Assignment 1

Due Date 5pm, September 27, 2023.

Submission Instructions Please submit your response as a `pdf` file via Canvas.

Collaboration Policy Collaborating on assignments is permitted, provided the submission lists the students who you collaborated with. As per the Rice Honor System Handbook, this means that students are allowed to develop answers to specific problems together and check answers with each other. However, collaboration does not confer the right for students to submit the exact same document— students must write down the answers themselves. While the "core" of the answer can be the same, the wording cannot be identical unless precise wording is necessary to answer the question. Students must be able to demonstrate that they worked together when developing responses and that one student did not copy off the other. This means that students are barred from dividing questions among themselves.

Citation Policy As per the Rice Honor System Handbook, citations are how students credit other sources in their work and include both in-text citations and a references page. Students don't have an obligation to cite class slides, lectures, or class textbooks on assignments; these are considered common knowledge. An academic citation style is required for all other sources (such as research articles and blogs) on all assignments and project reports. Unless otherwise stated, use of Large Language Models (e.g., ChatGPT, Bard) or programming assistants (e.g., GitHub Copilot) is not permitted.

Late Policy Assignments should be turned on time via Canvas. Assignments handed in late will be marked off 10% per day. Assignments more than 3 days late will not be accepted. In turn, you can expect the teaching staff to grade your assignments and provide feedback in a timely manner.

Grading The assignment is worth 10% of your final grade.

Updates to the Assignment In case there are any updates to the assignment (e.g., additional clarifications, typo fixes, hints, etc.), they will be indicated via the following table.

Version	Date	Note
v1	September 7	Assignment released

1. Rock Paper Scissors 20pts

Rock paper scissors is a two-player hand game, in which each player simultaneously forms one of three shapes – rock, paper, or scissors – with their hand. Rock beats scissors, scissors beats paper, and paper beats rock. So, for example, if Alice and Rob are playing the game and Alice selects rock while Rob selects scissors, then Alice wins. In this problem, we will fill in the role of *Rob the robot*'s programmer and explore the use bandits to program Rob's game playing behavior.

1.1 3pts

Model the problem of solving for Rob's game playing behavior as a bandit problem. Assume that the reward for a win is +1, loss is -1, and tie is 0; and that Alice plays according to the following stochastic policy:

$$\pi_{\text{Alice}}(a) = \begin{cases} 0.8, & a = \text{Rock} \\ 0.1, & a = \text{Paper} \\ 0.1, & a = \text{Scissors}. \end{cases} \tag{1}$$

Report the action space (i.e., the set of actions available to Rob), the conditional reward distributions ($Pr(r|a)$ for each a), and the expected rewards ($q_*(a)$ for each a).

1.2 2pts

Given Alice plays according to the policy described in Eq. 1, compute Rob's optimal *expected cumulative reward* and optimal *policy.* Assume that the total number of games, $N_{\text{games}} = 10000$.

1.3 2pts

What is Rob's optimal policy if Alice plays according to the following policy instead?

$$\pi_{\text{Alice}}(a) = \begin{cases} p_{\text{Rock}}, & a = \text{Rock} \\ p_{\text{Paper}}, & a = \text{Paper} \\ 1 - (p_{\text{Rock}} + p_{\text{Paper}}), & a = \text{Scissors} \end{cases} \tag{2}$$

Hint: Similar to Problem 1.1, derive the expected rewards $q_*(a)$. Once you have derived the expected rewards, you can report the optimal policy simply as $\pi_*(a) = \arg\max_a q_*(a)$.

1.4 3pts

Model the problem of solving for Rob's game playing behavior as a bandit problem for the case when Alice uniformly randomly switches between two policies, i.e., at each round of the game Alice

- first uniformly randomly chooses between the policy of Eq. 1 and Eq. 3, and
- then selects rock, paper, or scissors based on the chosen policy.

Report the action space (i.e., the set of actions available to Rob), the conditional reward distributions ($Pr(r|a)$ for each a), and the expected rewards ($q_*(a)$ for each a).

$$\pi_{\text{Alice}}(a) = \begin{cases} 0.1, & a = \text{Rock} \\ 0.1, & a = \text{Paper} \\ 0.8, & a = \text{Scissors}. \end{cases} \tag{3}$$

Further, for this case, compute Rob's optimal *expected cumulative reward* and optimal *policy.*

1.5 2pts

Play rock paper scissors for at least ten rounds with a friend. In 2-4 sentences, describe your strategy for deciding which shape to play next. To the best of your ability, report a mathematical translation of your strategy (e.g., as a policy).

