Homework #2
DATA 37200: Learning, Decisions, and Limits (Winter 2025)

February 14, 2025

Due date: Midnight, Thursday February 27. Submit on gradescope like usual.

Problem 1: Finding stationary points with gradient descent

In class, we saw some relatively sophisticated analyses of gradient descent. Here we review one of the more classical and elementary ones for the offline setting. In this problem we won't assume convexity, but we also only prove that we reach an approximate stationary point, i.e. a point where $\nabla f \approx 0$.

Let $f : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a smooth and nonnegative function such that $(\nabla^2 f)(x) \preceq HI$ for all $x \in \mathbb{R}^n$.

1. Prove that for any x and $\eta > 0$, if we define $y = x - \eta \nabla f(x)$ then

$$f(y) \leq f(x) - \eta \|\nabla f\|^2 + \eta^2 \frac{H\|\nabla f\|^2}{2}.$$

2. What value of η optimizes the right hand side of the above inequality? (I.e. makes the right hand side as small as possible.)

3. Suppose that x_0 is any point in $\mathbb{R}^n$ and that we define for all $t \geq 1$,

$$x_t = x_{t-1} - \eta \nabla f(x_{t-1})$$

where η is the optimal step size chosen in the previous question. Let $T > 0$ be an arbitrary integer. Prove that that exists t with $0 \leq t \leq T$ such that

$$\|\nabla f(x_t)\|^2 \leq CH \frac{f(x_0)}{T}$$

for some absolute constant $C > 0$.

Problem 2: Playing adversarial bandits with projected gradient descent

In the first two weeks of class, we learned how to construct a good strategy for stochastic multi-armed bandits. A more difficult model, which is also popular to study, is called *adversarial multi-armed bandits*. Consider the following game between Nature and Gambler. As usual, let $K \geq 2$ be the number of possible actions (arms) the Gambler can choose between at each step.

First, Nature fixes[1] a reward function $r_t \in [K] \to [0, 1]$ for every t from 1 to T. They do *not* directly reveal these rewards to the gambler.

Then, for $t = 1$ to T:

1. Gambler, based off their past experiences and without knowledge of the hidden rewards, selects an distribution p_t over $[K]$ and samples $\pi_t \sim p_t$.
2. Gambler observes reward $r_t(\pi_t)$.

The regret for this game is

$$\max_{\pi \in [K]} \sum_{t=1}^{T} r_t(\pi) - \sum_{t=1}^{T} r_t(\pi_t).$$

It's not that obvious that this is a problem which can be solved with sublinear expected regret, since the rewards are unstructured and the gambler only gets feedback for the arm which they select at every round. Nevertheless, in this problem we show that it is possible using a simple strategy based on "projected" online gradient descent (i.e. gradient descent with the iterates constrained to a convex set).

1. For $i \in [K]$ let e_i denote the ith standard basis vector in $\mathbb{R}^K$, e.g. $e_1 = (1, 0, \ldots), e_2 = (0, 1, 0, \ldots)$ and so on. Show that
$$r_t = \mathbb{E}_{p_t \sim \pi_t} \left[\frac{r_t(\pi_t)}{p_t(\pi_t)} e_{\pi_t} \right]$$
2. Read Corollary 2.17 of the online learning survey (https://www.cs.huji.ac.il/~shais/papers/OLsurvey.pdf), which gives a guarantee for projected online gradient descent very similar to the one we covered in class for (normal) online gradient descent. Explicitly write down the special case of Corollary 2.17 when the convex set S is the truncated simplex
$$\Delta_\epsilon = \{q \in \mathbb{R}^K_{\geq 0} : q_i \geq \epsilon \forall i \in [K], \sum_i q_i = 1\}.$$
3. Combining the previous two items and optimizing over the choice of ϵ, show that there exists a strategy applying online projected gradient descent to the sequence of vectors $-\frac{r_t(\pi_t)}{p_t(\pi_t)} e_{\pi_t}$ which gets $o(T)$ regret for adversarial bandits when the number of arms A is fixed.

Problem 3: Concentration and learning

Complete Exercise 1 of https://arxiv.org/pdf/2312.16730 on page 19. You may use any results from Chapter 1 there (in particular, you will need to read the statement of Bernstein's inequality, which is an extremely important variant of Hoeffding's inequality that often gives tighter bounds). This type of analysis appears everywhere in learning — fairly similar ideas are used in the concentration analysis in LinUCB (which we skipped over in class).

[1] It's also possible to consider a slightly harder version of the problem where Nature picks the reward functions adaptively. Similar ideas work.