Exercise Set 4 - Reinforcement Learning
Chapter 7,8 - Control with approximation and policy gradients.

Instructions

This is the fourth exercise booklet for Reinforcement Learning. It covers both ungraded exercises to practice at home or during the tutorial sessions as well as graded homework exercises and graded coding assignments. The graded assignments are clearly marked.

- Make sure you deliver answers in a clear and structured format. LaTeXhas our preference. Messy handwritten answers will not be graded.
- Pre-pend the name of your TA to the file name you hand in and remember to put your name and student ID on the submission;
- The deadline for this first assignment is **October 4th 2023 at 19:00** and will cover the material of chapter 7-8. All questions marked 'Homework' in this booklet need to be handed in on Canvas. The coding assignments need to be handed in separately through the codegra.de platform integrated on canvas.

Contents

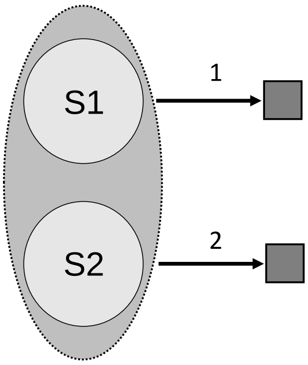

Figure 1: MDP exercise 7.3

Chapter 7: Off-policy and control with approximation

7.1 Geometry of linear value-function approximation (Theory)

1. Which error function is minimized by gradient Monte Carlo?

2. The Bellman error is zero only when the value error is zero (recall the Bellman equations). Why then does minimizing a TD objective (such as the mean squared (projected) Bellman error) not in general result in minimal mean squared value error ($\overline{VE}$) in the function approximation setting?

3. Is applying (full) gradient descent on the TD error a good approach to approximate the value function? Motivate your answer.

7.2 Deep Q Networks

1. The DQN paper [1] relies, amongst others, on the use of an earlier idea called *experience replay* [2]. What does this trick do that is important for the algorithm?

2. An other important trick is the use of a separate target network that is frozen for periods of time. What does this trick do that is important for the algorithm?

7.3 *Exam Question: Function approximation

1. Consider two types of function approximation for scalar s:

   (a) Using "Gaussian" radial basis features $\left(\phi_j(s) = exp(-\frac{(s-\mu_j)^2}{2\times width^2})\right)$.

   (b) Using polynomial features $\left(\phi_j(s) = s^j\right)$.

   Name one advantage of a) compared to b), and one advantage of b) compared to a). Assume the same number of features is used in both cases.

2. With linear function approximation, does gradient Monte Carlo (gradient MC) **always, sometimes, or never** converge to the same solution as semi-gradient TD(0)? Explain your answer.

3. Consider the simple MDP shown in Figure 1. States S1 and S2 are indistinguishable (have the same features). Only a single action can be applied, that always ends the episode. The reward obtained is 1 or 2, respectively. Episodes start in S1 or S2 with equal probability.

   (a) For the shown MDP, what is the minimal mean squared Bellman error? Why?

   (b) For the shown MDP, what is the minimal mean squared projected Bellman error? Why?

7.4 Homework: Geometry of linear value-function approximation (Application)

Consider the two-state MDP given in Figure 2. It consists of two states with one action, that transition into one another with reward 0. The features for both states are $\phi = 2$ for state s_0 and $\phi = 1$ for state s_1. We will now predict the value of the states using $v_w = w \cdot \phi$.

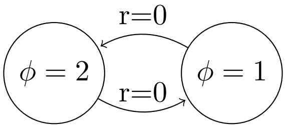

Figure 2: Two-state MDP

1. We can write the Bellman error vector as

$$\bar{\delta}_w = B^\pi v_w - v_w, \tag{1}$$

   where B^π is the Bellman operator. What is the Bellman error vector after initialization with $w = 1$ and using $\gamma = 1$?

2. What is the Mean Squared Bellman Error?

3. What w results in the value functions that is closest (in least-squares sense) to the target values $B^\pi v_w$? Use the target values you found in 1..

4. Plot v_w, $B^\pi v_w$ and $\Pi B^\pi v_w$. Explain what is happening. *Hint: Refer to Figure 11.3 in the book).*

7.5 Homework: Coding Assignment - Deep Q Networks

1. Download the notebook *RLLab4_DQN.zip* from canvas assignments and follow the instructions.

2. In the CartPole problem (in the notebook) our state is the current position of the cart, the current velocity of the cart, the current (angular) position of the pole and the (angular) speed of the pole. As these are continuous variables, we have an infinite number of states (ignoring the fact that a digital computer can only represent finitely many states in finite memory). Can you think of a way in which we can still use a tabular approach? Can you think of an example problem where this would not work? Explain why would this work for CartPole and not for the example you mentioned.