1.6 3pts

Some of you might have changed your policy over time in response to your friend's game play. Let us assume that Alice also exhibits similar behavior, where her policy changes over time in response to Rob's game play. Is multi-arm bandit a good model for this setting? If yes, why? If no, provide a better alternative with justifications.

1.7 0pts

For a video demonstration of a robot playing rock paper scissors, please follow this link.

1.8 5pts

List two real-world applications where use of the bandits framework is suitable. For each application, (either mathematically or in text) define action and the conditional reward distributions.

2. Prior Knowledge in Bandits 10pts

In our discussion of solving bandit problems, we assumed no prior knowledge of the conditional reward distributions $\Pr(r|a)$ or the action values $q_*(a)$. However, in practice, one might start with partial knowledge of either reward distributions or action values. In this problem, we will explore a particular case of this *partial prior knowledge* setting.

Consider a three-armed bandit with actions a^1, a^2, and a^3 for which rewards lie in the interval $[0, 1]$. Similar to the general bandit setting, the conditional reward distribution $\Pr(r|a^1)$ for action a^1 is unknown to the agent. However, unlike the general case,

- the conditional reward distributions $\Pr(r|a^2)$, and
- the action value $q_*(a^3)$,

are known to the agent *a priori*.

2.1 3pts

Provide pseudo-code for a modified version of the simple bandit algorithm (Section 2.4, textbook) that incorporates this prior knowledge. Remember that your objective should be to design an algorithm that maximizes the cumulative reward given this prior knowledge. Which algorithm – the original or your modified version – do you expect to converge faster?

Hint: Use the prior knowledge to refine the exploration behavior of the simple bandit algorithm.

2.2 4pts

Provide pseudo-code for a modified version of the gradient bandit algorithm (Section 2.8, textbook) that incorporates this prior knowledge.

Hint: Think about the procedure for initializing the policy parameters.

2.3 3pts

After implementing the modified version of the simple bandit algorithm, the agent realized that the prior knowledge that it had was incorrect. In particular, the prior knowledge of action value $q_*(a^3)$ had an error of δ. Given this, does your algorithm from Problem 2.1 still converge to the optimal policy? Why or why not? If no, modify your algorithm from Problem 2.1 such that it is both robust to this error in prior knowledge and in expectation converges faster than the simple bandit algorithm? If yes, modify your algorithm from Problem 2.1 such that it converges faster if the prior knowledge was correct?

3. Gradient Bandit Algorithm: Variance 25pts

Recall the policy gradient-based algorithm for solving bandits. While using this algorithm, the agent does not estimate action values $Q(\cdot)$ but directly learns the policy $\pi(\cdot)$. Consider that the policy being learnt is parameterized using parameters $H(a)\ \forall\ a$ as follows,

$$\pi_t(a) \doteq \Pr(a_t = a; H_t) \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}}, \tag{4}$$

where t denotes time and k denotes the total number of actions available to the agent. Recall that we derived the following update rule to learn the agent's policy

$$H_{t+1}(a) \doteq H_t(a) + \alpha(r_t)(\mathbb{1}_{(a=a_t)} - \pi_t(a)), \quad \forall \ \ a \tag{5}$$

where a_t is the action chosen by the agent at time t, r_t is the reward provided by the environment at time t, $\mathbb{1}(\cdot)$ is the indicator function, and $\alpha > 0$ is a step-size parameter.

3.1 1pts

To succeed, bandit agents need to both explore and exploit. Does the policy gradient algorithm incorporate exploration?

3.2 1pts

If the policy parameters are all initialized to 1, i.e., $H_0(a) = 1 \forall a$, then what is the probability of selecting action x? How does your answer change if the policy parameters are all initialized to 10?

