Homework 3

Out: *Feb 23* Due: *March 15*

Instructions:

- Each problem is worth twenty-five points. **Attempt any 4 of the 5 problems.** If you submit all 5 then your total score will be the **sum of top 4 problem scores**.
- Upload your solutions to the problems as a single PDF on Gradescope. Please anonymize all your submissions (i.e., do not list your name in the PDF), but if you forget, it's OK.
- You may collaborate with any classmates, textbooks, Internet, etc. Please upload a brief "collaboration statement" listing any collaborators as the last page of your non-extra-credit solutions PDF on Gradescope. But after the collaboration, always **write your solutions individually**.
- If you choose to do extra credit, upload your solution to the extra credits as a single separate PDF file to Gradescope. Please again anonymize your submission.
- For each problem, you should aim to keep your writeup below one page. For some problems, this may be infeasible, and for some problems you may write significantly less than a page. This is not a hard constraint, but part of the assignment is figuring out how to easily convince the grader of correctness, and to do so concisely. "One page" is just a guideline: if your solution is longer because you chose to use figures (or large margins, display math, etc.) that's fine.

Problems:

§1 (Strongly-Convex) Consider an α-strongly convex differentiable function f[1] with the 2-norm of its gradient always bounded by G. Suppose we want to minimize f and let x^* denote its minimum. Show that the gradient descent algorithm with step size $\frac{1}{\alpha(t+1)}$ satisfies $f\left(\frac{\sum_t x_t}{T}\right) - f(x^*) \leq \frac{G^2(1+\log T)}{2\alpha T}$. (Thus, strong-convexity allows us to get $1/T$ dependency in regret instead of $1/\sqrt{T}$ dependency for general convex functions.)

(Hint: Change the potential function in the analysis from Lecture 11 to $\Phi(t) = \frac{t\alpha}{2}\|x_t - x^*\|^2$. Also, use that $\sum_{t\in\{1,\ldots,T\}} \frac{1}{t} \leq 1 + \log T$.)

§2 (General PIs) In Lecture 13 we used Online Contention Resolution Scheme (OCRS) to design a 2-competitive algorithm for Prophet Inequality problem of selecting one item on weighted Bernoulli distributions. In this exercise we will extend this technique

[1] A differentiable function f is α-strongly convex for $\alpha > 0$ if for all $x, y \in \mathbb{R}^n$ we have $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\alpha}{2}\|y - x\|_2^2$.

to more general discrete distributions and to selecting $k \geq 1$ items (i.e., our online algorithm can select at most k random variables and its goal is to maximize their sum in expectation. The benchmark is the expected sum of the highest k elements.)

Suppose we are given distributions $\mathcal{D}_i$ of n independent random variables supported on non-negative real numbers $w_1, w_2, \ldots, w_m$, where $X_i \sim \mathcal{D}_i$ takes value w_j with probability p_{ij} for $i \in [n]$ and $j \in [m]$ (this means $\sum_j p_{ij} = 1$ for all $i \in [n]$). Our online algorithm will be revealed outcomes $X_1, \ldots, X_n$ one-by-one and it has to immediately accept/reject, where in total it can accept at most k random variables.

(a) Prove that the following LP relaxation gives an upper bound on the expected offline optimum, i.e, the expected sum of the highest k random variables.

$$\begin{aligned} \max_x \quad & \sum_i \sum_j x_{ij} w_j \\ \text{s.t.} \quad & x_{ij} \leq p_{ij} \qquad \forall i \in [n], j \in [m] \\ & \sum_i \sum_j x_{ij} \leq k \\ & x_{ij} \geq 0 \qquad \forall i \in [n], j \in [m] \end{aligned}$$

Next, we discuss an online rounding algorithm. For the problem of selecting at most k elements, an α-OCRS for $0 \leq \alpha \leq 1$ is defined as follows: we are given $x_1, \ldots, x_n \in [0,1]$ s.t. $\sum_i x_i \leq k$; here x_i denotes the probability with which element i is active independently. Our OCRS algorithm is revealed one-by-one whether the i-th element is active or not, and it has to immediately accept/reject the i-th element if it's active (an inactive element automatically gets rejected). The OCRS algorithm can accept at most k elements in total. We say that such an algorithm is an α-OCRS if it guarantees that each element $i \in [n]$ gets selected at least $\alpha \cdot x_i$ fraction of the times.

