Reinforcement Learning — Homework 1

Chapters 1-8 Sutton & Barto 2nd edition

Deadline: (19th of November 2018)

1 Instructions

This is the first assignment for Reinforcement Learning. This assignment covers a refresher of earlier material and some RL basics. Please note the following:

- You are expected to hand in your solutions in LaTeX;
- Pre-pend the name of your TA to the file name you hand in and remember to put your name on the submission;
- The deadline for this first assignment is (19th of November 2018).

2 Introduction

1. Explain what is meant by the 'curse of dimensionality'.
2. Suppose you are trying to design a predator agent that can learn to catch a randomly moving prey on a 5×5 toroidal grid. You have been given the (x, y)-coordinates of the predator and the (x, y)-coordinates of the prey to use as the state.
   (a) How many possible states are there in this naive approach?
   (b) There is a way of reducing the state space considerably a priori. Write down how you would adapt the given state representation to reduce the size of the state space. Note that you only care about a representation that you can use to solve the problem.
   (c) How many possible states are there now?
   (d) What is the advantage of doing this?
   (e) Consider the Tic-Tac-Toe example in Chapter 1.5 of the book. Here, too, we can exploit certain properties of the problem to reduce the size of the state space. Give an example of a property you can exploit.
3. Suppose you want to implement a Reinforcement Learning agent that learns to play Tic-Tac-Toe, as outlined in Chapter 1 of the book.
   (a) Which agent do you think would learn a better policy in the end: a greedy agent that always chooses the action it currently believes is best, or a non-greedy agent that sometimes tries new actions? Why?

4. Assume we start with an exploration rate of ϵ, meaning that whenever the agent chooses an action, it has a probability of ϵ to pick an action at random, and a probability of $1 - \epsilon$ to pick the greedy action. If we assume the environment has been sufficiently explored, we may want to reduce the amount of exploration after some time.

   (a) Write down how you would do this.

   (b) Does your method work if the opponent changes strategies? Why/why not? If not, provide suggestions on a heuristic that can adapt to changes in the opponent's strategy.

3 Exploration

1. In ϵ-greedy action-selection for the case of n actions, what is the probability of selecting the greedy action?

2. Consider a 3-armed bandit problem with actions $1, 2, 3$. If we use ϵ-greedy action-selection, initialization at 0, and sample-average action-value estimates, which of the following sequence of actions are certain to be the result of exploration? $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 1$.

3. You are trying to find the optimal policy for a two-armed bandit. You try two approaches: in the pessimistic approach, you initialize all action-values at -5, and in the optimistic approach you initialize all action-values at +5. One arm gives a reward of +1, one arm gives a reward of -1. Using a greedy policy to choose actions, compute the resulting Q-values for both actions after three interactions with the environment. In case of a tie between two Q-values, break the tie at random.

4. Which initialization leads to a higher (undiscounted) return? What if you had broken the tie differently?

5. Which initialization leads to a better estimation of the Q-values?

6. Explain why one of the two initialization methods is better for exploration.

4 Markov Decision Processes

1. (a) For the first four examples outlined in Section 1.2 of the book, describe the state space, action space and reward signal.

   (b) Come up with an example of your own that you might model with an MDP. State the action space, state space and reward signal.

   (c) Come up with an example for a problem that you might have trouble solving with an MDP. Why doesn't this fit the framework?

   (d) Consider the example in exercise 3.3 of the book. Why might you choose to view the actions as handling the accelerator, brake and steering wheel? What is the disadvantage of doing this?

   (e) Why might you choose to view the actions as choosing where to drive? What is the disadvantage of doing this?

   (f) Can you think of some way to combine both approaches?

2. (a) Eq. 3.8 in the book gives the discounted return for the continuous case. Write down the formula for the discounted return in the episodic case.
   (b) Show that $\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$ if $0 \leq \gamma < 1$. (Hint: if you're stuck, have a look at the Wikipedia page on *geometric series*)
   (c) Consider exercise 3.7 in the book. Why is there no improvement in the agent?
   (d) How would adding a discount factor of $\gamma < 1$ help solve this problem?
   (e) How might changing the reward function help solve this problem?

5 Dynamic Programming

1. Write the value, $v^\pi(s)$, of a state s under policy π, in terms of π and $q^\pi(s, a)$. Write down both the stochastic and the deterministic policy case.
2. In Policy Iteration, we first evaluate the policy by computing its value function, and then update it using a Policy Improvement step. You will now change Policy Iteration as given on page 80 of the book to compute action-values. First give the new policy evaluation update in terms of $Q^\pi(s, a)$ instead of $V^\pi(s)$. Note that Policy Improvement uses deterministic policies.
3. Now change the Policy Improvement update in terms of $Q^\pi(s, a)$ instead of $V^\pi(s)$.
4. The Value Iteration update, given in the book in Eq. 4.10 can also be rewritten in terms of Q-values. Give the Q-value Iteration update.

