Online Algorithms for Stochastic Adversaries

Online Algorithms

Summer 2014

Introductory Example

- Suppose you are trying to hire a new employee (e.g., a secretary) from a pool of applicants.
- The number n of applicants is known, you try to hire the best one.
- The quality of an applicant is unknown until you interview him.
- After each single interview you have to make an immediate decision whether to hire or reject the applicant.

There is a simple sample-and-learn strategy that allows to hire the best applicant with constant probability.

Sample-and-Learn

Sample-and-Learn

- Consider applicants in random order, interview and reject first r^* ones.
- Continue interviewing randomly and hire the first applicant that is better than the best one so far.
- If you reach the end of the pool, hire the last applicant.

With appropriate choice of r^*, the best applicant gets hired with probability $1/e \approx 0.37$.

Similar problems arise in online markets, where buyers arrive online in a market and sellers have to decide instantly about whether to sell items or not. Buyers and sellers can have combinatorial constraints to buy or sell only certain subsets of items (e.g., DVD collections, movie tickets, slots for display-ads, etc.)

We consider a general framework for combinatorial online allocation problems.

Generalized Secretary Problems

The Items and Values

- There are *m elements* from a ground set $\mathcal{R}$.
- Each element $x \in \mathcal{R}$ has *value* $w_x \geq 0$.
- There is a collection $\mathcal{I} \subseteq 2^{\mathcal{R}}$ of *feasible sets*.
- $\mathcal{I}$ is closed under containment: If $I \in \mathcal{I}$, then $J \in \mathcal{I}$ for all $J \subseteq I$.

Arrival and Selection

- Structure of $\mathcal{R}$ and $\mathcal{I}$ are known in advance.
- Elements arrive in random order, revealed with their value upon arrival.
- *Online algorithm* $\mathcal{A}$ decides to select or reject an element.
- An element must be selected or rejected before seeing the next one.
- The decision to select or reject is irreversible.
- The set of selected items must belong to $\mathcal{I}$ at all times.

Solution Quality

Competitiveness

- Algorithm $\mathcal{A}$ tries to maximize value of set S of selected items $w(S) = \sum_{x \in S} w_x$.
- $\mathcal{A}$ is called α-*competitive* if

$$\mathbb{E}\left[w(S)\right] \quad \geq \quad 1/\alpha \cdot w(S^*)$$

  with S^* an optimum set from $\mathcal{I}$ that maximizes $w(S)$.

- The expectation is taken over random order arrival and internal randomization of $\mathcal{A}$.

Examples

Secretary Problem

- $\mathcal{I}$ is the set of all singleton sets.

k-Choice Secretary

- $\mathcal{I}$ is the set of all sets S with $|S| \leq k$.

Matroid Secretary

- $\mathcal{I}$ is the set of all independent sets from a matroid.

Knapsack Secretary

- Each item $x \in \mathcal{R}$ has a value w_x and a size s_x.
- The knapsack has a capacity of 1.
- $\mathcal{I}$ is the set of all sets S with $\sum_{x \in S} s_x \leq 1$.

and many more...

Matroids

Definition (Matroid)

A tuple $M = (\mathcal{R}, \mathcal{I})$ is a matroid if $\mathcal{R} = \{1, \ldots, m\}$ is a finite set of elements and $\mathcal{I}$ is a nonempty family of subsets of $\mathcal{R}$ such that

- if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$, and
- if $I, J \in \mathcal{I}$ and $|J| < |I|$, then there exists an $i \in I \backslash J$ with $J \cup \{i\} \in \mathcal{I}$.

Notation

- a set $I \in \mathcal{I}$ is called independent set
- a maximal independent set $B \in \mathcal{I}$ is called a basis
- the cardinality of the bases is equal and called rank, $rk(M)$

Matroids

Definition (Weighted Matroid)

- A matroid with a weight function $w : \mathcal{R} \to \mathbb{N}$ is called *weighted*.
- The weight of an independent set I is $w(I) = \sum_{r \in I} w(r)$.
- An optimal basis is a basis of minimum weight.