3.3 3pts

Recall that we derived the update rule of Eq. 5 by showing that the expected value of $(r_t)(\mathbb{1}_{(a=a_t)} - \pi_t(a))$ corresponds to the gradient of agent's objective $J(\pi; H) = \sum_a \pi(a) q_*(a)$, i.e.,

$$\nabla_{H(a)} J(\pi; H) = \mathbb{E}[(r_t)(\mathbb{1}_{(a=a_t)} - \pi_t(a))]. \tag{6}$$

You may notice that the update rule of Eq. 5 differs than the one stated in the textbook (cf. Eq. 2.12 in the textbook). In the next few questions, we will investigate this difference and show that *in expectation* the two are equivalent.

Show that for the following equation is true for any value of the constant B,

$$\nabla_{H(a)} J(\pi; H) = \mathbb{E}[(r_t - B)(\mathbb{1}_{(a=a_t)} - \pi_t(a))]. \tag{7}$$

3.4 3pts

We will refer to the term B as the baseline. Under certain conditions, Eq. 7 also holds true if B varies with time (i.e., it is no longer a constant). Mathematically state and derive this condition under which Eq. 7 holds true even if B varies with time.

3.5 2pts

The textbook suggests using the average of past rewards $\overline{R}_t \doteq \frac{\sum_1^{t-1} r_t}{(t-1)}$ as the baseline term. Show that this term satisfies the condition derived in your answer to Problem 3.4.

3.6 5pts

Despite several choices of baseline B result in identical expected gradient, their resulting performance for solving bandit differs. For example, Fig. 2.5 (textbook) shows the advantage of selecting $B = \overline{R}_t$ over $B = 0$. The reason certain baseline values are better than others is due to the fact that while B (under the conditions that you derived in Problem 3.3) does not change the expected value of the gradient $\nabla_H J(\pi; H)$, it *does* change its variance. Compute the variance of the sample gradient, i.e.,

$$\mathbb{V}[(r_t - B)(\mathbb{1}_{(a=a_t)} - \pi_t(a))]. \tag{8}$$

Recall that for a random variable X, the variance can be computed as: $\mathbb{V}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2$.

Hint: For ease of analysis, let us denote

$$\{(r_t - B)(\mathbb{1}_{(a=a_t)} - \pi_t(a))\} \doteq X \qquad \text{, and}$$
$$(\mathbb{1}_{(a=a_t)} - \pi_t(a)) \doteq g(a, a_t, \pi_t) = g.$$

Show that

$$\mathbb{V}(X) = \mathbb{E}[r_t^2 g^2] - 2B\mathbb{E}[r_t g^2] + B^2\mathbb{E}[g^2] - (\mathbb{E}[r_t g])^2.$$

3.7 8pts

Derive B for which the variance $\mathbb{V}[(r_t - B)(\mathbb{1}_{(a=a_t)} - \pi_t(a))]$ is lowest.

Hint: Solve the optimization problem

$$\arg\min_B \mathbb{V}[(r_t - B)(\mathbb{1}_{(a=a_t)} - \pi_t(a))],$$

to arrive at the optimal baseline term

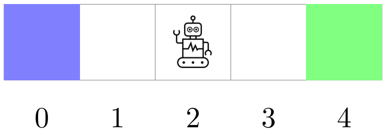
$$B_* = \frac{\mathbb{E}[r_t(\mathbb{1}_{(a=a_t)} - \pi_t(a))^2]}{\pi_t(a)(1 - \pi_t(a))}.$$

3.8 2pts

In stochastic gradient algorithms, sample gradients with lower variance are preferred. As $B = \overline{R}_t$ has a lower variance than $B = 0$, it results in superior performance in the experiments reported in the book. However, as you showed in Problem 3.5, $B = \overline{R}_t$ is not the baseline term for which the variance is lowest. Why might have the authors selected $B = \overline{R}_t$ instead of the B which results in the lowest variance?

4. Markov Property 10pts

Consider a robot navigating the following discrete line, which we call line world.



The robot continuously shuttles between the two shaded grids. Let x_t denote the robot position (i.e., the grid index) at time t.

Hint: If the robot starts at grid 2, it can next end up in either grid 1 or grid 3 depending on which shaded grid it is navigating towards.

4.1 2pts

Let $(X_0, X_1, ...)$ be a sequence of random variables. What is the necessary condition for this sequence to satisfy the Markov property?

4.2 2pts

Is the sequence $(x_0, x_1, ...)$ Markovian? Why or why not?

4.3 2pts

Does your previous answer depend on the starting position of the robot, x_0?

Let s_t denote the state of the line world at time t.

