Exercises - Function Approximation

Reinforcement Learning

1. Consider the $\boldsymbol{w} \in \mathbb{R}^2$ and the criterion function

$$J(\boldsymbol{w}) = 6w_1^2 + 10w_1w_2 - 8w_1 + 6w_2^2 - 8w_2 + 3$$

   (a) Compute the gradient $\nabla J(\boldsymbol{w})$.

   (b) Starting in $\boldsymbol{w} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, take one step with gradient descent using step length $\alpha = 0.01$. Is $J(\boldsymbol{w})$ smaller after the update?

   (c) It can be of interest to write a short Python code that performs gradient descent on this function, to see that it converge to the minimizing $\boldsymbol{w}$. It should converge to very near the optimum within about 1000 updates.

   Remark: The optimal $\boldsymbol{w}$ in this case is $\boldsymbol{w} \approx \begin{bmatrix} 0.3636 \\ 0.3636 \end{bmatrix}$.

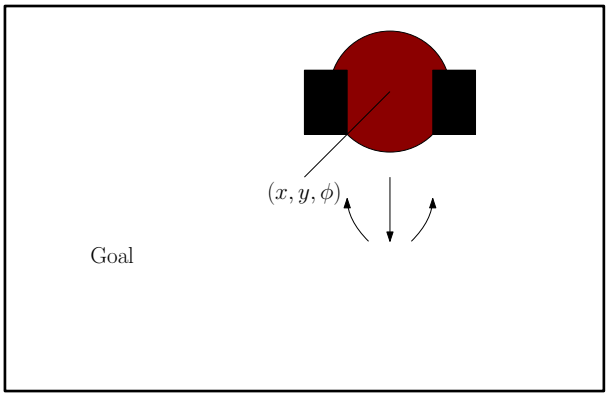
2. Consider the case when we use a linear function approximation of the state-value function,

$$\hat{v}(s, \boldsymbol{w}) = \sum_{i=1}^{d} w_i x_i(s) = \boldsymbol{w}^\top \boldsymbol{x}(s),$$

   where

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}, \quad \boldsymbol{x}(s) = \begin{bmatrix} x_1(s) \\ \vdots \\ x_d(s) \end{bmatrix}.$$

   Show that $\nabla \hat{v}(s, \boldsymbol{w}) = \boldsymbol{x}(s)$.



Figur 3.1: The robot

3. In this problem we consider the robot shown in Figure 3. The robots start in an arbitrary position and tries to get close to the goal. When it is close enough to the goal the episode terminates. The state of the robot is $s = \begin{bmatrix} x & y & \phi \end{bmatrix}$ where (x, y) determines the position of the robot, while ϕ determines its angle in radians. At each time-step the robot can choose between three actions:

   - a_c turns the robot clockwise $\pi/4$ radians.
   - a_f makes the robot move forward 1 m.

- a_a turns the robot anticlockwise $\pi/4$ radians.

This is an undiscounted task, so $\gamma = 1$.

An agent was used to collect data from one episode and the results are shown in Table 3, and in time step 4 a terminating state was reached.

t	s_t	a_t	r_{t+1}
0	$[5, 5, \pi]^\top$	a_f	-2.24
1	$[4, 5, \pi]^\top$	a_a	-2.24
2	$[4, 5, \frac{5\pi}{4}]^\top$	a_f	-1.33
3	$[3.29, 4.29, \frac{5\pi}{4}]^\top$	a_f	-0.72
4	$[2.59, 3.59, \frac{5\pi}{4}]^\top$		

Table 1: Experience from robot

(a) Compute the every-visit Monte-Carlo return G_t for time-step $t = 0, 1, 2, 3$. Would your answer be different if we instead used the first-visit Monte-Carlo return?