Chapter 8: Basic policy search: REINFORCE; approximations

8.1 REINFORCE / G(PO)MDP (Application)

We consider a simple example in a MDP setting with discount factor $\gamma = 1$, where we have two states (s_0 and a terminal state T) and two actions $\{1, 2\}$. The agent followed the policy $\pi(a|s, \boldsymbol{\theta})$ to take actions $a_0 = 1$ and $a_1 = 2$; then the episode $(s_0, a_0, -1, s_0, a_1, 2, T)$ was observed. We use the learning rate $\alpha = 1$ and initialize $\boldsymbol{\theta} = [1, 1]^\top$ in each of the following questions.

1. Assume that action probabilities are distributed following a categorical distribution with probabilities proportional to exponentiated weights:

$$\pi(a|s, \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\phi}(s,a)^\top \boldsymbol{\theta}}}{\sum_{b \in \{a_0, a_1\}} e^{\boldsymbol{\phi}(s,b)^\top \boldsymbol{\theta}}}, \tag{2}$$

$$\nabla_\theta \log \pi(a|s, \boldsymbol{\theta}) = \boldsymbol{\phi}(s, a) - \sum_{b \in \{1,2\}} \pi(b|s, \boldsymbol{\theta}) \cdot \boldsymbol{\phi}(s, b). \tag{3}$$

   And for state s_0 and actions, the feature vectors are $\boldsymbol{\phi}(s_0, 1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\phi}(s_0, 2) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Please compute the update of $\boldsymbol{\theta}$ according to the REINFORCE / G(PO)MDP update rule.

2. Assume that the policy follows Gaussian distribution,

$$a \sim \mathcal{N}(\boldsymbol{\phi}(s)^\top \boldsymbol{\theta}, 1), \tag{4}$$

$$\nabla_\theta \log \pi(a|s, \boldsymbol{\theta}) = (a - \boldsymbol{\phi}(s)^\top \boldsymbol{\theta}) \cdot \boldsymbol{\phi}(s). \tag{5}$$

   Given $\boldsymbol{\phi}(s_0) = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$, please compute the update of $\boldsymbol{\theta}$ according to the REINFORCE / G(PO)PMDP update rule.

3. According to the updated $\boldsymbol{\theta}$ computed in above two questions, which action do you think will be more likely to be taken in the next episode starting from s_0? Why do you think that action should be taken?

8.2 REINFORCE / G(PO)MDP (Theory)

In this question, we will show that

$$\nabla_\theta J = \mathbb{E}_\tau \left[G(\tau) \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \right] = \mathbb{E}_\tau \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \sum_{t'=t+1:T} r'_t \right],$$

as claimed in the lecture (we focus on the non-discounted case). That is, that the gradient of the log probability of taking an action needs to be multiplied only with the rewards obtained after that action was executed.

1. Show that:

$$\mathbb{E}_\tau\left[G(\tau)\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\right]$$
$$= \mathbb{E}_\tau\left[\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\sum_{t'=t+1:T} r'_t\right] + \underbrace{\sum_{t=1}^{T}\mathbb{E}_{\tau_{1:t-1},r_t}\mathbb{E}_{\mathbf{s}_t,\mathbf{a}_t|\tau_{1:t-1},r_t}\left[\nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\sum_{t'=1:t} r'_t\right]}_{\text{term 2}}. \tag{6}$$

   Hint: use the definition of G, as well as that $\mathbb{E}_{X,Y,Z}f(X,Y) = \mathbb{E}_X\mathbb{E}_{Y|X}f(X,Y)$ for (sets of) random variables X, Y, and Z.

2. Show that the second term of (6) is equal to 0.

   Hint: The main step is similar to how we showed the baseline does not introduce a bias in the lecture slides.

8.3 Baseline and gradient variance

In the lecture, we have seen that introducing a constant baseline b for the trajectory reward $G(\tau)$ does not introduce a bias to our policy gradient.

$$\nabla J = \mathbb{E}_\tau\Big[\Big(G(\tau) - b\Big)\nabla \log p(\tau)\Big] \tag{7}$$

We now want to consider the variance when introducing a baseline.