Proposition *(swap property)*

Let B be a basis and B^* an optimal basis. Then there exists a sequence $(r_1, r_1^*), \ldots, (r_k, r_k^*) \in B \times B^*$, $0 \leq k \leq m$, such that, for $0 \leq i \leq k$, $B_i = B \cup \{r_1^*, \ldots, r_i^*\} \setminus \{r_1, \ldots, r_i\}$ is a basis, $B^* = B_k$ and $w(B_i) \leq w(B_{i-1})$, for $1 \leq i \leq k$.

A Logarithmic Algorithm

Sample-and-Learn with Threshold

- Reject the first $m/2$ elements, denote this set by Y.
- Pick $j \in \{0, 1, 2, \ldots, \lceil \log k \rceil\}$ uniformly at random
- Set threshold $\tau = \max_{x \in Y} w_x / 2^j$, initialize $S = \emptyset$
- At time $t = m/2 + 1, \ldots, m$, element x_t is observed.
- If $w_{x_t} \geq \tau$ and $x_t \cup S$ is an independent set, then add x_t to S.

Theorem (Babaioff, Immorlica, Kleinberg 2007)

Sample-and-Learn with Threshold is $O(\log k)$-competitive for any matroid domain where k is the rank of the matroid.

Logarithmic Factor

Proof: We first restrict attention to elements with significant value.

- S^* consists of elements $x_1, \ldots, x_k$ with value $w_1 \geq \ldots \geq w_k$.
- Let q be such that $w_q \geq w_1/k$ and either $q = k$ or $w_{q+1} < w_1/k$.
- Observe that $\sum_{i=q+1}^{k} w_i < w_1$, so $\sum_{i=1}^{q} w_i \geq w(S^*)/2$.

We analyze the algorithm based on value classes derived from S^*.

- $n_i(T)$ – number of elements of $T \subset \mathcal{R}$ with value at least w_i.
- $m_i(T)$ – number of elements of $T \subset \mathcal{R}$ with value at least $w_i/2$.
- Observe that

$$\sum_{i=1}^{q} w_i = \left[\sum_{i=1}^{q-1} (w_i - w_{i+1}) n_i(S^*)\right] + w_q n_q(S^*).$$

- For any output S we have

$$w(S) \geq \frac{1}{2} \cdot \left[\sum_{i=1}^{q-1} (w_i - w_{i+1}) m_i(S)\right] + \frac{1}{2} \cdot w_q m_q(S).$$

Proof

Lemma

For all $i = 1, \ldots, q$ *we have*

$$n_i(S^*) \quad \leq \quad 8(\lceil \log k \rceil + 1) \cdot \mathbb{E}\left[m_i(S)\right].$$

Using this lemma allows to show the theorem:

$$\begin{aligned}
\mathbb{E}\left[w(S)\right] &\geq \frac{1}{2} \cdot \left[\sum_{i=1}^{q-1} (w_i - w_{i+1}) \cdot \mathbb{E}\left[m_i(S)\right]\right] + \frac{1}{2} \cdot w_q \cdot \mathbb{E}\left[m_q(S)\right] \\
&\geq \frac{1}{16(\lceil \log k \rceil + 1)} \left[\sum_{i=1}^{q-1} (w_i - w_{i+1}) n_i(S^*)\right] + \frac{1}{16(\lceil \log k \rceil + 1)} w_q n_q(S^*) \\
&= \frac{1}{16(\lceil \log k \rceil + 1)} \sum_{i=1}^{q} w_i \\
&\geq \frac{1}{32(\lceil \log k \rceil + 1)} \cdot w(S^*)
\end{aligned}$$

Proof of Lemma

We show the lemma for each i individually. The case $i = 1$ is left as an exercise. Assume $i > 1$.

- Denote by x^* the element with maximum value.
- We condition on the event E that $x^* \in Y$ and j is such that $w_i \geq w_{x^*}/2^j \geq w_i/2$.