(b) We are now going to use function approximation to estimate $v_\pi(s)$ for the policy used to collect the experience in Table 3. We use a linear function approximator

$$\hat{v}(s, \boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x}(s)$$

where the features are given by

$$\boldsymbol{x}(s) = \begin{bmatrix} 1 & x & y & x^2 & y^2 & \phi \end{bmatrix}^\top.$$

We initialize our weights with

$$\boldsymbol{w} = \begin{bmatrix} 0 & 0 & 0 & -0.2 & -0.2 & 0 \end{bmatrix}^\top,$$

and use the step-size $\alpha = 0.01$. In this problem we use the MC-target in the update. What will $\boldsymbol{w}$ be after the first update? (That is, after the transition S_0, A_0, R_1, S_1).

(c) Consider exactly the same question as (b) but use the TD-target instead.

4. Consider a finite state-space $\mathcal{S} = \{1, \ldots, n\}$. We are going to use MC with linear function approximation. We will use $d = n$ features given by

$$x_i(s) = \begin{cases} 1 & \text{if } s = i \\ 0 & \text{otherwise} \end{cases}.$$

The updates of the weights when we use the MC-target are

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha[G_t - \hat{v}(S_t, \boldsymbol{w})]\boldsymbol{x}(S_t).$$

Show that, with these features, the MC-updates using function approximation are equivalent to the updates used in the tabular case (see equation 6.1 in textbook). (Equivalent in the sense that w_i will be the same as the estimated $V(s)$ for $s = i$ in the tabular method).

Remark: In general, linear function approximation with the features described in this problem is equivalent to using tabular methods. Hence, tabular methods can actually be seen as a special case of linear function approximation.

5. Consider an environment where $\mathcal{S} = \mathbb{R}$ and $\mathcal{A} = \mathbb{R}$. That is, both the state and action are scalars that can take on any real value.

The environment state evolves according to the linear difference equation

$$S_{t+1} = 2S_t + A_t + E_t,$$

where E_t is a stochastic disturbance. We assume that E_t is a white noise process, that is independent of S_t and A_t, with zero mean and variance σ^2. That is $\mathbb{E}[E_t] = 0$ and $\mathbb{E}[E_t^2] = \sigma^2$.

The reward is given by

$$R_{t+1} = -S_t^2 - A_t^2,$$

with the discount factor $\gamma = 0.8$. That is, the reward is maximized by getting both the state and action close to zero.

We will consider the policy

$$\pi(s) = -1.5s.$$

(a) We first assume that the dynamics of the environment is unknown, and the aim is to estimate the value function $v_\pi(s)$. To do this, we use the value function approximation

$$\hat{v}(s, \mathbf{w}) = w_1 s^2 + w_2, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

The following data was generated from the environment using the policy $\pi(s)$:

t	S_t	A_t	R_{t+1}
0	2.00	-3.00	-13.00
1	1.09	-1.64	-3.88
2	0.44		

Do two steps of semi-gradient TD(0) (see page 203 in the textbook) using this data. Initialize with $\mathbf{w} = \mathbf{0}$ and use the step length $\alpha = 0.1$.

(b) In this part we assume that there is no disturbance (i.e. the variance $\sigma^2 = 0$ so $E_t = 0$ for all t). That is, the environment is deterministic and the state evolves according to

$$S_{t+1} = 2S_t + A_t.$$

If the policy is $\pi(s) = -1.5s$, then the value function will be on the form

$$v_\pi(s) = c_1 s^2.$$

Determine what c_1 is in this case.

(c) Consider the same setting as in part 4b. Compute the action value function $q_\pi(s, a)$ and do one step of policy improvement.

What is the improved policy $\pi'(s)$?

(d) Let us now instead consider the case when $\sigma^2 = 0.1$. In this case the value function can be written as

$$v_\pi(s) = c_1 s^2 + c_2.$$

Determine what c_1 and c_2 are in this case.

Solutions

1.

(a) The gradient is given by

$$\nabla J(\boldsymbol{w}) = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \end{bmatrix} = \begin{bmatrix} 12w_1 + 10w_2 - 8 \\ 10w_1 + 12w_2 - 8 \end{bmatrix}.$$

(b) With $\boldsymbol{w} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ we get

$$\nabla J(\boldsymbol{w}) = \begin{bmatrix} 4 \\ 2 \end{bmatrix}.$$

So the gradient descent update is

$$\boldsymbol{w} \leftarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 0.01 \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.96 \\ -0.02 \end{bmatrix}.$$

We can see that this decreases $J(\boldsymbol{w})$ from 1 to 0.82.

