives value $V^\star(1-\delta)$ where $\delta \to 0$ as $k \to \infty$. We will set $\delta = O(k^{-1/3} \log k)$ and $\varepsilon = \delta/2$. Ignore the first δn items. (We expect $\delta k \approx k^{2/3}$ items from $S^\star$ fall in this ignored set.) Now look at the value $\hat{v}$ of the $(1-\varepsilon)\delta k^{th}$-highest valued item in this ignored set, and pick the first (at most) k elements with values greater than $\hat{v}$ along the remaining $(1-\delta)n$ elements.

> Why is this algorithm good? There are two failure modes: (i) if $v' = \min_{i \in S^\star} v_i$ be the lowest value item we care about, then we don't want $\hat{v} \leq v'$ else we may pick low valued items, and (ii) we want the number of items from $S^\star$ in the last $(1-\delta)n$ and greater than $\hat{v}$ to be close to k.
>
> Let's sketch why both bad events happen rarely. For event (i) to happen, fewer than $(1-\varepsilon)\delta k$ items from $S^\star$ fall in the first δn locations: i.e., their number is less than $(1-\varepsilon)$ times its expectation, which has probability $\exp(-\varepsilon^2 \delta k) = 1/\operatorname{poly}(k)$ by a Hoeffding bound. For event (ii) to happen, more than $(1-\varepsilon)\delta k$ of the top $(1-\delta)k$ items from $S^\star$ fall among the ignored items. This means their number exceeds $(1 + O(\varepsilon))$ times its expectation, which again has probability $\exp(-\varepsilon^2 \delta k) = 1/\operatorname{poly}(k)$.
>
> An aside: the standard concentration bounds we know are for sums of i.i.d. r.v.s whereas the random order model causes correlations. The easiest way to handle that is to ignore not the first δn items but a random number of items $\sim \operatorname{Bin}(n, \delta)$. Then each item has probability δ of being ignored, independent of others.

Is this tradeoff optimal? No. Kleinberg showed that one can get expected value $V^\star(1 - O(k^{-1/2}))$, and this is asymptotically optimal [Kle05].

In fact, one can extend this even further: a set of vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in [0,1]^m$ is fixed, along with values $v_1, v_2, \ldots, v_n$. These are initially unknown. Now they are revealed one-by-one to the algorithm in a uniformly random order. The algorithm, on seeing a vector and its value must decide to pick it or irrevocably reject it. It can pick as many vectors as it wants, subject to their sum being at most k

in each coordinate; the goal is to maximize the expected value of the picked vectors. The k-secretary case is the 1-dimensional case when each $\mathbf{a}_i = (1)$. Indeed, this is the problem of solving a packing linear program online, where the columns arrive in random order. A series of works have extended the k-secretary case to this online packing LP problem, getting values which are $(1 - O(\sqrt{(\log m)/k}))$ times the optimal value of the LP. See, e.g., papers by [KRTV14, GM16, AD15].

2.1.3 Extension: Matroids

One of the most tantalizing generalizations of the secretary problem is to matroids. Suppose the n elements form the ground set of a matroid, and the elements we pick must form an independent set in this matroid. Babioff, Immorlica, and Kleinberg [BIK07] asked: if the max-weight independent set has value V^*, can we get $\Omega(V^*)$ using an online algorithm? The current best algorithms, due to Lachish [Lac14], and Feldman, Svensson, and Zenklusen [FSZ15], achieve expected value $\Omega(V^*/\log\log k)$, where k is the rank of the matroid. Can we improve this further, say to a constant? A constant factor is known for many classes of matroids, like graphical matroids, laminar matroids, transversal matroids, and gammoids.

2.1.4 Other Random Arrival Models

One can consider other models for items arriving online: say a set of n items (and their values) is fixed by an adversary, and each timestep we see one of these items sampled uniformly *with replacement*. (The random order model is same, but without replacement.) This model, called the *i.i.d. model*, has been studied extensively—results in this model are often easier than in the random order model (due to lack of correlations). See, e.g., references in this monograph by Aranyak Mehta [Meh12].

Do we need the order of items to be uniformly random, or would weaker assumptions suffice? Kesselhiem, Kleinberg, and Niazadeh consider this question in a very nice paper and show that much less independence is enough for many of these results to hold [KKN15].

In general the random-order model is a clean way of modeling the fact that an online stream of data may not be adversarially ordered. Many papers in online algorithms have used this model to give better results than in the worst-case model: some of my favorite ones are this paper of Meyerson [Mey01] on facility location, and this paper of Bahmani, Chowdhury, and Goel on computing PageRank incrementally [BCG10].

