EL2805 Reinforcement Learning

Exercise Session 6

December 12, 2019

Department of Automatic Control
School of Electrical Engineering
KTH Royal Institute of Technology

6 Exercises

6.1

Provide short answers to the following questions (not more than ca 5 sentences per question).

a) What is the (computational) complexity of finite horizon dynamic programming? [1 pts]

b) What function approximation class is used in DQN? [1 pts]

c) Q-learning is an *(...)*-policy algorithm? What does this mean? [1 pts]

d) Under what two conditions on the step-size α_k does a stochastic approximation algorithm converge? [1 pts]

e) Why is function approximation important in RL? [1 pts]

f) What is *experience replay* and why is it useful? [1 pts]

g) Provide two interpretations of the discount factor λ in an MDP with infinite horizon objective $\mathbb{E}\{\sum_{t=0}^{\infty} \lambda^t r(s_t, a_t)\}$. [1 pts]

h) What is the Markov property? [1 pts]

i) Will policy iteration (PI) always terminate in a finite number of iterations if $\mathcal{S}$ and $\mathcal{A}$ are finite? Why/why not? [1 pts]

j) Mention a setting in which one cannot employ Monte Carlo methods, and TD methods have to be employed. Motivate. [1 pts]

6.2

We are observing and recording the outcomes of dependent flips of a coin. The probability that any flip will have the same outcome as the previous flip is equal to p.

a) Model the sequence of coin-flips as a Markov chain. [1pts]

b) For what values of p does the Markov chain have a unique stationary distribution? [2pts]

c) For the values of p such that the stationary distribution is unique, compute it. [1pt]

Assume now that we are observing the outcomes (of the same dependent coin-flips) at a distance on a foggy day. Our observations of the experimental outcomes are imperfect. In fact, the probability that we shall properly record the outcome of any trial is equal to q and is independent of all previous or future errors. After every flip (and corresponding foggy observation), we have to guess the *actual outcome* of the coin-flip. If our bet is correct, we get 10 SEK; if our bet is incorrect, we get 0 SEK. Assume that the game is played over 10 flips, and that we want to maximize our wealth.

d) Model this as an MDP, and specify any changes to the standard framework that have to be done. *Do not try to solve the MDP.* [3pts]

Let y_k be the observation we record in the kth trial (either h or t for heads and tails, respectively).

e) Can the possible sequences of our observations be modeled using the state history of a Markov chain? If so, define the quantities of such a Markov chain. *Hint:* You might need to consider a Markov chain whose state includes the true coin flip outcome and its observation. [3pts]

6.3

A manufacturer receives an order at each time period for her product with probability p. At any period, she has the choice of processing all the unfilled orders in a batch, or to process no order at all. The cost per unfilled order at any period is $c > 0$, and the setup cost to process unfilled order is $K > 0$. Assume that the total number of orders that can remain unfilled is n (at this point, the orders have to be processed), and that there is a discount factor $\lambda < 1$. The goal is to find a processing policy that minimizes the total expected (discounted) cost.

a) Provide an MDP of this problem. [2 pts]

b) Derive the optimality conditions for your MDP. [3 pts]

c) Characterize the optimal processing policy. [5 pts]

Hint: If $V(i)$ is the value of having i unprocessed jobs; prove that $V(i)$ is a monotonic function in i using induction. You might want to recall the value iteration (VI) algorithm.

6.4

Consider a system with $\mathcal{S} = \{\text{A, B, C}\}$ and $\mathcal{A} = \{\varphi, \beta, \gamma\}$. We are currently at time-step 5734 of an episode, with current state-value function estimate:

$$Q^{(5734)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} \varphi & \beta & \gamma \end{array} \\ \begin{bmatrix} 100 & 200 & 80 \\ 120 & 50 & 140 \\ 90 & 200 & 100 \end{bmatrix} \end{array} .$$

The trajectory we are observing goes as

$$(\dots, \underbrace{A, \quad \varphi, \quad 200}_{s_{t=5734},\, a_{t=5734},\, r_{t=5734}}, B, \varphi, 150, A, \gamma, 140, C, \gamma, \dots).$$

a) Provide the updated Q-values, if we are using *Q-learning*, after these observations, assuming that the discount factor is $\lambda = 0.1$ and the learning rate is fixed to $\alpha = 0.5$. [3 pts]

b) What is the current greedy policy? [1 pts]

c) Provide the updated Q-values, if we are using *SARSA*, after these observations, assuming that the discount factor is $\lambda = 0.1$ and the learning rate is fixed to $\alpha = 0.5$. [3 pts]

d) What is the current greedy policy? [1 pts]

e) With ε-greedy action selection, Q-learning and SARSA will converge to state-action value functions corresponding to different policies. Describe the characteristics of these. Assume we want to find the optimal policy. How would you accomplish this in SARSA? [2 pts]