(c) Example code:

```
import numpy as np
w = np.array([ 1.0,0.0])
alpha = 0.01
print("Initial:", w)
for i in range(0,1000):
    w[0] = w[0] - alpha*(12*w[0] + 10*w[1] - 8)
    w[1] = w[1] - alpha*(10*w[0] + 12*w[1] - 8)

print("After:", w)
```

2. The gradient is given by

$$\nabla \hat{v}(s, \boldsymbol{w}) = \begin{bmatrix} \frac{\partial \hat{v}}{\partial w_1} \\ \vdots \\ \frac{\partial \hat{v}}{\partial w_d} \end{bmatrix}.$$

Furthermore

$$\frac{\partial \hat{v}}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_{i=1}^{d} w_i x_i(s) = x_i(s),$$

since

$$\frac{\partial w_j}{\partial w_i} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}.$$

Hence

$$\nabla \hat{v}(s, \boldsymbol{w}) = \begin{bmatrix} \frac{\partial \hat{v}}{\partial w_1} \\ \vdots \\ \frac{\partial \hat{v}}{\partial w_d} \end{bmatrix} = \begin{bmatrix} x_1(s) \\ \vdots \\ x_n ds) \end{bmatrix} = \boldsymbol{x}(s).$$

3.

(a) Since we do not have any discounting ($\gamma = 1$ we get

$$\begin{aligned} G_3 &= -0.72 \\ G_2 &= -1.33 + G_3 = -2.05 \\ G_1 &= -2.24 + G_2 = -4.29 \\ G_0 &= -2.24 + G_1 = -6.53 \end{aligned}$$

No state in the trajectory is visited more then once, hence there is no difference between every-visit and first-visit MC. This is often the case in continuous state-spaces, that you never visit exactly the same state twice (but you may visit two states that are very close to each other).

(b) From the lectures and/or textbook we know that the update is given by

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha[G_0 - \hat{v}(S_0, \boldsymbol{w})]\boldsymbol{x}(S_0)$$

since $\nabla \hat{v}(s, \boldsymbol{w}) = \boldsymbol{x}(s)$. We first note that

$$\hat{v}(S_0, \boldsymbol{w}) = \begin{bmatrix} 0 & 0 & 0 & -0.2 & -0.2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \\ 5 \\ 25 \\ 25 \\ \pi \end{bmatrix} = -10$$

Hence, the update is

$$\boldsymbol{w} \leftarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.2 \\ -0.2 \\ 0 \end{bmatrix} + \underbrace{\alpha(-6.53 + 10)}_{0.0347} \begin{bmatrix} 1 \\ 5 \\ 5 \\ 25 \\ 25 \\ \pi \end{bmatrix} = \begin{bmatrix} 0.0347 \\ 0.1735 \\ 0.1735 \\ 0.6675 \\ 0.6675 \\ 0.0347\pi \end{bmatrix}$$

(c) In this case the first update is instead given by

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha[R_1 + \hat{v}(S_1, \boldsymbol{w}) - \hat{v}(S_0, \boldsymbol{w})]\boldsymbol{x}(S_0).$$

As in (b) $\hat{v}(S_0, \boldsymbol{w}) = -10$, and

$$\hat{v}(S_1, \boldsymbol{w}) = \begin{bmatrix} 0 & 0 & 0 & -0.2 & -0.2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 5 \\ 16 \\ 25 \\ \pi \end{bmatrix} = -0.2 \times 16 - 0.2 \times 25 = -8.2.$$

Hence, the update is

$$\boldsymbol{w} \leftarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.2 \\ -0.2 \\ 0 \end{bmatrix} + \underbrace{\alpha(-2.24 - 8.2 + 10)}_{-0.0044} \begin{bmatrix} 1 \\ 5 \\ 5 \\ 25 \\ 25 \\ \pi \end{bmatrix} = \begin{bmatrix} -0.0044 \\ -0.022 \\ -0.022 \\ -0.31 \\ -0.31 \\ -0.0044\pi \end{bmatrix}.$$

