CMPUT 653: Theoretical Foundations of Reinforcement Learning, Winter 2022
Homework #0

Instructions: You need to submit a zip file, named `p00_<name>.zip` where `<name>` is your name. The zip file should include a report in PDF, typed up (we strongly encourage to use pdfLaTeX) and the code that we asked for. Write your name on your solution. I provide a template that you are encouraged to use. You have to submit the zip file on the eclass website of the course.

Collaboration and sources Work on your own. You can consult the problems with your classmates, use books or web, papers, etc. Also, the write-up must be your own and you must acknowledge all the sources (names of people you worked with, books, webpages etc., including class notes.) Failure to do so will be considered cheating. Identical or similar write-ups will be considered cheating as well. Students are expected to understand and explain all the steps of their proofs.

Scheduling Start early: It takes time to solve the problems, as well as to write down the solutions. Most problems should have a short solution (and you can refer to results we have learned about to shorten your solution). Don't repeat calculations that we did in the class unnecessarily.

Deadline: January 10 at 11:55 pm

1 In search of ...

Consider the following simple learning problem: Fix $\delta > 0$ and an integer $k > 0$. Here, k will be the number of actions available. Each action $i \in [k] := \{1, \dots, k\}$ is assigned a value, say, μ_i, which is a real number. The learner does not know these values. The problem of the learner is to find an action that is δ close to the best action out of k actions (higher values are better). The learner can try any action in any order and can even randomize when choosing the next action. When choosing an action, the learner receives in response the value of the action (without noise!). When choosing an action, the learner can take into account all the past observations and choices: It has no limit of what it can remember, or on the precision of its memory. At one point, the learner needs to stop and return an action.

A *problem instance* that the learner interacts with is fully characterized by $\mu = (\mu_1, \dots, \mu_k) \in \mathbb{R}^k$, the k values assigned to the actions. The learner is called *δ-sound* for a set of instances $\mathcal{H} \subset \mathbb{R}^k$ if no matter the problem instance taken from $\mathcal{H}$, when the learner interacts with the aforementioned way with the chosen problem instance, the learner always stops and returns an action that is strictly less than δ away from the optimal action on that instance. If the instance was $\mu \in \mathbb{R}^k$, the learner chose $A \in [k]$, with probability one, it has to hold that $\mu_A > \max_{i \in [k]} \mu_i - \delta$ (note the strict inequality). For a learner $\mathcal{A}$, let $q(\mathcal{A}, \mu)$ be the expected value of the number of rounds that the learner spends with interacting with problem instance μ (we need the expected value, because learners may randomize). By slightly abusing notation, let $q(\mathcal{A}, \mathcal{H}) = \max_{\mu \in \mathcal{H}} q(\mathcal{A}, \mu)$ be the expected number of rounds that the learner $\mathcal{A}$ will spend on the instance in $\mathcal{H}$ which makes it the "slowest".

Let $\mathcal{S}(\mathcal{H}, \delta)$ be the set of δ-sound learners for $\mathcal{H}$. Further, let $\mathcal{B} = \{e_1, \dots, e_k\} \subset \mathbb{R}^k$ where e_i is the ith standard basis vector: $e_{ij} = 0$ if $j \neq i$ and $e_{ii} = 1$.

Question 1 Describe a *deterministic* learner $\mathcal{A} \in \mathcal{S}(\mathcal{B}, 1)$ such that the learner stops on any instance in $\mathcal{B}$ after at most $k - 1$ queries: $q(\mathcal{A}, \mathcal{B}) \leq k - 1$.

Total: **5 points**

Question 2 Formally prove that for the learner $\mathcal{A}$ you described $\mathcal{A} \in \mathcal{S}(\mathcal{B}, 1)$ holds.

Total: **5 points**

Question 3 Formally prove that for the learner $\mathcal{A}$ you described $q(\mathcal{A}, \mathcal{B}) \leq k - 1$ holds.

Total: **5 points**

Question 4 Show that any deterministic learner $\mathcal{A} \in \mathcal{S}(\mathcal{B}, 1)$ needs at least $k - 1$ queries in the worst case for instances in $\mathcal{B}$.

Total: **10 points**

Question 5 Describe a learner $\mathcal{A} \in \mathcal{S}(\mathcal{B}, 1)$ (deterministic or not) such that the learner stops on any instance in $\mathcal{B}$ after at most $(k+1)/2 - 1/k$ queries on expectation: $q(\mathcal{A}, \mathcal{B}) \leq (k+1)/2 - 1/k$. Formally prove that the learner is indeed sound and it indeed stops after at most $(k+1)/2 - 1/k$ queries on expectation, on every problem instance in $\mathcal{B}$.

Total: **15 points**

Question 6 Show that any learner $\mathcal{A} \in \mathcal{S}(\mathcal{B}, 1)$ needs at least $(k+1)/2 - 1/k$ queries on expectation in the worst case for instances in $\mathcal{B}$. That is, prove that for any $\mathcal{A} \in \mathcal{S}(\mathcal{B}, 1)$, $q(\mathcal{A}, \mathcal{B}) \geq (k+1)/2 - 1/k$.

Hint: Yao's lemma states "that the expected cost of a randomized algorithm on the worst-case input is no better than the expected cost for a worst-case probability distribution on the inputs of the deterministic algorithm that performs best against that distribution." (source: wikipedia). Use this lemma.

