ECE239AS: Problem Set

Instructor: Prof. Lin Yang

TAs: J. Kenanian, Q. Lu, C. Talegaonkar

April 24, 2020

- You need to choose and solve 10 problems among the problems below. In the list of problems you choose, you must pick at least one problem among Exercises 1-6, one problem among Exercises 7-10, one problem among Exercises 11-15, one problem among Exercises 16-20, two problems among Exercises 21-30.
- All exercises have the same weight.
- Submission deadline is Tuesday 05/05 at 23:59 PST.
- Submission will be on CCLE.
- Students are strongly encouraged to type their answers in LaTeX, but if they absolutely cannot do it, other methods will be accepted (included scanned hand-written documents, at the condition they are clearly readable).
- This work is individual, no collaboration is accepted.

Exercise 1

1. In ε-greedy action selection, for the case of two actions and $\varepsilon = 0.5$, what is the probability that the greedy action is selected?

2. A simple example. Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ε-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1$, $R_1 = 1$, $A_2 = 2$, $R_2 = 1$, $A_3 = 2$, $R_3 = 2$, $A_4 = 2$, $R_4 = 2$, $A_5 = 3$, $R_5 = 0$. On some of these time steps the ε case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?

3. Suppose you are given a k-arm bandit problem. The reward of each arm satisfies

$$\forall i > 1 : \quad \mathbb{E}[R_1] = r^* > \mathbb{E}[R_i] = 0, \quad \text{and} \quad \mathrm{Var}(R_1) = \mathrm{Var}(R_i) = \sigma^2.$$

   Let Q denote the optimal action-value function of the problem (what is Q?). Suppose one plays an ϵ-greedy policy based on Q for T rounds. Compute, with probability at least $1 - \delta$ for some $\delta \in (0, 1)$, how much total rewards can one obtain after the T rounds.

Exercise 2

Suppose you face a 2-armed bandit task whose true action values change randomly from time step to time step. Specifically, suppose that, for any time step, the true values of actions 1 and 2 are respectively 0.1 and 0.2 with probability 0.5 (case A), and 0.9 and 0.8 with probability 0.5 (case B). If you are not able to tell which case you face at any step, what is the maximum expected reward you can achieve and how should you behave to achieve it? Now suppose that on each step you are told whether you are facing case A or case B (although you still don't know the true action values). This is an associative search task. What is the best expected reward you can achieve in this task, and how should you behave to achieve it?

Exercise 3

We denote by $H_t(a)$ the preference for each action a at time t. The larger the preference, the more often that action is taken, but the preference has no interpretation in terms of reward. Only the relative preference. The action probabilities follow a soft-max distribution, defined as follows:

$$\mathbb{P}(A_t = a) = \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} = \pi_t(a)$$

where $\pi_t(a)$ represents the probability of taking action a at time t. Initially, all action preferences are the same (e.g., $H_1(a) = 0$, for all a), so that all actions have an equal probability of being selected. The preference can change to denote different policies.

1. Show that in the case of two actions, the soft-max distribution is the same as that given by the logistic, or sigmoid, function often used in statistics and artificial neural networks.

2. Suppose at some time t, a preference-based policy π_t chooses action a_1 with probability 0.9 and action a_2 with probability 0.1. Give an assignment for H_t satisfying the probability distribution π_t. Let R_1 and R_2 be expected rewards of a_1 and a_2 respectively. What is the expected reward (denoted as V_t) if play the policy π_t for one step?

3. Derive the gradient of V_t with respect to $H_t(a_1)$ and $H_t(a_2)$ (denote V_t as a function of $\theta := (H_t(a_1), H_t(a_2))$, derive $\nabla_\theta V_t$).

Exercise 4

An n-armed bandit instance I, where $n \geq 2$, has $p_1, p_2, \ldots, p_n \in (0, 1)$ as the mean rewards of its arms. Each arm implements a Bernoulli distribution (that is, returns 0-1 rewards).

1. Let $r^0, r^1, \ldots, r^{T-1}$ be the rewards obtained in the first T pulls, where $T \geq 1$. Let
$$x^T = r^0 + r^1 + \cdots + r^{T-1}.$$
Consider an algorithm that picks an arm uniformly at random at each round and pulls it (you may think of it as ε-greedy sampling with $\varepsilon = 1$). If this algorithm is executed, what is the variance of x^T? (Recall the variance is defined by $\mathbb{E}[(x^T)^2] - (\mathbb{E}[x^T])^2$)

2. Let
$$y^T = (1 - r^0) + (1 - r^1) + \cdots + (1 - r^{T-1}) = T - x^T.$$
While x^T is the total number of 1-rewards in the first T pulls, y^T is the total number of 0-rewards in the first T pulls. Let
$$z^T = \max(x^T, y^T),$$
that is, z^T counts the total number of the more-frequent reward value in the first T pulls. We are interested in maximizing $\mathbb{E}[z^T]$.
Suppose one has knowledge of I: that is, one knows $p_1, p_2, \ldots, p_n$. What algorithm L_* must one apply in order to maximize $\mathbb{E}[z^T]$?