4. Consider the case when $S_t = i$. Note that in $\boldsymbol{x}(S_t)$ only feature i is active (non-zero). That is, $\boldsymbol{x}(S_t)$ is equal to zero on every row except row i where it is equal to 1. This means that in the update of $\boldsymbol{w}$ only element w_i will change, and the change is given by

$$w_i \leftarrow w_i + \alpha[G_t - \hat{v}(S_t, \boldsymbol{w})]$$

Furthermore

$$\hat{v}(S_t, \boldsymbol{w}) = \sum_{j=1}^{n} w_j x_j(S_t) = w_i$$

since $x_j(S_t) = x_j(i) = 0$ if $j \neq i$. Hence the update is

$$w_i \leftarrow w_i + \alpha[G_t - w_i],$$

and for $j \neq i$ there will be no change in w_j. Now if we just let $V(i) = w_i$ then we see that the update is

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

which is exactly the update in tabular MC (with fixed step length).

It can be seen in general that the tabular methods are equivalent to using linear function approximation with the features described in this problem.

5.

(a) We use the function approximation

$$\hat{v}(s, \mathbf{w}) = w_1 s^2 + w_2 = \underbrace{\begin{bmatrix} w_1 & w_2 \end{bmatrix}}_{\mathbf{w}^\top} \underbrace{\begin{bmatrix} s^2 \\ 1 \end{bmatrix}}_{\mathbf{x}(s)}$$

The update is then given by

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha[R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})]\mathbf{x}(s)$$

Step $t = 0$ At $t = 0$ we have $\mathbf{w} = \mathbf{0}$, so

$$\hat{v}(S_t, \mathbf{w}) = \hat{v}(S_{t+1}, \mathbf{w}) = 0$$

Furthermore,

$$\mathbf{x}(s) = \begin{bmatrix} 4 \\ 1 \end{bmatrix}.$$

Hence, the update gives

$$\mathbf{w} \leftarrow \mathbf{0} - 13\alpha \begin{bmatrix} 4 \\ 1 \end{bmatrix} = -1.3 \begin{bmatrix} 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -5.20 \\ -1.30 \end{bmatrix}.$$

Step $t = 1$ Now

$$\hat{v}(S_t, \mathbf{w}) = -5.20 \times 1.09^2 - 1.30 = -7.48$$
$$\hat{v}(S_{t+1}, \mathbf{w}) = -5.20 \times 0.44^2 - 1.30 = -2.31$$
$$\mathbf{x}(s) = \begin{bmatrix} 1.09^2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.19 \\ 1 \end{bmatrix}$$

The update is thus

$$\mathbf{w} \leftarrow \begin{bmatrix} -5.20 \\ -1.30 \end{bmatrix} + 0.1[-3.88 + 0.8 \times (-2.31) - (-7.48)] \begin{bmatrix} 1.19 \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} -5.20 \\ -1.30 \end{bmatrix} + 0.18 \begin{bmatrix} 1.19 \\ 1 \end{bmatrix} = \begin{bmatrix} -4.99 \\ -1.12 \end{bmatrix}$$

(b) In this case the environment is deterministic, so the Bellman equation is simply given by

$$v_\pi(S_t) = R_{t+1} + \gamma v_\pi(S_{t+1}). \tag{5.1}$$

That is, the immediate reward plus the discounted future reward. Since we follow the policy $A_t = -1.5S_t$ we get the reward

$$R_{t+1} = -S_t^2 - (-1.5S_t)^2 = -3.25S_t^2.$$

Furthermore,

$$S_{t+1} = 2S_t + A_t = 2S_t - 1.5S_t = 0.5S_t,$$

so according to the hint

$$v_\pi(S_{t+1}) = c_1 S_{t+1}^2 = 0.25 c_1 S_t^2.$$

Using this in (5.1) we get

$$c_1 S_t^2 = -3.25S_t^2 + 0.25\gamma c_1 S_t^2 \iff (1 - 0.25\gamma)c_1 S_t^2 = -3.25S_t^2.$$