6 Monte Carlo

1. Consider an MDP with a single state s_0 that has a certain probability of transitioning back onto itself with a reward of 0, and will otherwise terminate with a reward of 5. Your agent has interacted with the environment and has gotten the following three trajectories: $[0, 0, 5], [0, 0, 0, 0, 5], [0, 0, 0, 5]$. Use $\gamma = 0.9$.
   (a) Estimate the value of s_0 using first-visit MC.
   (b) Estimate the value of s_0 using every-visit MC.
2. What is a disadvantage of using *ordinary importance sampling* in off-policy Monte Carlo?
3. What is a disadvantage of using *weighted importance sampling* in off-policy Monte Carlo?

7 Temporal Difference Learning (Application)

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 episode using a random policy:

- $A \xrightarrow[r_3=-3]{a_3=1} B \xrightarrow[r_4=4]{a_4=1} A \xrightarrow[r_5=-4]{a_5=2} A \xrightarrow[r_6=-3]{a_6=1} B \xrightarrow[r_7=1]{a_7=2} T$

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

1. What are the state(-action) value estimates $V(s)$ (or $Q(s,a)$) after observing the sample episode when applying:

   (a) TD(0)
   (b) 3-step TD
   (c) SARSA
   (d) Q-learning

   where we initialize state values to 0 and use a learning rate $\alpha = 0.1$

2. Choose a deterministic policy that you think is better than the random policy given the data. Refer to any of the state(-action) value estimates to explain your reasoning.

3. Let π_{random} denote the random policy used so far and $\pi_{student}$ denote the new policy you proposed. Suppose you can draw new sample episodes indefinitely until convergence of the value estimates.

   (a) Discuss how do you expect the final value estimates to differ if you ran Q-Learning with π_{random} as compared to $\pi_{student}$.
   (b) What problems may arise with π_{random} or $\pi_{student}$ respectively?
   (c) Do you think using an ϵ-greedy policy as behavior policy would be beneficial? Explain why/why not?

8 Temporal Difference Learning (Theory)

1. We can use Monte Carlo to get value estimates of a state with $V_M(S) = \frac{1}{M}\sum_{n=1}^{M} G_n(S)$ where $V_M(S)$ is the value estimate of state S after M visits of the state and $G_n(S)$ the return of an episode starting from S. Show that $V_M(S)$ can be written as the update rule $V_M(S) = V_{M-1}(S) + \alpha_M[G_M(S) - V_{M-1}(S)]$ and identify the learning rate α_M.

2. Consider the TD-error
$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t). \tag{1}$$
   (a) What is $\mathbb{E}[\delta_t | S_t = s]$ if δ_t uses the true state-value function V^π
   (b) What is $\mathbb{E}[\delta_t | S_t = s, A_t = a]$ if δ_t uses the true state-value function V^π

3. The Monte-Carlo error can be written as the sum of TD-errors if the value estimates don't change (cf. equation 6.6 in the book) as
$$G_t - V(S_t) = \sum_{k=t}^{T-1} \gamma^{k-t}\delta_k. \tag{2}$$
   (a) If V changes during the episode, then this equation only holds approximately, what would the difference be between the two sides? Let V_t denote the array of state values used at time t in the TD error δ_t and in the TD update $V_{t+1}(S_t) = V_t(S_t) + \alpha\delta_t$. Redo the derivation above to determine the additional amount that must be added to the sum of TD errors in order to equal the Monte Carlo error.
   (b) Show that the n-step error as used in
$$V_{t+n}(S_t) = V_{t+n-1}(S_t) + \alpha[G_{t:t+n} - V_{t+n-1}(S_t)], \quad 0 \le t < T \tag{3}$$
   can also be written as a sum TD errors (assuming the value estimates don't change).

9 Maximization Bias

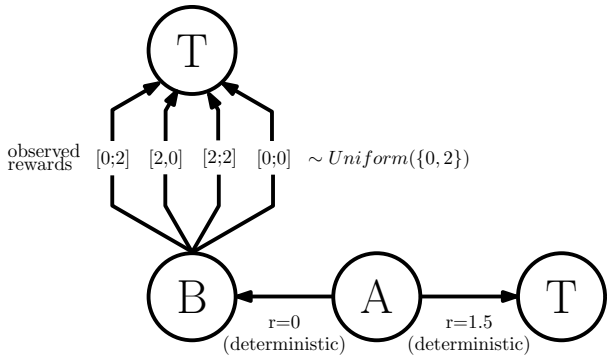

Figure 1: MDP: Maximization Bias

Consider the undiscounted MDP in Figure 1 where we have a state A with two actions, one ending in the terminal state T (with $V(T) = 0$) and always giving a reward of 1.5 and another action that transitions to state B with zero reward. From state B we can take a total of four different actions each transitioning to the terminal state and giving a reward of either 0 or 2 with equal probability. Suppose we sample all possible episodes two times and record the rewards. The observed rewards for each of the four actions from B are indicated in the Figure, e.g. the leftmost action received one time a reward of 0 and one time a reward of 2.