Exercise 5

Consider a 2-armed bandit instance whose arms a_1 and a_2 have means p_1 and p_2, respectively, with $1 > p_1 > p_2 > 0$. Each arm yields i.i.d. Bernoulli rewards with the corresponding mean. Hence, each reward obtained is either 0 or 1.
An algorithm L is applied to this bandit instance. At every step, L pulls whichever arm has obtained the least number of 0-rewards up to then, breaking ties uniformly at random. Thus, the very first pull is equally likely to come from a_1 and a_2. Suppose a_2 was pulled and it gives a 1-reward, then again both arms are equally likely to be picked. If a_1 is now pulled, and it gives a 0-reward, then a_2 will be pulled next, and repeatedly until it gives a 0-reward. At this point, both arms will again have an equal number of 0-rewards, and therefore be equally likely to be pulled, and so on. For $T \geq 1$, let z^T denote the number of 0-rewards obtained in the first T pulls.

- What is $\mathbb{E}[z^2]$?
- What is $\lim_{T\to\infty} \frac{\mathbb{E}[z^T]}{T}$?
- Let R^T denote the cumulative regret after T pulls. What is $\lim_{T\to\infty} \frac{\mathbb{E}[R^T]}{T}$? Hint: you can express R^T in terms of z^T, and then use your answer from the previous question.

Exercise 6

Unbiased Constant-Step-Size Trick. Using sample averages to estimate action values is a way to avoid the initial bias obtained when considering constant step sizes (see Chapter 2.5 of Reinforcement Learning, R. Sutton and G. Barto). However, sample averages are not a completely satisfactory solution because they may perform poorly on non-stationary problems. A possible way to avoid the bias of constant step sizes while retaining their advantages on non-stationary problems is to use the step size

$$\beta_n = \frac{\alpha}{\overline{o}_n}$$

to process the n-th reward for a particular action a, where $\alpha > 0$ is a conventional constant step size, and o_n is a trace of one that starts at 0:

$$\overline{o}_n = \overline{o}_{n-1} + \alpha(1 - \overline{o}_{n-1})$$

for $n \geq 0$ and with $\overline{o}_0 = 0$.
Show that Q_n is an exponential recency-weighted average without initial bias. Hint: you can carry out an analysis similar to the one conducted in Chapter 2.5 of Reinforcement Learning, R. Sutton and G. Barto.

Exercise 7

Policy Evaluation. Consider the 4×4 grid world shown in Figure 1.

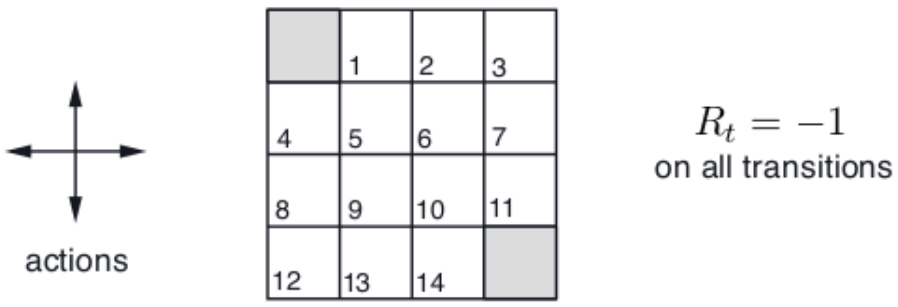

Figure 1: Grid world for Exercise 7

The nonterminal states are $S = \{1, 2, \ldots, 14\}$. There are four actions possible in each state, $A = \{\text{up}, \text{down}, \text{right}, \text{left}\}$, which deterministically cause the corresponding state transitions,