- We can compute S^* with a greedy algorithm, so $w_q \geq w_1/k \geq w_{x^*}/2^{\lceil \log k \rceil}$. Hence, there exists a suitable j for every v_i with $i \leq q$.

- The algorithm selects this j with probability $1/(\lceil \log k \rceil + 1)$.
- The combined probability of event E is, thus, $1/2(\lceil \log k \rceil + 1)$.

Proof of Lemma

We show a bound conditioned on event E.

- There is independent set $S' = \{x_1, \ldots, x_i\}$ of at least i elements with value at least w_i that exceed the threshold $\tau = w_{x^*}/2^j$.
- As $x^* = x_1$ is in Y by assumption, in expectation, at least $(i-1)/2 \geq i/4$ elements of S' appear for selection in the second half.
- By the exchange property, the expected size of S conditioned on E is at least $i/4$.
- As $\tau \geq v_i/2$ and every element chosen has value at least τ, we have

$$\mathbb{E}[m_i(S) \mid E] \quad \geq \quad i/4 \quad = \quad n_i(S^*)/4.$$

Removing the conditioning on E yields the lemma. □

Is there a constant-competitive algorithm?

Matroid Secretary Conjecture

For every matroid domain there is an algorithm that is ...

Weak: ... constant-competitive.

Strong: ... e-competitive.

The conjecture has been proved in some form for many special classes of matroids in recent years, but in general it is still open. The currently best algorithm by Chakraborty and Lachish (2012) for general matroids is $O(\sqrt{\log k})$-competitive.

For a special case of matroids we here prove the weak conjecture.

Graphic Matroids

For graphic matroids there is an undirected graph $G = (V, E)$ and we can choose any cycle-free edge set $E' \subseteq E$.

Parallel Sample-and-Learn

- Fix arbitrary ordering $v_1, v_2, \ldots, v_n$ for V
- Choose $X \in \{0, 1\}$ uniformly at random
- If $X = 0/1$, orient every $e \in E$ towards node with higher/lower index
- For each v in parallel, run Sample-and-Learn on edges oriented towards v
- Output S as union of edges chosen by the Sample-and-Learn algorithms

Theorem (Korula, Pal 2009)

Parallel Sample-and-Learn is 2e-competitive for graphic matroid domains.

Proof

Proof:

- The orientation implies that G becomes a directed acyclic graph. Hence, if every vertex picks any incoming edge, we create no cycles and S is feasible.
- It remains to bound the expected value of S.
- Let G_X be the oriented graph for $X \in \{0, 1\}$.
- Let $h_X(v)$ be an incoming edge of v in G_X with maximum value
- Let $S_X = \{h_X(v) \mid v \in V\}$, and S^* an optimum forest in G.

Proposition

$$w(S^*) \quad \leq \quad \sum_{v \in V} w_{h_0(v)} + w_{h_1(v)} \quad = \quad w(S_0) + w(S_1) \quad .$$

Proof

Conditioned on the coin flip X, each vertex recovers an expected value of $1/e$ of the incoming edge of maximum value. Hence, in total for both $x = 0, 1$

$$\mathbb{E}\left[w(S) \mid X = x\right] \quad \geq \quad 1/e \cdot w(S_x) \ .$$

Using the previous proposition, we see that

$$\begin{aligned}\mathbb{E}\left[w(S)\right] &= \frac{1}{2} \cdot \left(\mathbb{E}\left[w(S) \mid X = 0\right] + \mathbb{E}\left[w(S) \mid X = 1\right]\right) \\ &\geq \frac{1}{2e} \cdot \left(w(S_0) + w(S_1)\right) \\ &\geq \frac{1}{2e} \cdot w(S^*)\end{aligned}$$

which proves the theorem. □

Bipartite Matching with Random Arrivals

There is an underlying bipartite graph $G = (L, R, E)$ with edge values $w_e \geq 0$.