1. We repeatedly apply Q-learning and SARSA on the observed data until convergence. Give all final state-action values for Q-learning and SARSA respectively.

2. This problem suffers from maximization bias. Explain where this can be observed. Do both Q-learning and SARSA suffer from this bias? Why/why not?

3. To circumvent the issue of maximization bias, we can apply Double Q-learning. Use the given example to explain how Double Q-learning alleviates the problem of maximization bias.

4. What are the true state-action values that we would expect to get (after convergence) if we continued sampling episodes.

10 Model-based RL

Consider the setup from Section 7 where we observed one additional episode:

- $A \xrightarrow[r_3=-3]{a_3=1} B \xrightarrow[r_4=4]{a_4=1} A \xrightarrow[r_5=-4]{a_5=2} A \xrightarrow[r_6=-3]{a_6=1} B \xrightarrow[r_7=1]{a_7=2} T$
- $A \xrightarrow[r_1=3]{a_1=1} A \xrightarrow[r_2=1]{a_2=2} A \xrightarrow[r_3=-3]{a_3=1} B \xrightarrow[r_4=-10]{a_4=1} T$

1. Draw a diagram of the MDP that best explains the observed episodes (i.e. the model that maximizes the likelihood of the data). Show rewards and transition probabilities on your diagram. Rewards can be expressed as point estimates. Transitions can be probabilistic.

2. Consider the Dyna-Q algorithm as described in the book in Section 8.2.

(a) Which assumption does it make that is not satisfied by the MDP that generated the data?

(b) How would you modify the Dyna-Q algorithm to make it work in this setting?

(c) Does your suggested modification deal well with a changing environment? If not, how could that be addressed?

3. Suppose we have two agents that we initialize equally. The first agent uses the Dyna-Q algorithm with infinite planning steps (i.e. after observing one episode the agent uses its planning module until the Q-values converge). The second agent uses Q-learning, but after each episode it repeatedly applies its update rule on the sampled episode until the Q-values converge. Afterwards the episode data is discarded and a new episode is sampled. If we give both agent the same experience sampled from a random policy, do we expect that their Q-values differ after

   (a) ... the first episode?

   (b) ... the second episode?

   (c) Do both learning schemes converge to the optimal Q-values if we continued sampling new episodes indefinitely?

   Explain.

11 Bonus exercises

11.1 Contraction Mapping

Consider the following Theorem, from Csaba Szepesvari's *Algorithms for Reinforcement Learning*:

Theorem 1. (Banach's Fixed Point Theorem). Let V be a Banach space and $T : V \to V$ be a contraction mapping. Then T has a unique fixed point. Further, for any $v_0 \in V$, if $v_{n+1} = Tv_n$, then $v_n \to_{||\cdot||} v$, where v is the unique fixed point of T and the convergence is geometric:

$$||v_n - v|| \leq \gamma^n ||v_0 - v|| \tag{4}$$

1. (a) Consider the contraction mapping $T(x) := 1 + \frac{1}{3}x$. Find the fixed point of this mapping.

   (b) Consider the contraction mapping $(Tf)(s) := \frac{-1}{2}f(s) + g(s)$. Find the fixed point of this mapping in terms of $g(s)$.

2. Consider an MDP with finite state space $\mathcal{S}$, finite action space $\mathcal{A}$, and discount factor γ. Let V be the value function which gives the value per state, and $R(s, a)$ the expected immediate reward for action a in state s and $Pr(s'|s, a)$ the probability of transitioning to state s' after taking action a in state s. Then the *Bellman optimality operator* B^* is given by

$$(B^*V)(s) = \max_a \left[R(s, a) + \gamma \sum_{s' \in \mathcal{S}} Pr(s'|s, a)V(s') \right] \tag{5}$$

   (a) Show that

$$|\max_z f(z) - \max_z h(z)| \leq \max_z |f(z) - h(z)| \tag{6}$$

   for arbitrary $f(z)$ and $h(z)$.

Let $(X, ||\cdot||)$ be a metric space. $T : X \to X$ is said to be a contraction if there exists a constant $0 \leq c < 1$ such that

$$||T(x) - T(y)|| \leq c||x - y|| \quad \forall x \in X, \forall y \in X \tag{7}$$

In our case, the metric space is $(\mathcal{S}, ||\cdot||_\infty)$. Thus, the norm is the supremum norm $||V_1 - V_2||_\infty = \max_{s \in \mathcal{S}} |V_1(s) - V_2(s)|$, or the largest difference between state values.

(b) Show that B^* is a contraction mapping in supremum norm, i.e.

$$||(B^*V_1)(s) - (B^*V_2)(s)||_\infty \leq c||V_1(s) - V_2(s)||_\infty \tag{8}$$