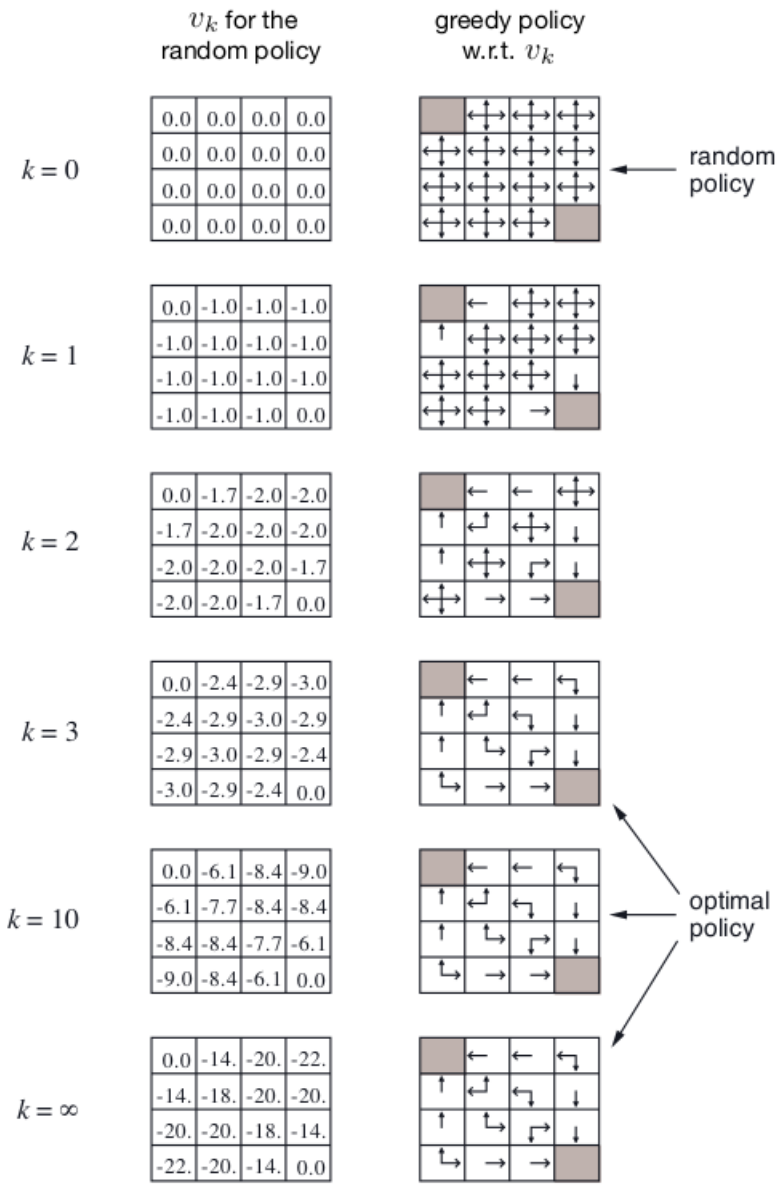

Figure 2: Convergence of iterative policy evaluation on a small gridworld. The left column is the sequence of approximations of the state-value function for the random policy (all actions equally likely). The right column is the sequence of greedy policies corresponding to the value function estimates (arrows are shown for all actions achieving the maximum, and the numbers shown are rounded to two significant digits). The last policy is guaranteed only to be an improvement over the random policy, but in this case it, and all policies after the third iteration, are optimal.

except that actions that would take the agent off the grid in fact leave the state unchanged. Thus, for instance, $p(6, -1|5, \text{right}) = 1$, $p(7, -1|7, \text{right}) = 1$, and $p(10, r|5, \text{right}) = 0$ for all

$r \in \mathcal{R}$. This is an undiscounted, episodic task. The reward is -1 on all transitions until the terminal state is reached. The terminal state is shaded in the figure (although it is shown in two places, it is formally one state). The expected reward function is thus $r(s, a, s') = -1$ for all states s, s' and actions a. Suppose the agent follows the equiprobable random policy (all actions equally likely). The left side of Figure 2 shows the sequence of value functions $\{v_k\}$ computed by iterative policy evaluation. The final estimate is in fact v_π, which in this case gives for each state the negation of the expected number of steps from that state until termination.

If π is the equiprobable random policy, what is $q_\pi(11, \text{down})$? What is $q_\pi(7, \text{down})$?

Exercise 8
In the situation exposed in Exercise 7, suppose a new state 15 is added to the grid world just below state 13, and its actions, left, up, right, and down, take the agent to states 12, 13, 14, and 15, respectively. Assume that the transitions from the original states are unchanged.

What is $v_\pi(15)$ for the equiprobable random policy? Now suppose the dynamics of state 13 are also changed, such that action down from state 13 takes the agent to the new state 15. What is $v_\pi(15)$ for the equiprobable random policy in this case?