- Set L of customers, arrive in random order
- Set R of goods for sale, given in advance
- Value w_e of edge $e = (\ell, r)$ is value of customer ℓ for good r.
- Each customer has unit demand, i.e., strives to obtain at most one good.
- Upon arrival, each $\ell \in L$ reveals values of all incident edges
- Goal: Find a matching of goods to customers with maximum value.

This scenario slightly extends the framework, in which we would only pick the set of *matched vertices* from L and construct the matching in hindsight. Instead, we strive to allocate goods to customers upon arrival, i.e., we want to adaptively construct the set of *matching edges*. Note that matching edges do not arrive in fully random order, they arrive batched with their common endpoint in L.

Sample-and-Price

Sample-and-Price

- $k = Binom(|L|, p)$, $S = \emptyset$.
- Reject the first k vertices of L, denote this set by L'
- Consider edges in $E \cap (L' \times R)$ in non-increasing order of value and greedily construct a matching for L', denote this by M_1
- For each $r \in R$ let $price(r)$ be the value of the edge incident in M_1
- For each $t = k+1, \ldots, |L|$, denote $\ell_t \in L$ the vertex arriving in step t
- Let $e = (\ell_t, r)$ be the highest-value edge with $w_e \geq price(r)$
- If $S \cup e$ is a matching, add e to S

Theorem (Korula, Pal 2009)

With $p = 1/2$, Sample-and-Price is 8-competitive for matching domains.

We will only show a factor of 16 here.

Reformulation

To analyze Sample-and-Price, we consider a reformulation that works with all elements of L' being present in the beginning. Obviously, in expectation, it produces the same output S. It just separates the construction of L' and maintains M_2, the set of all candidate edges.

Sample-and-Permute

- Initialize $L' = \emptyset$, $S = \emptyset$, $M_1, M_2 = \emptyset$.
- For each $\ell \in L$, flip a coin with prob. p of heads. If heads, add ℓ to L'.
- Consider edges in $E \cap L' \times R$ in non-increasing order of value
- Greedily construct a matching for L', denote this by M_1
- For each $r \in R$ let $price(r)$ be the value of the edge incident in M_1
- For each $\ell \in L - L'$ in random order:
- Let $e = (\ell, r)$ be the highest-value edge with $w_e \geq price(r)$
- Add e to M_2
- If $S \cup e$ is a matching, add e to S

Reformulation II

We continue to reformulate the algorithm to make its analysis simpler. In particular, we contract the consideration of edges into a single loop.

Simulate

- Sort all edges in E in non-increasing order of value
- Initialize $M_1, M_2 = \emptyset$, mark each $\ell \in L$ unassigned
- For each edge $e = (\ell, r) \in E$ in sorted order:
- If (ℓ unassigned) and ($M_1 \cup e$ is matching), then:
- Mark ℓ assigned
- Flip a coin with prob. p of heads
- If heads, then $M_1 = M \cup e$; else $M_2 = M_2 \cup e$.
- S $= M_2$
- For each $r \in R$:
- If r has degree > 1 in S, remove all edges incident to r from S

Observations

We note the following facts that are the basis for our analysis.

- We make (at most) one coin flip for each $\ell \in L$ in both Simulate and Sample-and-Permute.
- If the flips turn out the same, sets M_1 and M_2 are similar in both algorithms. (Why?)
- To build S, both algorithms keep edges which have a unique endpoint in R. Simulate drops all edges from M_2 with a common endpoint, Sample-and-Permute keeps some of these edges.
- Thus, for the matchings S_{SPr} of Sample-and-Price, S_{SP} of Sample-and-Permute, and S_{Sim} of Simulate we have

$$\mathbb{E}[w(S_{SPr})] = \mathbb{E}[w(S_{SP})] \geq \mathbb{E}[w(S_{Sim})] .$$

Analysis of Simulate

Greedy matching in decreasing order of edge-weight yields a stable matching. Hence, if we greedily match the whole instance, such a matching S_g has $w(S_g) \geq w(S^*)/2$. In fact, we get a similar guarantee for greedily matching only the random subset L' for which the coin flips turn up heads.

