CS4803/7643: Deep Learning
Fall 2020
Problem Set 5

Instructor: Dhruv Batra
TAs: Sameer Dharur, Joanne Truong, Yihao Chen, Hrishikesh Kale
Tianyu Zhan, Prabhav Chawla, Guillermo Nicolas Grande, Michael Piseno
Discussions: https://piazza.com/gatech/fall2020/cs48037643

Due: Wednesday November 4th, 2020, 11:59pm

Instructions

1. We will be using Gradescope to collect your assignments. Please read the following instructions for submitting to Gradescope carefully!

   - Each subproblem must be submitted on a separate page. When submitting to Gradescope, make sure to mark the page(s) corresponding to each problem/sub-problem. For instance, Q3 has 3 subproblems - the solution to each must start on a new page and be marked accordingly.
   - Remember to append your notebook PDFs to the solutions for the problem set, as described in the instructions!
   - For the coding problem, please use the provided collect_submission.sh script and upload hw5_code.zip to the HW5 Code assignment on Gradescope. While we will not be explicitly grading your code, you are still required to submit it. Please make sure you have saved the most recent version of your jupyter notebook before running this script.
   - Note: This is a large class and Gradescope's assignment segmentation features are essential. Failure to follow these instructions may result in parts of your assignment not being graded. We will not entertain regrading requests for failure to follow instructions. Please read https://stats200.stanford.edu/gradescope_tips.pdf for additional information on submitting to Gradescope.

2. LaTeX'd solutions are strongly encouraged (solution template available at cc.gatech.edu/classes/AY2021/cs7643_fall/assets/sol5.tex), but scanned handwritten copies are acceptable. Hard copies are **not** accepted.

3. We generally encourage you to collaborate with other students.

   You may talk to a friend, discuss the questions and potential directions for solving them. However, you need to write your own solutions and code separately, and *not* as a group activity. Please list the students you collaborated with.

1 Optimal Policy and Value Function [4 points]

Consider a simple 2-state MDP shown in Figure 1, $\mathcal{S} = \{S_1, S_2\}$. From each state, there are two available actions $\mathcal{A} = \{stay, go\}$. The reward received for taking an action at a state is shown in the figure for each (s, a) pair. Also, taking action go from state S_2 ends the episode. All transitions are deterministic.

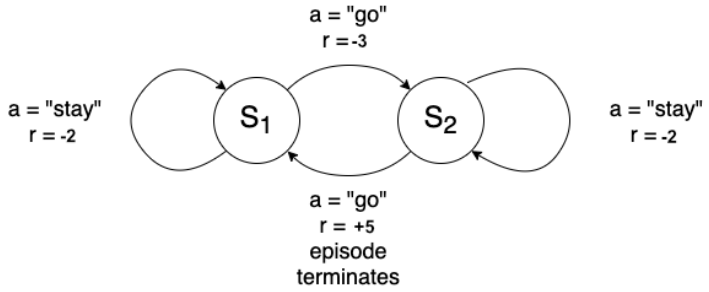

Figure 1: 2-state MDP

(a) **[1 point]** Consider a policy that always chooses the *stay* action at every state. Compute the sum of discounted rewards obtained by this policy starting at state S_1, assuming a discount of γ i.e. compute $\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)$.

(b) **[1 point]** What is the optimal policy? Write it as a tuple (a_1, a_2) of the optimal actions at (s_1, s_2) respectively. Compute the sum of discounted rewards obtained by this policy, given that the start state is S_1, with discount γ.

(c) **[2 points]** Recall that one update of the value iteration algorithm performs the following operation, where $r(s, a)$ denotes the reward for taking action a at state s.

For each $s \in \mathcal{S}$:

$$V^{i+1}(s) \leftarrow \max_a \sum_{s'} p(s'|s, a) \left[r(s, a) + \gamma V^i(s') \right] \tag{1}$$

Note: Since taking action go at s_2 will terminate the episode, the value of taking the go action at s_2 is just $r(s_2, go)$ instead of $r(s_2, go) + \gamma V(s_1)$ (since you don't care about future values after the episode has ended). Thus the update for state $V(s_2)$ will look like:

$$V^{i+1}(s_2) \leftarrow \max \left(r(s_2, stay) + \gamma V^i(s_2),\ r(s_2, go) \right) \tag{2}$$

The value function V is stored as an array of size $|\mathcal{S}|$ ($|\mathcal{S}| = 2$ in our case). For example, the array $V' = [0, 1]$ denotes that $V'(s_1) = 0, V'(s_2) = 1$. Starting with a initial $V^0 = [0, 0]$, peform the value iteration update to compute V^1, V^2, V^3 assuming a discount factor $\gamma = 1$. What is the optimal V^*?