Exercise 9
Policy iteration Jack manages two locations for a nationwide car rental company. Each day, some number of customers arrive at each location to rent cars. If Jack has a car available, he rents it out and is credited \$10 by the national company. If he is out of cars at that location, then the business is lost. Cars become available for renting the day after they are returned. To help ensure that cars are available where they are needed, Jack can move them between the two locations overnight, at a cost of \$2 per car moved. We assume that the number of cars requested and returned at each location are Poisson random variables, meaning that the probability that the number is n is $\frac{\lambda^n}{n!}e^{-\lambda}$, where λ is the expected number. Suppose λ is 3 and 4 for rental requests at the first and second locations and 3 and 2 for returns. To simplify the problem slightly, we assume that there can be no more than 20 cars at each location (any additional cars are returned to the nationwide company, and thus disappear from the problem) and a maximum of 5 cars can be moved from one location to the other in one night. We take the discount rate to be $\gamma = 0.9$ and formulate this as a continuing finite MDP, where the time steps are days, the state is the number of cars at each location at the end of the day, and the actions are the net numbers of cars moved between the two locations overnight. Figure 3 shows the sequence of policies found by policy iteration starting from the policy that never moves any cars.

1. The policy iteration algorithm given on page 80 of Reinforcement Learning, R. Sutton and G. Barto, has a subtle bug in that it may never terminate if the policy continually switches between two or more policies that are equally good. Give a modified pseudo-code so that convergence is guaranteed.

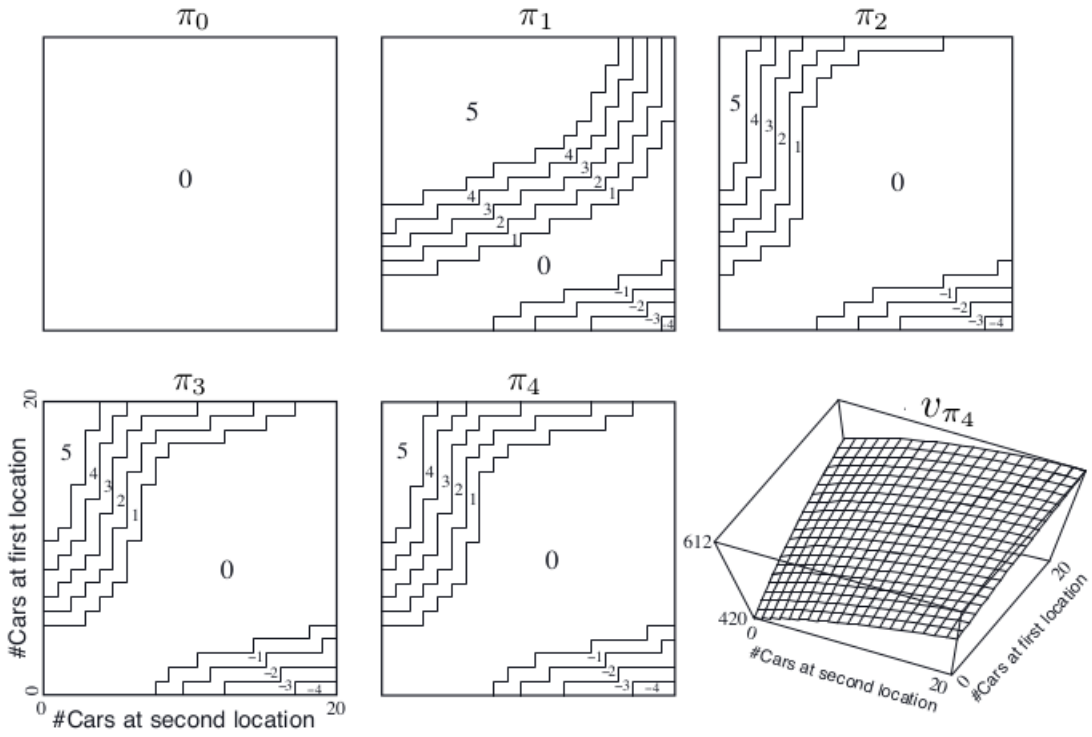

Figure 3: The sequence of policies found by policy iteration on Jack's car rental problem, and the final state-value function. The first five diagrams show, for each number of cars at each location at the end of the day, the number of cars to be moved from the first location to the second (negative numbers indicate transfers from the second location to the first). Each successive policy is a strict improvement over the previous policy, and the last policy is optimal.

2. How would policy iteration be defined for action values? Give a complete algorithm for computing q_*, analogous to that on page 80 of Reinforcement Learning, R. Sutton and G. Barto for computing v_*.