Lemma

$$\mathbb{E}\left[w(M_1)\right] \quad \geq \quad p \cdot w(S^*)/2 \ .$$

Every $e \in M_2$ could have gone into M_1 as well at the point of consideration. It is easy to show $\mathbb{E}\left[w(M_2)\right] \geq (1-p)/p \cdot \mathbb{E}\left[w(M_1)\right]$ and, hence,

Lemma

$$\mathbb{E}\left[w(M_2)\right] \quad \geq \quad (1-p) \cdot w(S^*)/2 \ .$$

Proof of Key Lemma

The following is our key lemma that relates the output of Simulate to S^*.

Lemma

For the expected value of S_{Sim} we have

$$\mathbb{E}\left[w(S_{Sim})\right] \quad \geq \quad \frac{p^2(1-p)}{2} \cdot w(S^*) \ .$$

With $p = 1/2$ we obtain a competitive ratio of 16. With a more complicated analysis that applies directly to Sample-and-Permute, one can show a factor of $p(1-p)/2$ resulting in a competitive ratio of 8. Here, however, we stick to the simpler analysis of Simulate.

Proof of Key Lemma

Proof of Lemma:

- For $v \in R$, "revenue" earned by v is sum of values of edges incident in M_2.
- Denote the revenue by $Rev_2(v)$ [Note: $\sum_{v \in R} Rev_2(v) = w(M_2)$.]
- Assume e is first edge incident to v added to M_2. We denote by $\mathbb{E}\left[Rev_2(v) \mid e\right]$ the expected revenue of v in this case.

It is easy to see that $\mathbb{E}\left[Rev_2(v) \mid e\right] \leq w_e/p$:

- If the next edge incident to v is added to M_1, v stops earning revenue.
- Each addition to M_1 happens with probability p.
- If earning stops after i edges, v has earned w_e (first edge by assumption) and at most $(i-1) \cdot w_e$ from later edges.
- This happens with probability $(1-p)^{i-1} \cdot p$. Hence,

$$\mathbb{E}\left[Rev_2(v) \mid e\right] \leq w_e \cdot \sum_{i=1}^{\infty} i \cdot (1-p)^{i-1} \cdot p = w_e/p$$

Proof of Key Lemma

- Similarly, denote by $Rev_3(v)$ the revenue in S, i.e., the value of (at most) one incident edge in the final S_{Sim}.
- Again, let $\mathbb{E}[Rev_3(v) \mid e]$ be the expected revenue conditioned on the case that e is the first edge incident to v added to M_2.
- With probability p, the next edge incident to v is added to M_1.
- Then, no more edges incident to v are added and v has degree 1 in M_2.
- Thus,

$$\mathbb{E}[Rev_3(v) \mid e] \quad \geq \quad p \cdot w_e \quad \geq \quad p \cdot p \cdot \mathbb{E}[Rev_2(v) \mid e]$$

As the last bound holds for all vertices v and all incident edges e, linearity of expectation implies

$$\mathbb{E}[w(S_{Sim})] \quad \geq \quad p^2 \mathbb{E}[w(M_2)] \quad \geq \quad \frac{p^2(1-p)}{2} \cdot w(S^*) \ .$$

□

Set Packing Secretary

The algorithm above is quite flexible and can be used to solve more general variants of assignment problems. Consider the following version of set packing.

- Set L of customers, arrive in random order
- Set R of goods for sale, given in advance
- w_e of hyperedge $e = (\ell, I)$ is value of customer ℓ for set of goods $I \subseteq R$.
- Each customer strives to obtain one subset of goods.
- Upon arrival, each $\ell \in L$ reveals values of all incident hyperedges
- Goal: Construct an assignment of goods to customers with maximum value.

Intuitively, customers represent (collections of) sets or hyperedges, and we strive to obtain a set packing of maximum value by choosing a subset of sets that are pairwise disjoint.