(b) Design a $1-O(\sqrt{\frac{\log k}{k}})$-OCRS for selecting k elements. (Note that this approaches 1 as k becomes large.)

(Hint: Generalize the 1/4-OCRS proof that we saw in Lecture 13. In particular, argue using Chernoff bounds that an online algorithm that rejects the i-th element with probability $\Theta(\sqrt{\frac{\log k}{k}})$ and otherwise accepts if it's active gives a $1 - O(\sqrt{\frac{\log k}{k}})$-OCRS.)

(c) Using (a), show that an α-OCRS for selecting k elements implies a $1/\alpha$-competitive algorithm for the PI problem of selecting at most k random variables.

(Hint: Suppose x^*_{ij} is the optimal LP solution. Let $x_i := \sum_j x^*_{ij}$ denote the probability with which the i-th element is active in the α-OCRS for selecting k elements .)

§3 (Pandora's Box Matching) In this problem we consider a combinatorial generalization of the Pandora's Box problem that we saw in Lecture 15 to selecting a matching.

Suppose we are given a graph $G = (V, E)$. The weight/value of each edge $e \in E$ is given by a random variable $X_e \sim \mathcal{D}_e$ (thus each edge corresponds to a box), where

$\mathcal{D}_e$ is known probability distribution. Our algorithm can find the outcome/realization X_e by paying a known cost $c_e \in \mathbb{R}_{\geq 0}$. The goal is to design a policy that opens a subset $O \subseteq E$ of the boxes/edges and selects a subset $S \subseteq O$ of them that form a valid matching to maximize the expected total utility:

$$\mathbb{E}\Big[\max_{S \subseteq O \text{ s.t. } S \text{ is a matching}} \sum_{e \in S} X_e - \sum_{o \in O} c_o\Big].$$

(a) Prove that this Pandora's box matching problem generalizes the single item Pandora's box problem that we saw in class.

(Hint: Consider the star graph: a single node u connected to multiple nodes $v_1, v_2, \ldots, v_n$.)

(b) Recall the definitions of σ_e and κ_e from Lecture 15. That is, we define σ_e s.t. it satisfies $\mathbb{E}[(X_e - \sigma_e)^+] = c_e$ and define $\kappa_e = \min\{X_e, \sigma_e\}$. Prove that the expected utility of the optimal policy for Pandora's Matching problem is at most

$$\mathbb{E}\Big[\max_{S \subseteq E \text{ s.t. } S \text{ is a matching}} \sum_{e \in S} \kappa_e\Big].$$

(Hint: In class we saw why for single-item Pandora's box the optimal utility is at most $\mathbb{E}[\max \kappa_i]$. Slightly modify that proof to prove this result.)

§4 (Gittin's Index $\Rightarrow$ Pandora's Box) Suppose we are given a Pandora's Box instance where Box i contains a random value X_i that we can find by paying a cost c_i, where X_i takes a discrete value $r_j \in \mathbb{R}_{\geq 0}$ for $j \in \{1, \ldots, m\}$ with probability p_{ij} and $\sum_j p_{ij} = 1$. In Lecture 15 we already saw the optimal policy for Pandora's Box. In this problem we will design the same policy and prove its optimality but using the Gittin's Index Theorem from Lecture 16. (That is, you cannot use any results from Lecture 15 to solve this problem.)

For the reduction, we define the following Gittin's Index instance with n different Markov Chains where the i-th chain (that corresponds to i-th box) has $m+1$ states. The i-th Markov chain starts at State 0 and if we Play it then we transition to one of the m states where we reach State j w.p. p_{ij} (this state corresponds to $X_i = r_j$) and receive reward $-c_i$. If we further play this Markov chain then we stay in State j with probability 1 and receive reward $r_j(1-\gamma)$ every time we play it (see Figure 1).