2 Value Iteration Convergence [7 points]

In part (c) of the previous question, we computed V^i for a few updates of value iteration and also V^*, the optimal value function. What happened to the "error" in the value estimate at every update i.e. did $|V^i(s) - V^*(s)|$ every increase for a particular state s? The answer is yes, the value estimate

for any s may fluctuate before eventually converging to $V^*(s)$. If this is the case, do we have any hope that the value iteration algorithm will converge to V^*? In this question, we will prove that value iteration indeed converges due to the monotonic decrease in a different error - the maximum error over all states $\max_{s \in \mathcal{S}} |V^i(s) - V^*(s)|$ or the L^∞ norm $\left\|V^i - V^*\right\|_\infty$.

(a) **[1 point]** For V^1, V^2, V^3 obtained in 1(c), compute $\left\|V^i - V^*\right\|_\infty$ and verify that this error decreased monotonically.

(b) **[2 points]** Let $\mathrm{T} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ be the function that takes the vector V^i as input and applies the value iteration update to produce V^{i+1}. We will show that for any two vectors $V, V' \in \mathbb{R}^{|\mathcal{S}|}$, this operator decreases the L^∞ norm between them.

Prove that: $\|\mathrm{T}(V) - \mathrm{T}(V')\|_\infty \leq \gamma \|V - V'\|_\infty$ for any $V, V' \in \mathbb{R}^{|\mathcal{S}|}$,

(c) **[2 points; Extra credit for 4803, regular credit for 7643]** Now consider the sequence of value functions $\{V^n\}$ obtained by iteratively performing the value iteration update as per Eq. 1. This sequence has the interesting property of being a *cauchy* sequence i.e. the elements of the sequence become arbitrarily close to each other as the sequence progresses. Formally, we say a sequence is *cauchy* w.r.t. the L^∞ norm if for every positive number ϵ, there is a positive integer N s.t. for all positive integers m, n greater than N, $\|x_m - x_n\|_\infty < \epsilon$. Equipped with this fact, show that the error of V^{n+1} w.r.t the optimal value function can be bounded as: $\forall \epsilon > 0, \exists N > 0$ s.t. $\forall n > N$

$$\left\|V^{n+1} - V^*\right\|_\infty \leq \frac{\gamma}{1-\gamma}\epsilon, \tag{3}$$

Hint: First try to prove that $\left\|V^{n+1} - V^*\right\|_\infty \leq \frac{\gamma}{1-\gamma} \left\|V^n - V^{n+1}\right\|_\infty$, then apply the Cauchy sequence condition on V^n

Discussion: Given Equation 3, we now have a guarantee that value iteration converges to V^*. However, we made the assumption that $\{V^n\}$ is a Cauchy sequence. As a bonus question below, we will prove that the condition in (a) is sufficient to conclude that $\{V^n\}$ is indeed a cauchy sequence.

(d) **[2 point; Extra credit for 4803, regular credit for 7643]** Given a function $\mathrm{T} : \mathbb{R}^n \to \mathbb{R}^n$ such that $\|\mathrm{T}(x_1) - \mathrm{T}(x_2)\|_\infty \leq \alpha \|x_1 - x_2\|_\infty$, $\alpha \in [0, 1)$, prove first that T has a fixed point i.e. $\exists x^* : \mathrm{T}(x^*) = x^*$ and second that the fixed point of T is unique.

3 Learning the Model [5 points; Extra credit for both 4803 and 7643]

While value iteration (VI) allows us to find the optimal policy, performing VI updates (Eq. 1) requires the transition function $\mathbb{T}(s, a) := p(s' \mid s, a)$ [1] and the reward function $\mathcal{R}(s, a)$. In many practical situations, these quantities are unknown. However, we can step through the environment and observe transitions to collect data of the form (s, a, s', r). Using this data, if we can obtain an estimate of the transition and reward functions, we can hope to still find the optimal policy via value iteration. Let M denote the true (unknown to us) model of the world (a model consists of both a transition and reward function), and let $\hat{M}$ represent our approximate model of the world

[1] we will denote $\mathbb{T}(s, a, s')$ as the probability value of s' give (s, a), whereas $\mathbb{T}(s, a)$ denotes the entire probability distribution $p(s' \mid s, a)$

that we estimate from data.

In this question, given estimates of both the transition function $\mathbb{T}(s,a) = p(s' \mid s,a) \in \Delta^{|\mathcal{S}|-1}$ and the reward function $\mathcal{R}(s,a)$, we want to study the error of acting optimally in this estimated MDP as opposed to acting optimally in the "true" MDP. Naturally, we should expect this error to be smaller as we get better estimates which in turn is proportional to how many observations we make.