Sample-and-Price for Set Packing

Sample-and-Price

- $k = Binom(|L|, p)$, $S = \emptyset$.
- Reject the first k vertices of L, denote this set by L'
- Consider hyperedges incident to L' in non-increasing order of value and greedily construct an assignment for L', denote this by M_1
- For $r \in R$ let $price(r)$ be the value of the hyperedge incident in M_1
- For each $t = k + 1, \ldots, |L|$, denote $\ell_t \in L$ the vertex arriving in step t
- Let $e = (\ell_t, I)$ be the highest-value edge s.t. for all $r \in I$, $w_e \geq price(r)$
- If e is disjoint from S, add e to S

Theorem (Korula, Pal 2009)

Let d be the size of the largest hyperedge. With $p = 1 - 1/(2d)$, Sample-and-Price is $O(d^2)$-competitive for set packing domains.

Prophets and Secretaries

- Suppose you are trying to hire a new employee from a pool of applicants.
- The number n of applicants is known, applicant i has random value X_i.
- X_i is drawn independently at random from known distribution D_i.
- The true value of X_i becomes known only after an interview.
- Applicants might arrive in adversarial order for interview.
- After each single interview you have to make an immediate decision whether to hire or reject the applicant.

If the expected optimum $\mathbb{E}[\max_i X_i] < \infty$, there is a simple stopping criterion to recover at least half of the expected optimum.

An optimal algorithm

Optimal-Stop

- Let $T = \mathbb{E}[\max_j X_j]/2$ be half of the expected optimum.
- If current candidate i has $X_i \geq T$, select i; otherwise reject i

An algorithm is α-competitive if it computes a solution S that in expectation recovers an α fraction of the *expected* optimum:

$$\mathbb{E}[w(S)] \quad \geq \quad 1/\alpha \cdot \mathbb{E}\left[\max_S w(S)\right]$$

Theorem (Krengel, Sucheston 1978)

Optimal-Stop is 2-competitive and this ratio is optimal.

We prove the upper bound, the lower bound is left as an exercise.

Analysis

The algorithm stops at the first candidate τ that has value at least T. It might reject everyone, but nevertheless obtains an expected value of $\mathbb{E}[X_\tau] \geq T$.

- With probability $p = \Pr[\max_i X_i \geq T]$ at least one candidate has value T, so with probability $(1-p)$ the algorithm accepts nobody.
- Suppose the algorithm processes candidate i. With probability at least $(1-p)$ it has accepted nobody so far.
- Also, with probability $\Pr[X_i > x]$ candidate i has $X_i > x \geq T$. Then the algorithm accepts i.
- Hence, the probability that the algorithm accepts i and $X_i > x$ is at least $(1-p) \cdot \Pr[X_i > x]$.
- For every $x \geq T$ we have at acceptance time τ:

$$\Pr[X_\tau > x] \quad \geq \quad (1-p)\sum_{i=1}^{n} \Pr[X_i > x] \quad .$$

Analysis

- The union bound states that

$$\sum_{i=1}^{n} \Pr\left[X_i > x\right] \quad \geq \quad \Pr\left[\max_i X_i > x\right] \quad ,$$

and hence

$$\Pr\left[X_\tau > x\right] \quad \geq \quad (1-p) \cdot \Pr\left[\max_i X_i > x\right] \quad .$$

- Observe that by definition

$$\begin{aligned} 2T = \mathbb{E}\left[\max_i X_i\right] &= \int_{x=0}^{\infty} \Pr\left[\max_i X_i > x\right] dx \\ &= \int_{x=0}^{T} \Pr\left[\max_i X_i > x\right] dx + \int_{x=T}^{\infty} \Pr\left[\max_i X_i > x\right] dx \end{aligned}$$

- The first term is at most T, hence the latter must be at least T.