Exercise 10

In the situation exposed in Exercise 9, suppose you are now restricted to considering only policies that are ε-soft, meaning that the probability of selecting each action in each state, s, is at least $\frac{\varepsilon}{|\mathcal{A}(s)|}$. Describe qualitatively the changes that would be required in each of the steps 3, 2, and 1, in that order, of the policy iteration algorithm for v_* given on page 80 of Reinforcement Learning, R. Sutton and G. Barto.

Exercise 11

Suppose a Q-learning agent, with fixed α and discount γ, was in state 34, did action 7, received reward 3, and ended up in state 65. What value(s) get updated? Give an expression for the new value. (Be as specific as possible.)

Exercise 12
Explain what happens in reinforcement learning if the agent always chooses the action that maximizes the Q-value. Suggest two ways to force the agent to explore.

Exercise 13
Consider four different ways to derive the value of α_k from k in Q-learning (note that for Q-learning with varying α_k, there must be a different count k for each state-action pair).

1. Let $\alpha_k = \frac{1}{k}$.
2. Let $\alpha_k = \frac{10}{9+k}$.
3. Let $\alpha_k = 0.1$.
4. Let $\alpha_k = 0.1$ for the first 10,000 steps, $\alpha_k = 0.01$ for the next 10,000 steps, $\alpha_k = 0.001$ for the next 10,000 steps, $\alpha_k = 0.0001$ for the next 10,000 steps, and so on.

Which of these will converge to the true Q-value in theory? Which converges to the true Q-value in practice (i.e., in a reasonable number of steps)? Try it for more than one domain. Which can adapt when the environment adapts slowly?

Exercise 14
Suppose your friend presented you with the following example where SARSA(λ) seems to give non intuitive results. There are two states, A and B. There is a reward of 10 coming into state A and no other rewards or penalties. There are two actions: left and right. These actions only make a difference in state B. Going left in state B goes directly to state A, but going right has a low probability of going into state A. In particular:

- $P(A|B, \text{left}) = 1$; reward is 10
- $P(A|B, \text{right}) = 0.01$; reward is 10
- $P(B|B, \text{right}) = 0.99$; reward is 0
- $P(A|A, \text{left}) = P(A|A, \text{right}) = 0.999$,
  $P(B|A, \text{left}) = P(B|A, \text{right}) = 0.001$; reward is 0

This is small enough that the eligibility traces will be close enough to zero when state B is entered. γ and λ are 0.9 and α is 0.4. Suppose that your friend claimed that that $Q(\lambda)$ does not work in this example, because the eligibility trace for the action right in state B ends up being bigger than the eligibility trace for action left in state B and the rewards and all of the parameters are the same. In particular, the eligibility trace for action right will be about 5 when it ends up entering state A, but it be 1 for action left. Therefore, the best action will be to go right in state B, which is not correct.

What is wrong with your friend's argument? What does this example show?

Exercise 15
In SARSA with linear function approximators, if you use linear regression to minimize $r + \gamma Q_w(s', a') - Q_w(s, a)$, you get a different result than we have here. Explain what you get and why what is described in the text may be preferable (or not).

Exercise 16
Consider an MDP with a single nonterminal state and a single action that transitions back to the nonterminal state with probability p and transitions to the terminal state with probability $1 - p$. Let the reward be $+1$ on all transitions, and let $\gamma = 1$. Suppose you observe one episode that lasts 10 steps, with a return of 10. What are the first-visit and every-visit estimators of the value of the nonterminal state?

Exercise 17
Prove that for a given policy π in an infinite-horizon MDP, the n-step SARSA target is a less-biased (in absolute value) estimator of the true state-action value function $Q_\pi(s_t, a_t)$ than is the 1-step SARSA target. Assume that $n \geq 2$ and $\gamma < 1$. Further, assume that the current value estimate $\hat{q}$ is uniformly biased across the state-action space (that is, $\text{Bias}(\hat{q}(s, a)) = \text{Bias}(\hat{q}(s_0, a_0))$ for all states $s, s_0 \in S$ and all actions $a, a_0 \in A$. You need not assume anything about the specific functional form of $\hat{q}$.

Exercise 18
Consider an undiscounted Markov Reward Process with two states A and B. The transition matrix and reward function are unknown, but you have observed two sample episodes:
$A + 3 \rightarrow A + 2 \rightarrow B - 4 \rightarrow A + 4 \rightarrow B - 3 \rightarrow$ terminate
$B - 2 \rightarrow A + 3 \rightarrow B - 3 \rightarrow$ terminate
In the above episodes, sample state transitions and sample rewards are shown at each step, e.g. $A + 3 \rightarrow A$ indicates a transition from state A to state A, with a reward of $+3$.