(a) **[2 points]** For the first part of this problem, assume that we have used our favorite learning method to obtain *estimates* of the transition function $\hat{\mathbb{T}}$ and the reward function $\hat{\mathcal{R}}$. Given that $\max_{s,a} |\hat{\mathcal{R}}(s,a) - \mathcal{R}(s,a)| \leq \epsilon_R$ and $\max_{s,a} ||\hat{\mathbb{T}}(s,a) - \mathbb{T}(s,a)||_1 \leq \epsilon_P$, then for any policy $\pi : \mathcal{S} \to \mathcal{A},\ \forall s \in \mathcal{S}$

$$||V^{\pi}_{\hat{M}} - V^{\pi}_{M}||_\infty \leq \frac{\epsilon_R}{1-\gamma} + \frac{\gamma \epsilon_P}{(1-\gamma)^2} \tag{4}$$

Assume that the reward function is in $[0,1]$.

Hint: 1) Expand $V_M(s)$ using the Bellman equation 2) Holder's Inequality: $||u \cdot v||_1 \leq ||u||_1 \cdot ||v||_\infty$

(b) **[1 point]** We just bounded the error of acting according to any policy $\pi : \mathcal{S} \to \mathcal{A}$ under the estimated model, $\hat{M}$ vs. the true model, M. Now, we use this to bound the value error in acting optimally vs. acting optimally according to the estimated model ($\pi^*_{\hat{M}}$).

Using (4)

$$V^*_M(s) - V^{\pi^*_{\hat{M}}}_M(s) \leq 2 \sup_{\pi : \mathcal{S} \to \mathcal{A}} ||V^{\pi}_{\hat{M}} - V^{\pi}_{M}||_\infty \tag{5}$$

(c) **[1 point]** Given

$$\epsilon_R = \sqrt{\frac{1}{2n} \log \frac{4|\mathcal{S} \times \mathcal{A}|}{\delta}} \tag{6}$$

$$\epsilon_P = |\mathcal{S}| \cdot \sqrt{\frac{1}{2n} \frac{4|\mathcal{S} \times \mathcal{A} \times \mathcal{S}|}{\delta}} \tag{7}$$

use (4) to simplify (5) *i.e.* $V^*_M(s) - V^{\pi^*_M}_M(s)$. What is the dependence on this error and number of observations required for each state action pair, n?

(d) **[1 point]** Given a dataset $D_{s,a} = \{(s,a,s'_1,r_1), \ldots (s,a,s'_n,r_n)\}$ we estimate the *unknown* transition and reward function using empirical frequencies as:

$$\hat{\mathbb{T}}(s,a,s') = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(s'_i = s') \tag{8}$$

$$\hat{R}(s,a) = \frac{1}{n} \sum_{i=1}^{n} r \tag{9}$$

For notational convenience, we will use $\hat{\mathbb{T}}(s,a)$ to denote a vector of size $|S|$ whose element is the probability computed in (8) for each next state, $s' \in \mathcal{S}$. Prove that ϵ_R and ϵ_P take values as in (6) and (7).

Hint: Hoeffding's inequality[2]

[2] https://en.wikipedia.org/wiki/Hoeffding%27s_inequality#General_case_of_bounded_random_variables

4 Policy Gradients Variance Reduction [4 points]

In class, we derived the policy gradient as

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}[\mathcal{R}(\tau)] \tag{10}$$
$$= \mathbb{E}_{\tau\sim\pi_\theta}[\mathcal{R}(\tau)\nabla \log \pi_\theta(\tau)] \tag{11}$$
$$\approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{R}(\tau_i)\nabla_\theta \log \pi_\theta(\tau_i) \tag{12}$$

where τ is a trajectory[3], $\mathcal{R}(\tau)$, the associated reward and N, the number of trials to estimate the expected reward, $J(\cdot)$, the expected reward and θ, the parameters of the policy. These gradients are often very noisy and lead to unstable training. In this question, we show a simple method to reduce the variance in the gradients.

(a) **[2 point]** Show that adjusting the reward as $\mathcal{R}(\tau) := \mathcal{R}(\tau) - b$ does not change the gradient estimate in Eq. 10.

(b) **[2 points]**Compute the variance of $\nabla_\theta J(\theta)$ and notice how subtracting b helped reduce it. What value of b will lead to the least variance?
Hint: The variance of $J(\theta)$ is written as $Var(x) = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$ where x is substituted as $\mathcal{R}(\tau)\nabla_\theta \log \pi_\theta(\tau)$

5 Paper Review [4 points; Extra credit for 4803, regular credit for 7643]

In this course's final paper review section, we examine an interesting project that upends conventional research in reinforcement learning. While most RL problems are aimed at maximizing rewards, this work proposes 'upside down reinforcement learning', where the objective is remodelled as a supervised learning problem. Limited experiments demonstrate promising and competitive results.
The paper can be accessed here.
As in the previous assignments, please limit your reviews to 350 words.
Briefly summarize the key findings, strengths and potential limitations of this work.

6 Coding: Dynamic Programming and Deep Q-Learning [20 regular points + 6 extra credit points for both CS4803 and CS7643]

Follow the instructions at this link to the HW5 coding webpage.

[3] A trajectory is defined as a sequence of states and actions i.e. $(s_0, a_0, s_1, a_1, \ldots s_{n-1}, a_{n-1}, s_n)$