Advanced Topics in Artificial Intelligence: EE6180
Indian Institute of Technology Madras — November 18, 2020
Instructor: Abhishek Sinha

Problem Set 4

• This problem set is due on **December 2, 2020** in the class.
• Each problem carries 10 points.
• No collaboration among the students allowed. Any two or more identical or nearly-identical solutions will automatically receive zero points each.

---

1. **(A PAC-Bayesian Theorem)** In this problem, we will prove a different version of the PAC-Bayesian Theorem from what we derived in the class. Recall the notations introduced in the class. We will show that for any fixed prior distribution P on $\mathcal{H}$ and any $0 < \delta \leq 1$ the following statement holds with probability greater than $1 - \delta$ over S:

$$D(\hat{L}_S(Q)||L(Q)) \leq \frac{D(Q||P) + \log \frac{2m}{\delta}}{m-1}, \quad \forall Q. \tag{1}$$

The statement above will follow from the following two bounds.

(a) First, using Donsker-Varadhan's inequality, show that for any fixed prior distribution P on $\mathcal{H}$, we have:

$$(m-1)D(\hat{L}_S(Q)||L(Q)) \leq D(Q||P) + \ln \mathbb{E}_{h\sim P}[e^{(m-1)D(\hat{L}_S(h)||L(h))}].$$

(b) Next, following the steps below, show that for any fixed probability distribution P on $\mathcal{H}$ and any $0 < \delta \leq 1$, the following upper bound holds with probability at least $1 - \delta$:

$$\mathbb{E}_{h\sim P}[e^{(m-1)D(\hat{L}_S(h)||L(h))}] \leq \frac{2m}{\delta}.$$

I Prove that for any real valued random variable X satisfying $\mathbb{P}(X \leq \epsilon) \leq e^{-mf(\epsilon)}$ where $f(\cdot)$ is a non-negative non-increasing function, the following inequality holds:

$$\mathbb{E}[e^{(m-1)f(X)}] \leq m.$$

II Chernoff-Hoeffding's bound states that for i.i.d. random variables $X_1, X_2, \ldots, X_m$ from the interval $[0, 1]$, we have

$$\mathbb{P}(\bar{X} \leq \epsilon) \leq e^{-mD^+(\epsilon||\mathbb{E}(X_1))},$$

where $D^+(p||q) = 0$ if $p \geq q$ and is $D(p||q)$ otherwise. Using the Chernoff-Hoeffding's bound and part I, show that

$$\mathbb{E}_{S\sim D^m}[e^{(m-1)D^+(\hat{L}_S(h)||L(h))}] \leq m.$$

III Use Markov's inequality to prove that for any $\delta \in [0,1]$, we have with probability at least $1 - \frac{\delta}{2}$ over S:

$$\mathbb{E}_{h\sim P}[e^{(m-1)D^{+}(\hat{L}_S(h)||L(h))}] \leq \frac{2m}{\delta}.$$

IV Prove the PAC-Bayes bound given in Eqn. (1).

2. **(Non-parametric Least-Square Estimation)** Consider the function class $S_{\alpha,\gamma}(C_{\max}, L)$ which we introduced in the notes. Recall that,

$$S_{\alpha,\gamma}(C_{\max}, L) = \{f : [0,1] \to \mathbb{R} : |f^{(j)}|_\infty \leq C_{\max}, \forall 0 \leq j \leq \alpha, \text{ and } |f^\alpha(x) - f^\alpha(y)| \leq L|x-y|^\gamma, \forall x, y \in [0,1].\}$$

It can be shown that for some C (which depends on the parameters), the δ-covering number of $S_{\alpha,\gamma}(C_{\max}, L)$ in the sup-norm may be bounded as follows:

$$\log N(\delta, S_{\alpha,\gamma}(C_{\max}, L), ||\cdot||_\infty) \leq C\left(\frac{1}{\delta}\right)^{1/(\alpha+\gamma)}.$$

Suppose we observe

$$Y_i = f^*(x_i) + \epsilon_i, \quad 1 \leq i \leq n,$$

where $f^* \in S_{\alpha,\gamma}(C_{\max}, L)$, and ϵ_i are i.i.d. standard Gaussians and the x_i's are deterministic points in $[0,1]$. Consider the non-parametric least-square estimator

$$\hat{f} \in \arg\min_{f \in S_{\alpha,\gamma}(C_{\max}, L)} \frac{1}{n}\sum_{i=1}^{n}\big(Y_i - f(x_i)\big)^2.$$

Using the notion of Gaussian complexity of the function class $S_{\alpha,\gamma}(C_{\max}, L)$ and **Dudley's entropy integral**, prove an upper-bound for the mean-squared estimation error:

$$\mathsf{MSE} \equiv \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}\big(\hat{f}(x_i) - f^*(x_i)\big)^2\right).$$

3. **(Online Mirror Descent)** Besides FTRL and FTPL, Online Mirror Descent (OMD) is yet another general framework to derive online learning algorithm for OCO. Recall the OCO framework as discussed in the class. For a differentiable convex function $\psi : \Omega \to \mathbb{R}$, the Bregman divergence (w.r.t. ψ) between two points w and u is defined as

$$D_\psi(w,u) = \psi(w) - \psi(u) - \langle \nabla\psi(u), w - u\rangle.$$

The function ψ is chosen such that the mapping $\nabla\psi : \Omega \to \Omega$ is invertible [1]. The update of OMD is then

$$w_{t+1} = \arg\min_{w\in\Omega}\left[\langle w, \nabla f_t(w_t)\rangle + \frac{1}{\eta}D_\psi(w, w_t)\right],$$