Analysis

- Hence, we can bound as follows:

$$\begin{aligned}\mathbb{E}[X_\tau] &= \int_{x=0}^{T} \Pr[X_\tau > x]\, dx + \int_{x=T}^{\infty} \Pr[X_\tau > x]\, dx \\ &\geq pT + (1-p)\int_{x=T}^{\infty} \Pr\left[\max_i X_i > x\right] dx \\ &\geq pT + (1-p)T = T \\ &= \mathbb{E}\left[\max_i X_i\right]/2 \ .\end{aligned}$$

- This proves the theorem. □

The inequality $\mathbb{E}[X_\tau] \geq \mathbb{E}[\max_i X_i]/2$ is called *prophet inequality*, as it allows to bound the obtained value against an optimal prophet that knows the input.

Secretary Problem

Matroid Secretary

Matching Secretary

Prophet Inequalities

Matroid Prophets

Matroid Prophets

The scenario extends to packing domains in the obvious way. Let us consider the following matroid prophet domain:

- Underlying matroid $M = (\mathcal{R}, \mathcal{I})$ is known
- Each $i \in R$ has value w_i drawn independently from known distribution D_i
- Elements arrive in adversarial order, must be selected or rejected.
- Goal: Build an independent set $S \in \mathcal{I}$ with maximum value.

We present a monotone algorithm that – instead of a global threshold – uses deterministic thresholds T_i for each element i. We assume that $T_i = \infty$ if i cannot be added to the currently selected independent set. The algorithm accepts every i for which $w_i \geq T_i$.

Such an algorithm crucially relies on suitable thresholds T_i that (1) allow to accept elements with a large enough value and (2) do not reject too many valuable elements. The following notion of α-balancedness is a formalization of this condition.

α-balanced Thresholds

- For each $i \in R$, w_i is the input weight, and w'_i is a sample weight. Both are drawn independently at random from D_i.
- The input sequence is $\sigma = (i_1, w_{i_1}), \ldots, (i_n, w_{i_n})$. In our definition below, we will fix the sequence (and thereby the w_i's) and demand a condition for every such sequence.
- Selected set of the algorithm is denoted $S = S(\sigma)$.
- Optimal basis for w' is denoted B.
- By matroid exchange axiom, there is at least one partition of B into B_c and B_r such that $S \cup B_r$ is a basis of M.
- Of all these partitions, let $(B_c(S), B_r(S))$ be one that maximizes $w'(B_r)$.

Definition

Definition

For $\alpha > 0$, a monotone algorithm has α-balanced thresholds if for every input sequence σ and V disjoint from $S = S(\sigma)$ with $S \cup V \in \mathcal{I}$, the deterministic thresholds $T_i = T_i(\sigma)$ are such that

$$\sum_{i \in S} T_i \geq \left(\frac{1}{\alpha}\right) \cdot \mathbb{E}\left[w'(B_c(S))\right] \qquad (1)$$

$$\sum_{i \in V} T_i \leq \left(1 - \frac{1}{\alpha}\right) \cdot \mathbb{E}\left[w'(B_r(S))\right] \qquad (2)$$

where the expectation is over the random choice of w'.

Proposition

If a monotone algorithm has α-balanced thresholds, then it is α-competitive for matroid domains against online weight-adaptive adversaries.

Proof

We denote $(x)^+ = \max\{x, 0\}$ and show the proposition with three inequalities:

$$\mathbb{E}\left[\sum_{i \in S} T_i\right] \geq \frac{1}{\alpha} \cdot \mathbb{E}\left[w'(B_c(S))\right] \quad (3)$$

$$\mathbb{E}\left[\sum_{i \in S} (w_i - T_i)^+\right] \geq \mathbb{E}\left[\sum_{i \in B_r(S)} (w'_i - T_i)^+\right] \quad (4)$$

$$\mathbb{E}\left[\sum_{i \in B_r(S)} (w'_i - T_i)^+\right] \geq \frac{1}{\alpha} \cdot \mathbb{E}\left[w'(B_r(S))\right] \quad (5)$$

For all $i \in S$ we have $w_i \geq T_i$ and $T_i + (w_i - T_i)^+ = w_i$. Summing (3) and (4) and applying (5) yields:

$$\mathbb{E}\left[w(S)\right] \geq \frac{1}{\alpha}\mathbb{E}\left[w'(B_c(S))\right] + \frac{1}{\alpha}\mathbb{E}\left[w'(B_r(S))\right] = \frac{1}{\alpha} \cdot \mathbb{E}\left[\max_S w(S)\right]$$

The latter is due to $B = B_c(S) \cup B_r(S)$ being an optimal basis for w', which has the same distribution as w.