1. Using first-visit Monte-Carlo evaluation, estimate the state-value function $V(A), V(B)$
2. Using every-visit Monte-Carlo evaluation, estimate the state-value function $V(A), V(B)$
3. Draw a diagram of the Markov Reward Process that best explains these two episodes (i.e., the model that maximizes the likelihood of the data -although it is not necessary to prove this fact). Show rewards and transition probabilities on your diagram.
4. Define the Bellman equation for a Markov reward process.
5. Solve the Bellman equation to give the true state-value function V(A),V(B). Hint: solve the Bellman equations directly, rather than iteratively.
6. What value function would batch TD(0) find, i.e., if TD(0) was applied repeatedly to these two episodes?

7. What value function would batch TD(1) find, using accumulating eligibility traces?

Exercise 19

For this question, assume that the MDP has a finite number of states. Please pick True or False for each of the following statements and explain the reason in detail.

1. True or False. For an MDP (S, A, T, γ, R), if we only change the reward function R the optimal policy is guaranteed to remain the same.

2. True or False. Value iteration is guaranteed to converge if the discount factor (γ) satisfies $0 < \gamma < 1$.

3. True or False. Policies found by value iteration are superior to policies found by policy iteration.

Exercise 20

In many situations such as healthcare or education, we cannot run any arbitrary policy and collect data from running those policies for evaluation. In these cases, we may need to take data collected from following one policy and use it to evaluate the value of a different policy. The equality proved in the following exercise can be an important tool for achieving this. The purpose of this exercise is to get familiar on how to compare the value of different policies, π_1 and π_2, on a fixed horizon MDP. A fixed horizon MDP is an MDP where the agent's state is reset after H time steps; H is called the horizon of the MDP. There is no discount (i.e., $\gamma = 1$) and policies are allowed to be non-stationary, i.e., the action identified by a policy depends on the time step in addition to the state. Let $x_t \sim \pi$ denote the distribution over states at time step t (for $1 \leq t \leq H$) upon following policy π and $V_t^\pi(x_t)$ denote the value function of policy π in state x_t and time step t, and $Q_t^\pi(x_t, a)$ denote the corresponding Q value associated to action a. As a clarifying example, we denote $E_{x_t \sim \pi_1} V(x_t)$ to represent the average value of the value function $V(\cdot)$ over the states at time step t encountered upon following policy π_1.
Show the following:

$$V_1^{\pi_1}(x_1) - V_1^{\pi_2}(x_1) = \sum_{t=1}^{H} E_{x_t \sim \pi_2}[Q_t^{\pi_1}(x_t, \pi_1(x_t, t)) - Q_t^{\pi_1}(x_t, \pi_2(x_t, t))].$$

Exercise 21

Consider the following linear approximation of the $Q_t(s, a)$ state-action value function at time t:

$$Q_t(s, a) = \theta_t^T \phi_{s,a} = \sum_{i=1}^{n} \theta_t^i \phi_{s,a}^i$$

where θ_t^i and $\phi_{s,a}$ denote the i^{th} component of the corresponding n dimensional vectors. Explain how the features vector $\phi_{s,a}$ and the parameters vector θ_t^i should be constructed in order to reproduce the tabular case of the Q function.

Exercise 22

One can generate high dimensional features for a state, to better represent the effect of the states on Value functions and Action Value functions. One way to do so is using a higher order polynomial basis. As an example, suppose a reinforcement learning problem has states with two numerical dimensions. For a single representative state s, let its two numbers be $s_1, s_2 \in \mathbb{R}$. You might choose to represent s simply by its two state dimensions, so that $x(s) = (s_1, s_2)^T$, but this representation won't take into account the interactions between the two dimensions. One can hence use a higher order (polynomial basis) feature representation $x(s) = (1, s_1, s_2, s_1 s_2)^T$. The first feature (constant) allows the representation of affine functions in the original state numbers, and the final product feature, $s_1 s_2$, enables interactions to be taken into account.

Generalizing to k dimensions, each n order polynomial basis x_i can be written as

$$x_i(s) = \prod_{j=1}^{k} s_j^{c_{i,j}}.$$

What n and $c_{i,j}$ can produce the feature vectors $x(s) = (1, s_1, s_2, s_1 s_2, s_1^2, s_2^2, s_1^2 s_2, s_2^2 s_1, s_1^2 s_2^2)^T$?

Exercise 23

Tile Coding is a useful way to generate coarse multi-dimensional feature vectors for continuous state spaces (Refer SB section 9.5.4).

1. Give two advantages of asymmetrically offset tilings.

2. Assume that for a 2-dimensional continuous state space, one of two state dimensions is more likely to have an effect on the value function than is the other. What kind of tilings could be used to take advantage of this prior knowledge?