4.4 2pts

Define s_t such that the sequence $(s_0, s_1, ...)$ is Markovian.

4.5 2pts

Provide an alternate definition of the state such that the sequence $(s_0, s_1, ...)$ is Markovian.

5. Value Functions and Optimal Policies 10pts

Recall that for a Markov decision process (S, A, T, R, γ) with policy $\pi(a|s)$, the state-only and state-action value functions are defined, respectively, as follows

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right], \tag{9}$$

$$q_\pi(s, a) \doteq \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]. \tag{10}$$

Let α and β be two policies defined as follows,

$$\alpha = \arg\max_\pi v_\pi(s), \tag{11}$$

$$\beta = \arg\max_\pi q_\pi(s, a). \tag{12}$$

5.1 5pts

Show that the policy that maximizes state-only value function, has the maximum possible value of state-action value function for all states and actions, i.e.,

$$q_\alpha(s, a) = \max_\pi q_\pi(s, a) \qquad \forall s \in S \text{ and } a \in A \tag{13}$$

5.2 4pts

Show that the policy that maximizes state-action value function, has the maximum possible value of state-only value function for all states, i.e.,

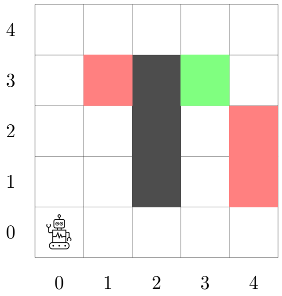
$$v_\beta(s) = \max_\pi v_\pi(s) \qquad \forall s \in S. \tag{14}$$

5.3 1pts

Are the two policies α and β identical?

6. Autonomous Navigation 15pts

Consider a robot situated in the following grid world.



Grids in black denote walls and those in red denote danger. The numbers along the grid represent the horizontal and vertical grid indices, i.e., the bottom left grid corresponds to grid# (0, 0). The robot can choose to *wait* or move to any of its neighboring grids that are not obstacles or walls (i.e., the boundary of the grid world). However, the robot's actuators are imperfect. The actions *left*, *right*, *up*, *down* succeed with probability 0.9. The actions to move *diagonally* succeed with probability 0.7. The action to *wait* always succeeds. If an action does not succeed, the robot randomly moves to any of its neighboring grids that is not an obstacle. The robot's task is to reach the green grid as soon as possible, while avoiding danger. The task terminates once the robot reaches its goal (green grid) or enters the danger zone (any of the red grids). In order to program the robot to do its task, let us explore the use of Markov Decision Processes (MDPs).

6.1 4pts

Define the state and action to represent the robot's task as an MDP. Report the size of MDP's state and action spaces.

6.2 5pts

Define the transition and reward functions for this MDP. While designing the reward, follow the principle: "the reward should convey *what* you want achieved instead of *how* you want it achieved."

6.3 2pts

Is the choice of reward function unique? If yes, provide justification. If no, provide an alternate definition of the reward function.

6.4 4pts

This discrete grid world can be viewed as a (highly simplified) approximation of autonomous driving, where the green grid denotes the car's goal, the red grids denote other parked cars, and white grids denote the road. It goes without saying that the actual problem of autonomous driving is significantly more complicated and requires a more complex MDP model. Model this actual problem of autonomous driving as an MDP. Making assumptions during this modeling exercise is okay but clearly state any assumptions that you make. Report the definition of state, action, transition, and reward for this MDP model.

7. Stochastic Gradient Descent 10pts

Consider the classical supervised learning setting. Given a supervised dataset $\mathbb{D}$ of (x, y)-pairs, the goal is to arrive at a parameterized function

$$f(x; \theta) = \hat{y}$$

that best predicts y given x by minimizing the objective

$$J(\theta) = \sum_{(x,y)\in\mathbb{D}} \|y - f(x; \theta)\|^2. \tag{15}$$

For simplicity, assume that x and y are scalars.

7.1 3pts

Provide pseudocode of an algorithm to learn $f(x; \theta)$ using (vanilla) gradient descent.

7.2 3pts

Provide pseudocode of an algorithm to learn $f(x; \theta)$ using stochastic gradient descent.

7.3 2pts

Which conditions motivate the use of stochastic gradient descent?

7.4 2pts

Which algorithm (gradient descent or stochastic gradient descent) exhibits higher variance?