Total: **15 points**

Question 7 (for fun) It appears that as far as the expected number of rounds is considered, randomized learners can do better than deterministic learners. Is this *real*? Would *you* implement the randomized learner? Why or why not?

Total: **0 points**

2 Basic probability questions

Assume that you are given a randomized algorithm $\mathcal{B}$ that returns the *unique* correct answer on any problem it is fed with (e.g., computing a shortest path) with probability at least $2/3$. Fix $\delta > 0$.

Question 8 Design an algorithm that returns the correct answer with probability at least $1 - \delta$. The algorithm may use $\mathcal{B}$. Describe how your algorithm works (pseudocode preferred).

Total: **5 points**

Question 9 Bound the expected runtime of your algorithm as a function of the expected runtime of $\mathcal{B}$. In particular, if the expected runtime of $\mathcal{B}$ on input x is $r(\mathcal{B}, x)$, then show that the expected runtime of your algorithm on input x is at most $\lceil 72 \ln(1/\delta) \rceil\, r(\mathcal{B}, x)$. If needed modify your algorithm (in which case modify your answer to the previous question).
Hint: Use the "Chernoff lower tail bound" for independent Bernoulli variables. This states the following: Let S be the sum of n independent Bernoulli random variables. Let $\mu = \mathbb{E}[S]$ be the expected value of S. Then, for any $0 \le \delta \le 1$, $\mathbb{P}(S \le (1-\delta)\mu) \le \exp(-\mu\delta^2/3)$.

Total: **5 points**

Question 10 For the same setting as in Question 8, now consider the case when $\mathcal{B}$ itself is correct with probability at least $1-\delta$. Consider the case when $\mathcal{B}$ is run on s different inputs. Show that for any $s \le 1/\delta$, the probability that $\mathcal{B}$ is correct on *all these inputs* is at least $1 - s\delta$.

Total: **5 points**

---

3 Basic calculus and such

Question 11 For $x \in \mathbb{R}^d$ and $p \ge 1$ let $\|x\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$. For $p = \infty$, let $\|x\|_\infty = \max_i |x_i|$. Show that for any x, $1 \le p \le q \le \infty$, $\|x\|_p \ge \|x\|_q$.

Total: **5 points**

Question 12 Let $1 \le p \le \infty$. Show that for any $c \in \mathbb{R}$, $x, y \in \mathbb{R}^d$, the following hold:

1. $\|x\|_p \ge 0$ and if $\|x\|_p = 0$ then $x = 0$;

   5 points

2. $\|cx\|_p = |c|\,\|x\|_p$;

   5 points

3. $\|x + y\|_p \le \|x\|_p + \|y\|_p$.

   5 points

(That is, $\|\cdot\|_p$ is a norm on $\mathbb{R}^d$.)

Total: **15 points**

Question 13 Let $\|\cdot\|$ be any norm on the vector-space of $d \times d$ matrices (that is, $\|\cdot\| : \mathbb{R}^{d\times d} \to \mathbb{R}$ satisfies the properties of a norm when we treat $\mathbb{R}^{d\times d}$ as a d^2-dimensional vector space over the reals). Further, assume that $\|\cdot\|$ is submultiplicative: for any $A, B \in \mathbb{R}^{d\times d}$, $\|AB\| \le \|A\|\|B\|$. Let I be the $d \times d$ identity matrix, $A \in \mathbb{R}^{d\times d}$ arbitrary. Show that $I - A$ is nonsingular when $\sum_{k=0}^\infty \|A\|^k < \infty$.
Hint: Perhaps the inverse of $I - A$ is equal to $\sum_{k\ge 0} A^k$? But is this infinite sum even well-defined?

Total: **15 points**

Question 14 Let now $\|\cdot\|$ be a norm on $\mathbb{R}^d$. Call $s : \mathbb{R}^{d\times d} \to \mathbb{R}$ the "norm induced by $\|\cdot\|$" when for any $A \in \mathbb{R}^{d\times d}$, $s(A) = \sup_{x\in\mathbb{R}^d, \|x\|=1} \|Ax\|$. Show that s is a submultiplicative norm on $\mathbb{R}^{d\times d}$, i.e., it satisfies the premise of the previous question.

Total: **5 points**

Question 15 Calculate a closed form expression for $s(A)$ with $A \in \mathbb{R}^{d\times d}$ and s the norm induced on $\mathbb{R}^{d\times d}$ by the vector-space norm $\|\cdot\|_\infty$.

Total: **5 points**

Question 16 Let $T : U \to V$ be a map from a normed vector space U over the reals to another normed vector space V over the reals. Possibly $U \neq V$, so the norms will also be different, but to reduce clutter, we just use $\|\cdot\|$ to denote the norms on all the vector spaces as from the context it will always be clear which norm we are concerned with. Fix $L \geq 0$. The map T is called L-Lipschitz if $\|T(u) - T(u')\| \leq L\|u - u'\|$.

1. Why can we write $T(u) - T(u')$? What set does the result of this operation belong to? Why?

   2 points

2. Let $T : U \to V$ be L-Lipschitz, $S : V \to W$ be M-Lipschitz (W is also a normed vector space and $M \geq 0$). Prove that $S \circ T$ is LM-Lipschitz. Here, $S \circ T$ is the composition of T and S. Thus, $S \circ T : U \to W$ and in particular $(S \circ T)(u) = S(T(u))$.

   8 points

Total: **10 points**

Total for all questions: 125