Exercise 24

Consider the MDP shown below in Figure 4, with states s_1, s_2, and s_3, and actions a_1 and a_2. The MDP is episodic. Each episode has an equal probability $\frac{1}{3}$ of starting in any of the three states. From s_1, action a_1 deterministically takes the agent to s_2, and action a_2 deterministically takes the agent to s_3. Transitions out of s_2 and s_3 terminate the episode and reset the agent in one of the three states. No discounting is used in the calculating values. In the figure, transitions resulting from action a_1 are shown with solid arrows; those from action a_2 are shown with dotted arrows. The reward for each transition is shown along with the corresponding arrow.

Suppose an agent follows policy π^{111}, such that

$$\pi^{111}(s_1) = a_1, \pi^{111}(s_2) = a_1, \pi^{111}(s_3) = a_1.$$

The agent uses a linear generalisation scheme to approximate the value function of π^{111}. It has a single scalar parameter θ, and the features corresponding to the states are: $\phi(s_1) = 1, \phi(s_2) = -1, \phi(s_3) = 2$. In other words, the agent intends to approximate

$$V^{\pi^{111}}(s_1) \approx \theta, V^{\pi^{111}}(s_2) \approx -\theta, V^{\pi^{111}}(s_3) \approx 2\theta.$$

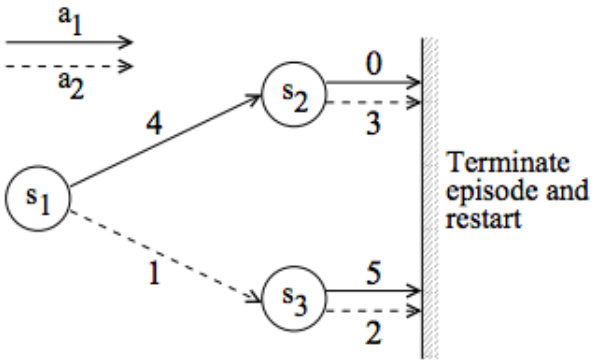

Figure 4: MDP for Exercise 24

If the agent applies TD(1) with this linear function approximation scheme, while following π^{111} and continuing to anneal the learning rate harmonically, to what will θ converge?

Exercise 25

In an MDP with three states, s_1, s_2, and s_3, a policy π is such that $V^\pi(s_1) = 4, V^\pi(s_2) = 6, V^\pi(s_3) = -5$. Under π, the steady-state probability of being in s_1 is $\frac{1}{2}$, and the probabilities of being in s_2 and s_3 are each $\frac{1}{2}$. A learning agent seeks to approximate

$$V^\pi(\cdot) \approx w_1 f(\cdot) + w_2$$

where $f(\cdot)$ is a state feature, and w_1 and w_2 are the weights to learn. Features for the three states are as follow: $f(s_1) = 2, f(s_2) = 1, f(s_3) = -1$.

1. If the agent performs Monte Carlo policy evaluation with this linear generalisation scheme, to what values will w_1 and w_2 converge?

2. How would your answer change if the agent uses linear TD(0) instead for policy evaluation?

Exercise 26

Let's examine the expressive power of 1-dimensional tile coding: that is, tile coding that uses a separate set of tilings for each dimension. We examine the complexity of functions over two variables that can be represented using 1-dimensional tile coding.

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ be $m \geq 1$ distinct point(s) in $\mathbb{R}^2$, and let these points be associated with function values $f(x_1, y_1), f(x_2, y_2), \ldots, f(x_m, y_m) \in \mathbb{R}$, respectively. In short, we have described a function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defined on m points.

We say that f can be 1-tile-coded if there exists a 1-dimensional tile coding scheme $T : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that for all $i \in \{1, 2, \ldots, m\}$, $T(x_i, y_i) = f(x_i, y_i)$. Recall that for point $(x, y) \in \mathbb{R}^2$, $T(x, y)$ is the sum of the weights of the tiles that are active for (x, y) in each dimension. Along each dimension, T may employ any number of regularly-spaced tilings, with any tile width (common to all the tiles in that dimension), and any origin. The real-valued weights assigned by T to the individual tiles in the x and y dimensions can be arbitrary.

Consider the following statement: **For every set of m distinct point(s), every function f over the points can be 1-tile-coded.** For which values of $m \in \{1, 2, \ldots\}$ is this statement true, and for which ones is it false? Justify your answer.