1. Derive the optimal constant baseline that minimizes the variance of the policy gradient. Interpret your result.
   Hint: First use the definition of variance to write out the variance of the gradient estimate. What should the derivative of this function w.r.t. b look like at optimality? Keep in mind the likelihood-ratio trick (Deisenroth et al, p.28).

2. Consider the simple example in a bandit setting (i.e. no states):

$$r = a + 2 \tag{8}$$
$$a \sim \mathcal{N}(\theta, 1) \tag{9}$$
$$\nabla_\theta \log \pi(a) = a - \theta \tag{10}$$

   Can you argue what should be the optimal constant baseline in this case?
   Hint: Use your result from 1.

3. Now consider a baseline that is not constant, but dependent on the state $b(s_t)$. We want to establish that in this case, the policy gradient remains unbiased. Show that

$$\mathbb{E}_\tau\left[\sum_{t=1}^{T}\nabla_\theta \log \pi(a_t|s_t)b(s_t)\right] = 0. \tag{11}$$

   Hint: You can use the linearity of expectation or the law of iterated expectation to "decouple" the full trajectory τ in two parts, $s_1, a_1, \dots, s_t$ and $a_t, r_{t+1}, s_{t+1}, \dots$.

8.4 *Exam Question: Policy Gradient Methods

You decide to build a simple e-mail answering app, that dependent on an email will decide to choose an action out of of 6 standard answers. Each answer will receive a rating between 0 and 5 stars. The action is chosen using a feedforward neural network that takes as input and ends with a softmax layer with 6 nodes, one for each answer. You want to train the network using reinforcement learning. Choosing an action and receiving the reward directly ends the episode. The next email to be answered is chosen independently from an initial state distribution.

1. Having chosen an answer to an email, resulting in a rating, you want to update the network weights using the REINFORCE algorithm. Give the REINFORCE update for this setting, and explain how you would implement this update. You can assume the use of an automatic differentiation framework that can give you the gradient of the network outputs with respect to the network weights.
2. Some e-mails can easily be identified as coming from dissatisfied customers, and these customers will give you a low rating no matter what. You decide to use a state-dependent baseline. What would be an advantage of this?
3. How would you train an additional neural network such that its output can be used as a state-dependent baseline?

8.5 Homework: REINFORCE

Consider an undiscounted Markov Decision Process (MDP) with two states A and B, each with two possible actions 1 and 2, and a terminal state T with $V(T) = 0$. The transition and reward functions are unknown, but you have observed the following two episodes using a policy $\pi(a|s,\theta)$. The form of π is not specified, however the parameters θ are split into θ_a and θ_b where the action in A only depends on θ_a and the action taken in state B only depends on θ_b ($\pi(a|A,\theta_a)$ and $\pi(a|B,\theta_b)$).

- $A \xrightarrow[r_1=200]{a_1=1} A \xrightarrow[r_2=-20]{a_2=2} B \xrightarrow[r_3=-2]{a_3=2} B \xrightarrow[r_4=-3]{a_4=2} T$
- $A \xrightarrow[r_1=-100]{a_1=1} A \xrightarrow[r_2=20]{a_2=2} B \xrightarrow[r_3=10]{a_3=1} B \xrightarrow[r_4=10]{a_4=1} T$

where the the arrow (→) indicates a transition and a_t and r_t take the values of the observed actions and rewards respectively.

1. For each episode, provide the expression for the update to parameters θ_a or θ_b given by the following algorithms:
   (a) Classical REINFORCE
   (b) REINFORCE/G(PO)MDP
2. For each update, explain what impact you would expect the update to have on probability of taking actions in state B.
3. Which algorithm do you think leads to a better update? Provide an intuitive explanation as to why you think this is the case.

4. Consider policy gradient as approximated from a single episode using classical REINFORCE:

$$\nabla_\theta J(\theta) \approx \left(\sum_{k=0}^{T} R(S_k, A_k)\right) \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(A_t \mid S_t), \tag{12}$$

   and as approximated using REINFORCE/G(PO)MDP:

$$\nabla_\theta J(\theta) \approx \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(A_t \mid S_t) \sum_{k=t}^{T} R(S_k, A_k). \tag{13}$$

   Which of the two forms has less variance around the true value of the gradient? Please provide an explanation.

5. We can also combine policy and value based methods, suggest a variation of REINFORCE/G(PO)MDP which leverages a value function, and discuss the value function's role in this algorithm.

Chapter 8: References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[2] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.