Again, see online for many many papers related to the secretary problem: numerous models, different constraints on what sets of items you can pick, and how you measure the quality of the picked set. It's a very clean model, and can be used in many different settings.

Exercises

1. Give an algorithm for general matroids that finds an independent set with expected value at least an $O(1/(\log k))$-fraction of the max-value independent set.
2. Improve the above result to $O(1)$-fraction for graphic matroids.

A Concentration Bounds for Sums of Random Variables

The following bound is used several times in this short survey:

Theorem A.1 (Hoeffding's Bound) *Suppose $X = X_1 + X_2 + \ldots + X_n$, where the X_is are independent random variables taking on values in the interval $[0, 1]$. Let $\mu = E[X] = \sum_i E[X_i]$. Then*

$$\Pr[X > \mu + \lambda] \leq \exp\left(-\frac{\lambda^2}{2\mu + \lambda}\right) \tag{A.7}$$

$$\Pr[X < \mu - \lambda] \leq \exp\left(-\frac{\lambda^2}{3\mu}\right) \tag{A.8}$$

As an example, this shows that if an unbiased coin is tossed n times, with 99% chance the number of heads lies in $\frac{n}{2} \pm \Theta(\sqrt{n})$. Indeed, if $X_i = 1$ w.p. $\frac{1}{2}$ and 0 w.p. $\frac{1}{2}$, then $\mu = n/2$, and both $\Pr[X > \mu + c\sqrt{n}]$ and $\Pr[X < \mu - c\sqrt{n}]$ are bounded by 1/200 for some constant c.

Exercise: show that if n balls are thrown into n bins, the probability that bin 1 has load more than L is at most $e^{-\Omega(L)}$. Hence show that the max load in any bin is $O(\log n)$ with probability $1 - 1/\operatorname{poly}(n)$.

References

[AD15] Shipra Agrawal and Nikhil R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1405–1424, 2015.

[AEE+17] Melika Abolhassani, Soheil Ehsani, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Robert Kleinberg, and Brendan Lucier. Beating 1-1/e for ordered prophets. *CoRR*, abs/1704.05836, 2017.

[AKW14] Pablo D. Azar, Robert Kleinberg, and S. Matthew Weinberg. Prophet inequalities with limited information. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1358–1377. ACM, New York, 2014.

[BCG10] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. Fast incremental and personalized pagerank. *PVLDB*, 4(3):173–184, 2010.

[BIK07] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA '07*, pages 434–443, 2007.

[BJS14] Niv Buchbinder, Kamal Jain, and Mohit Singh. Secretary problems via linear programming. *Math. Oper. Res.*, 39(1):190–206, 2014.

[FSZ15] Moran Feldman, Ola Svensson, and Rico Zenklusen. A simple $O(\log\log(\text{rank}))$-competitive algorithm for the matroid secretary problem. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1189–1201, 2015.

[FSZ16] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1014–1033, 2016.

[GM16] Anupam Gupta and Marco Molinaro. How the experts algorithm can help solve lps online. *Math. Oper. Res.*, 41(4):1404–1431, 2016.

[HK92] Theodore P. Hill and Robert P. Kertz. A survey of prophet inequalities in optimal stopping theory. In *Strategies for sequential search and selection in real time (Amherst, MA, 1990)*, volume 125 of *Contemp. Math.*, pages 191–207. Amer. Math. Soc., Providence, RI, 1992.

[HKS07] Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 58–65, 2007.

[KKN15] Thomas Kesselheim, Robert D. Kleinberg, and Rad Niazadeh. Secretary problems with non-uniform arrival order. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 879–888, 2015.

[Kle05] Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 630–631, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[KRTV14] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. Primal beats dual on online packing LPs in the random-order model. pages 303–312, 2014.

[KS78] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. In *Probability on Banach spaces*, volume 4 of *Adv. Probab. Related Topics*, pages 197–266. Dekker, New York, 1978.

[KW12] Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 123–135. ACM, New York, 2012.

[Lac14] Oded Lachish. O(log log rank) competitive ratio for the matroid secretary problem. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 326–335, 2014.

[Meh12] Aranyak Mehta. Online matching and ad allocation. *Found. Trends Theor. Comput. Sci.*, 8(4):265–368, 2012.

[Mey01] Adam Meyerson. Online facility location. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 426–431, 2001.