Exercise 27

Suppose, perhaps as part of a larger MDP, there are two states whose estimated values are of the functional form w and $2w$, where the parameter vector w consists of only a single component w. This occurs under linear function approximation if the feature vectors for the two states are each simple numbers (single-component vectors), in this case 1 and 2. In the first state, only one action is available, and it results deterministically in a transition to the second state with a reward of 0, as shown in Figure 5. Suppose initially $w = 10$, the discount factor $\gamma = 1$ and the step size α for TD update is 0.1.

Figure 5: Figure for Exercise 27

1. What is the TD error after first time step?
2. Derive a TD(0) update rule for w w.r.t time steps, i.e., w_t and comment on the convergence of w_t.
3. How can you ensure convergence for this update rule?
4. Does the convergence of w_t depend on α?

Exercise 28

Until now we dealt with value function approximation, but one can also use function approximation for the Q function. Assume a simple model that Q is represented by a function approximator that induces some noise on the estimates of the true Q.

More specifically, let us assume that the currently stored Q-values, denoted by $Q^{\text{approx}}(s', \hat{a})$, represent the true Q values denoted by $Q^{\text{target}}(s', \hat{a})$ corrupted by a noise term $Y_{s'}^{\hat{a}}$. This can be written as

$$Q^{\text{approx}}(s', \hat{a}) = Q^{\text{target}}(s', \hat{a}) + Y_{s'}^{\hat{a}}.$$

We assume the noise $Y_{s'}^{\hat{a}}$ to be uniformly distributed in the interval $[-\epsilon, \epsilon]$. This would cause a disturbance in the update equation for the Q function. We denote this disturbance as Z_s, which is defined as:

$$Z_s = \gamma(\max_{\hat{a}} Q^{\text{approx}}(s', \hat{a}) - \max_{\hat{a}} Q^{\text{target}}(s', \hat{a})).$$

Let n denote the number of actions applicable at state s'. If all n actions share the same target Q value, i.e., $\exists\, q : \forall \hat{a} : q = Q^{\text{target}}(s', \hat{a})$, then prove that $\mathbb{E}[Z_s] = \gamma\epsilon\frac{n-1}{n+1}$. $\mathbb{E}[Z_s]$ is often referred to as the average overestimation in Q value.

Exercise 29

Consider a discounted MDP $(\mathcal{S}, \mathcal{A}, P, \gamma, r)$, where $\mathcal{S}, \mathcal{A}$ are finite set of states and actions, $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the probability transition matrix, which is **unknown**, $\gamma \in (0, 1)$ is the discount factor, and $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ be a **known** function. Suppose you are given m samples from each state-action (s, a), i.e.,

$$s_{s,a}^{(1)}, s_{s,a}^{(2)} \ldots, s_{s,a}^{(m)},$$

where for each i,

$$s_{s,a}^{(i)} \sim P(\cdot|s, a).$$

1. Let $\Pi = \{\pi : \mathcal{S} \to \mathcal{A}\}$, determine the size of Π.
2. For a policy $\pi \in \Pi$, design an algorithm to use the samples give an estimate $\hat{V}^{\pi}$ for the value function V^{π}. Suppose you need $\|\hat{V}^{\pi} - V^{\pi}\|_{\infty} \leq \epsilon$ with probability at least 0.9 for $\epsilon \in (0, 1)$, determine the minimum m (as a function of $|\mathcal{S}|, |\mathcal{A}|, \gamma$).
3. Design a method to estimate V^* and express its ℓ_{∞} error in terms of m.

Exercise 30

Consider a discounted MDP $(\mathcal{S}, \mathcal{A}, P, \gamma, r)$, where $\mathcal{S}, \mathcal{A}$ are finite set of states and actions, $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the probability transition matrix, which is **unknown**, $\gamma \in (0, 1)$ is the discount factor, and $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ be a **known** function. Now, suppose the probability matrix satisfies assumption

$$\forall s, a, s' : \quad P(s'|s, a) = \phi(s, a)^{\top}\psi(s')$$

where $\phi(s, a) \in \mathbb{R}^d$ is a **known** d-dimensional vector and $\psi(s')$ is **unknown**. Suppose you can select a set of state action pairs $\Omega \subset \mathcal{S} \times \mathcal{A}$ to query m samples from each state-action $(s, a) \in \Omega$, i.e.,

$$s_{s,a}^{(1)}, s_{s,a}^{(2)} \ldots, s_{s,a}^{(m)},$$

where for each i,

$$s_{s,a}^{(i)} \sim P(\cdot|s, a).$$

1. What is the dimension of

$$\mathcal{V} := \text{span}\{Q^\pi : \quad \forall \pi \in \Pi\}?$$

   Give a base for the linear space $\mathcal{V}$.

2. Design an algorithm to estimate V^*: describe how you choose Ω, the algorithm, and the guarantee of the accuracy.