For $S_t = 0$ this always holds, and for $S_t \neq 0$ we can divide both sides with $(1 - 0.25\gamma)S_t^2$ to get

$$c_1 = -\frac{3.25}{1 - 0.25\gamma} = -4.0625.$$

Hence, the value function for this policy is given by

$$v_\pi(s) = -4.0625s.$$

(c) Since the environment is deterministic, we can write the action-value function as

$$q_\pi(S_t, A_t) = R_{t+1} + \gamma v_\pi(S_{t+1}). \tag{5.2}$$

Using $S_{t+1} = 2S_t + A_t$, we get

$$v_\pi(S_{t+1}) = c_1(2S_t + A_t)^2 = c_1(4S_t^2 + 4S_tA_t + A_t^2).$$

Inserting this into (5.2) we get

$$\begin{aligned} q_\pi(s,a) &= -s^2 - a^2 + \gamma c_1(4s^2 + 4sa + a^2) \\ &= (\gamma c_1 - 1)a^2 + 4\gamma c_1 sa + (4\gamma c_1 - 1)s^2. \end{aligned}$$

In order to do policy improvement we need to compute

$$\pi'(s) = \arg\max_a q_\pi(s,a).$$

We can find the maximum by setting the derivative with respect to a to zero, i.e.

$$\frac{dq}{da} = 2(\gamma c_1 - 1)a + 4\gamma c_1 s = 0 \iff a = -\frac{2\gamma c_1}{\gamma c_1 - 1}s = -1.5294s.$$

Hence, the improved policy is given by

$$\pi'(s) = -1.5294s.$$

(d) In the stochastic case we have to take the expected value in the Bellman equation. Then we get

$$v_\pi(s) = \mathop{\mathbb{E}}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s] \tag{5.3}$$

The policy is still $\pi(s) = -1.5s$. So

$$\mathop{\mathbb{E}}_\pi[R_{t+1}|S_t = s] = -s^2 - (-1.5s)^2 = -3.25s^2.$$

Furthermore

$$\begin{aligned} \mathop{\mathbb{E}}_\pi[v_\pi(S_{t+1})|S_t = s] &= \mathop{\mathbb{E}}_\pi[v_\pi(2S_t - 1.5S_t + E_t)|S_t = s] \\ &= c_1 \mathop{\mathbb{E}}_\pi[(0.5S_t + E_t)^2|S_t = s] + c_2 \\ &= c_1 \mathop{\mathbb{E}}_\pi[0.25S_t^2 + S_tE_t + E_t^2|S_t = s] + c_2 \\ &= 0.25c_1s^2 + c_1\sigma^2 + c_2, \end{aligned}$$

where we used $v_\pi(s) = c_1s^2 + c_2$ in the second equality, and $\mathbb{E}[E_t] = 0$ and $\mathbb{E}[E_t^2] = \sigma^2$ in the last equality.
Inserting this into (5.3) we can rewrite the Bellman equation as

$$c_1s^2 + c_2 = -3.25s^2 + 0.25c_1\gamma s^2 + \gamma(c_1\sigma^2 + c_2).$$

This must hold for all s. Rewriting the expression we get

$$(3.25 + c_1 - 0.25c_1\gamma)s^2 + (1-\gamma)c_2 - c_1\gamma\sigma^2 = 0.$$

If this is going to be true for all s we need to have

$$\begin{cases} 3.25 + c_1 - 0.25c_1\gamma = 0 \\ (1-\gamma)c_2 - c_1\gamma\sigma^2 = 0 \end{cases} \iff \begin{cases} c_1 = -\frac{3.25}{1-0.25\gamma} = -4.0625 \\ c_2 = \frac{\gamma}{1-\gamma}c_1\sigma^2 = -1.6250 \end{cases}$$

So the value function is

$$v_\pi(s) = -4.0625s^2 - 1.6250.$$