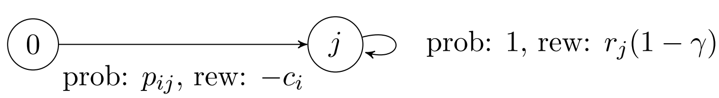

Figure 1: Markov chain showing the transition for $X_i = r_j$.

The idea of the above reduction is to imagine $\gamma \to 1$. Now the optimal strategy will first Play a few boxes (i.e., open them by paying their costs; since $\gamma \to 1$ the discount factor is negligible right now) and once the algorithm wants to stop at some box i then it keeps playing that i-th Markov chain till time ∞ (since it will continue remaining the highest index chain). Note that this is the reason the reward of playing in State

j is scaled by a factor of $(1-\gamma)$, since if we get $r_j(1-\gamma)$ for all the time steps then our total γ-discounted reward equals r_j.

(a) Let λ_i denote the Gittin's index for the i-th Markov Chain when in State 0. We define $\widetilde{\sigma}_i := \frac{\lambda_i}{1-\gamma}$. Prove that the following equation holds:

$$\mathbb{E}\Big[-c_i - (1-\gamma)\widetilde{\sigma}_i + \gamma\big(X_i - \widetilde{\sigma}_i\big)^+\Big] = 0$$

(b) Note that if $\gamma \to 1$ then $\widetilde{\sigma}_i$ approaches Weitzman's index σ_i from class (i.e., the solution to $\mathbb{E}[(X_i - \sigma_i)^+] = c_i$). Use this to prove that for $\gamma \to 1$ the Gittin's Index policy on this instance becomes the same as Weitzman's Index policy.

§5 (Adaptivity Gaps for Stochastic Matching) In this problem we will generalize ProbeMax from Lecture 17 to selecting a matching.

Suppose we are given a bipartite graph G with parts A, B and edges E. The weight of each edge $e \in E$ is given by a random variable X_e, where $X_e = v$ with some known probability $f_{e,v}$. Our algorithm can open at most k edges and find their outcomes/realizations X_e. The goal is to design a policy that opens a subset $O \subseteq E$ of the edges, where $|O| \leq k$, and select a subset $S \subseteq O$ of them that form a valid matching to maximize the weight of the matching:

$$\mathbb{E}\Big[\max_{S \subseteq O \text{ s.t. } S \text{ is a matching}} \sum_{e \in S} X_e\Big].$$

(a) Prove that the following LP gives an upper bound on the optimal policy:

$$\begin{aligned}
\text{maximize} \quad & \sum_{e,v} v \cdot z_{e,v} \\
\text{subject to} \quad & \sum_{e} y_e \leq k \\
& \sum_{e \,:\, e \text{ is incident to vertex } u} \; \sum_{v} z_{e,v} \leq 1 && u \in A \\
& \sum_{e \,:\, e \text{ is incident to vertex } u} \; \sum_{v} z_{e,v} \leq 1 && u \in B \\
& z_{e,v} \leq f_{e,v} \cdot y_e && \forall e \in E \ , \ \forall v \\
& z_{e,v} \geq 0 && \forall e \in E \ , \ \forall v
\end{aligned}$$

(b) Prove that the adaptivity gap for this problem is $O(1)$.
(Hint: To argue about an edge $e = (a, b)$, apply the union bound and Markov's inequality argument from both vertex a and vertex b.)

Extra Credit:

§6 (extra credit) In the setting of Problem 3, find an algorithm with expected utility, i.e. total reward minus total cost, at least 1/2 of the upper bound in Problem 3.

(Hint: Run the greedy matching algorithm w.r.t. the indices σ_e and use the fact that for offline max-weight matching if we greedily select edges in decreasing weight order then we get a 1/2 approximation.)

§7 (extra credit:) Consider the OCRS problem of selecting at most k elements in Problem 2. Show that there exists a $1 - O\big(\sqrt{\frac{1}{k}}\big)$-OCRS for selecting k elements, i.e., improve the result in Problem 2 by removing the extra $\sqrt{\log k}$ factor.

(Hint: Generalize the idea of 1/2-OCRS from Lecture Notes.)