Three Inequalities

Inequality (3) follows by (1) in the definition. For (4) we observe that

- The algorithm accepts every i with $w_i \geq T_i$, so

$$\sum_{i \in S} (w_i - T_i)^+ = \sum_{i=1}^{n} (w_i - T_i)^+ \ .$$

- Weight-adaptive adversaries do not learn w_i before choosing to reveal i, so T_i depends only on the sequence up to element i.
- Hence, the random variables w_i, w_i', and T_i are independent.
- w and w' are identically distributed, so

$$\mathbb{E}\left[\sum_{i=1}^{n} (w_i - T_i)^+\right] = \mathbb{E}\left[\sum_{i=1}^{n} (w_i' - T_i)^+\right] \geq \mathbb{E}\left[\sum_{i \in B_r(S)} (w_i' - T_i)^+\right]$$

and (4) follows.

Three Inequalities

Finally, to prove (5) we use property (2) from the definition of α-balanced thresholds with $V = B_r(S)$ and calculate:

$$\mathbb{E}\left[\sum_{i \in B_r(S)} w_i'\right] \leq \mathbb{E}\left[\sum_{i \in B_r(S)} T_i\right] + \mathbb{E}\left[\sum_{i \in B_r(S)} (w_i' - T_i)^+\right]$$

$$\leq \left(1 - \frac{1}{\alpha}\right) \cdot \mathbb{E}\left[\sum_{i \in B_r(S)} w_i'\right] + \mathbb{E}\left[\sum_{i \in B_r(S)} (w_i' - T_i)^+\right]$$

$$\frac{1}{\alpha} \cdot \mathbb{E}\left[\sum_{i \in B_r(S)} w_i'\right] \leq \mathbb{E}\left[\sum_{i \in B_r(S)} (w_i' - T_i)^+\right]$$

which proves the proposition. □

Optimal Algorithm for Matroids

Expected-Margin-Thresholds

- In step j, let S_{j-1} denote the set of selected elements up to step j.
- Denote the element presented in step j by $i = i_j$.
- If $(S_{j-1} \cup i) \notin \mathcal{I}$, set $T_i = \infty$; otherwise set

$$T_i = \frac{1}{2} \cdot \mathbb{E}\left[w'(B_r(S_{j-1})) - w'(B_r(S_{j-1} \cup i))\right] \quad ,$$

  where w'_i is sampled independently from D_i.
- Select i if $w_i \geq T_i$; reject i otherwise.

Theorem (Kleinberg, Weinberg 2012)

Expected-Margin-Thresholds has 2-balanced thresholds and is 2-competitive for matroid domains.

The prophet-inequality model seems "easier" than the secretary model. We obtain improved guarantees in the single-item case (2 vs. e). For matroids (2 vs. $O(\sqrt{\log k})$) the answer depends on the matroid secretary conjecture.

Recommended Literature

- M. Babaioff, N. Immorlica, R. Kleinberg. Matroids, Secretary Problems, and Online Mechanisms. SODA 2007.
- S. Chakraborty, O. Lachish. Improved Competitive Ratio for the Matroid Secretary Problem. SODA 2012.
- P. Freeman. The Secretary Problem and its Extensions: A Review. Intl. Stat. Rev., 51(2):189-206, 1983.
- R. Kleinberg, M. Weinberg. Matroid Prophet Inequalities. STOC 2012.
- N. Korula, M. Pal. Algorithms for Secretary Problems on Graphs and Hypergraphs. ICALP 2009.
- U. Krengel, L. Sucheston. On semiamarts, amarts, and processes with finite value. Adv. in Prob. Related Topics, 4:197-266, 1978.