[1] More precisely, the function $\psi : \Omega \to \mathbb{R}$ is chosen to be a Legendre function.

for some step size $\eta > 0$. In other words, OMD tries to find a point that minimizes the loss at time t while being close to the previous point w_t.
(a) Let w'_{t+1} be such that $\nabla\psi(w'_{t+1}) = \nabla\psi(w_t) - \eta\nabla f_t(w_t)$ (assume that it exists). Prove that

$$w_{t+1} = \arg\min_{w\in\Omega} D_\psi(w, w'_{t+1}).$$

(b) Verify that for any $u \in \Omega$, the instantaneous regret can be written as

$$\langle w_t - u, \nabla f_t(w_t)\rangle = \frac{1}{\eta}\big(D_\psi(u, w_t) - D_\psi(u, w'_{t+1}) + D_\psi(w_t, w'_{t+1})\big).$$

(c) Show that

$$D_\psi(u, w_{t+1}) \leq D_\psi(u, w'_{t+1}), \quad \forall u \in \Omega.$$

(d) Hence conclude the following regret bound for OMD

$$\sum_{t=1}^{T}\big(f_t(w_t) - f_t(u)\big) \leq \frac{D_\psi(u, w_1)}{\eta} + \frac{1}{\eta}\sum_{t=1}^{T} D_\psi(w_t, w'_{t+1}). \tag{2}$$

(e) Show that Hedge is an instance of OMD and recover its regret bound using Eqn. (2).

4. **(Experimenting with MAB algorithms)** This problem is designed to give you a step-by-step hands-on experience of working with Multi Armed Bandit (MAB) algorithms by understanding, modifying, and experimenting with an existing MAB code written in Python[2].

   (a) Download the Github repository:
   `https://github.com/johnmyleswhite/BanditsBook`
   The code is located in a directory named ∼`/BanditsBook/`.

   (b) Change your current directory to `/Banditsbook/`. Read the `README.md` file carefully and familiarize yourself with the structure of the codebase. This repository implements the following six standard bandit algorithms - ϵ`-Greedy, Softmax, UCB1, UCB2, Hedge,` and `Exp3`.

   (c) Change your current directory to `/python/algorithms/` and check out the source code of each of the above algorithms. The codes differ in how the functions `select_arm()` and `update()` are implemented for each of the above algorithms. Make sure you fully understand the working of these two functions for each of the above algorithms.

[2] Refer to `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011/python-tutorial/` for a quick tutorial.

(d) The code implements three different models of bandits - `adversarial, Bernoulli,` and `Normal`. Check out the relevant codes at `/python/arms/`.

(e) In this problem, we will compare the performance of ϵ`-Greedy` (for $\epsilon = 0.05$), `UCB1,` and `Exp3` (with random exploration probability $\gamma = 0.05$) policies for four Bernoulli bandits for a horizon of length $T = 10^4$ and averaging the result over $N = 100$ simulations. Set the expected reward values of the bandits to be $\boldsymbol{p} = [0.5, 0.95, 0.2, 0.8]$.

(f) Modify the parameters in the file `/python/demo.py` to set up the required simulation environment.

(g) By suitably augmenting and modifying the function `test_algorithm()` (defined at `/python/testing_framework/tests.py`), investigate the following:

- For a bandit algorithm π, let $N_a^\pi(t)$ denote the average fraction of times (averaged over N runs) the arm a was selected by the algorithm π by the time t. Plot $N_a^\pi(t), a \in [0, 1, 2, 3]$ as a function of $t \in [0, T]$ for each of the above three algorithms. What do you observe from the nature of the plots? Can you guess what happens when $T \to \infty$?
- For a bandit algorithm π, let $R^\pi(t)$ denote the *pseudo-regret* of the algorithm π up to time t. In other words, if $\bar{r}^\pi(t)$ denotes the average-reward (over N runs) obtained by the algorithm π at time t, then the pseudo-regret is defined as $R^\pi(t) = t \max_i p_i - \sum_{\tau=1}^{t} \bar{r}^\pi(\tau)$. Plot the time-evolution of $R^\pi(t)$ for the above three algorithms in the same graph. What do you observe from the plots for different range of values of t? How sensitive is the plot with respect to the parameters ϵ and γ?

5. **(Foresight and Hindsight Regret for the IID cost model)** In the class, we upper-bounded the pseudo-regret, also called the *Foresight* regret, where the comparator was taken to be the best arm in *expectation*. In this problem, we will explore the usual notion of regret, also known as the *Hindsight* regret, where the comparator is chosen to be the best *observed* arm.

Consider the adversarial bandit setting as discussed in the class with full feedback and i.i.d. costs from the interval $c_t(a) \in [0, 1], \forall t, a$.

(a) Prove that

$$\min_a \mathbb{E}(\texttt{cost}(a)) \leq \mathbb{E}(\min_a \texttt{cost}(a)) + O(\sqrt{T \log(KT)}).$$

TAKE AWAY: All $\tilde{O}(\sqrt{T})$ regret bounds for algorithms for stochastic bandits (*e.g.,* `UCB`, `Successive Elimination`) carry over to "hindsight regret".

(b) (LOWER BOUND FOR HINDSIGHT REGRET) Construct a problem instance with a deterministic adversary for which any algorithm suffers regret

$$\mathbb{E}[\texttt{cost(ALG)} - \min_a \texttt{cost}(a)] \geq \Omega(\sqrt{T \log K}).$$

(c) Prove that algorithms `UCB` and Successive Elimination achieve logarithmic regret bound even for hindsight regret, assuming that the *best-in-foresight* arm a^* is unique.