6.5

Consider an application where the action set is continuous $\mathcal{A} = \mathbb{R}$.

a) Give three examples of RL tasks where the action sets are continuous. [1 pts]

A common policy parametrization in this setting is to parametrize the policy's probability density function using Gaussian functions:

$$\pi_\theta(s,a) = p(a_t = a | s_t = s) = \frac{1}{\sigma(s,\theta)\sqrt{2\pi}} \exp\left\{ -\frac{(a-\mu(s,\theta))^2}{2\sigma(s,\theta)^2} \right\}.$$

b) Let $\theta = [\theta_\mu, \theta_\sigma]^T$ and take

$$\mu(s,\theta) = \theta_\mu^T x_\mu(s) \text{ and } \sigma(s,\theta) = \exp\left\{\theta_\sigma^T x_\sigma(s)\right\},$$

where $x(s)$ are feature vectors. Why is the variance parametrized differently (compared to the mean)? [1 pts]

c) Compute the eligibility vector $\nabla \log \pi_\theta(s,a)$. [4 pts]

Now consider applications where the action-set $\mathcal{A}$ is discrete (and finite).

d) Prove that

$$\mathbb{E}_{\pi_\theta}\left\{\nabla \log \pi_\theta(s_t, a_t)\right\} = 0,$$

for a *general* policy parametrization $\pi_\theta(s,a)$. [4 pts]

7 Solutions

Solution to Problem 6.1

a) $\mathcal{O}(S^2AT)$.

b) (Deep) neural networks.

c) Off-policy. It does not learn the value of the policy that is implemented.

d) $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 < \infty$.

e) In order to handle large or continuous spaces for which it is not feasible to work with tabulated quantities.

f) In Q-learning with function approximation, successive updates are strongly correlated (since they follow a particular trajectory). In order to improve the convergence rate, with experience replay, one maintains a buffer B of previous experiences (s, a, r, s') and then samples mini-batches of fixed size k from B uniformly at random.

g) *Interest rate:* The value of a unit reward decreases with time at geometric rate λ. *Random time horizon:* the decision maker has a time horizon T that is geometrically distributed.

h) A sequence of random variables x_k fullfils the Markov property if

$$\Pr\{x_{k+1}|x_k, x_{k-1}, \ldots, x_0\} = \Pr\{x_{k+1}|x_k\}.$$

i) Yes, since there is only a finite number of policies to evaluate.

j) When the episode does not have finite length.

Solution to Problem 6.2

a) The Markov chain is defined by:

- **State-space:** $\mathcal{S} = \{H, T\}$
- **Transitions:**

$$P = \begin{array}{c} \\ H \\ T \end{array} \begin{array}{c} \begin{array}{cc} H & T \end{array} \\ \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix} \end{array}.$$

b) The Markov chain is finite (2 states). A finite Markov chain has a unique stationary distribution when it is irreducible. This holds when $p < 1$. (*Note:* For it to have a limiting distribution, it has to be aperodic as well (which happens when $p > 0$.))

c) The stationary distribution is given as the left eigenvector of P corresponding to the eigenvalue 1. In math, for a row-vector π, it should hold that

$$\pi = \pi P,$$

and

$$\sum_i \pi_i = 1.$$

Solving these equations yields

$$\pi = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

d) The following MDP models the problem:

- **State-space:** $x_k \in \mathcal{S} = \{H, T\}$.
- **Actions:** $\mathcal{H}$ - heads, $\mathcal{T}$ - tails.
- **Objective:** $\mathbb{E}\{\sum_{k=0}^{T} r_k(x_k, a_k) + r_T(x_T)\}$, for $T = 10$.
- **Rewards:** Terminal: $r_T(\cdot) = 0$. Non-terminal: $r_k(x_k = H, a_k = \mathcal{H}) = r_k(x_k = T, a_k = \mathcal{T}) = 10$, all other zero.

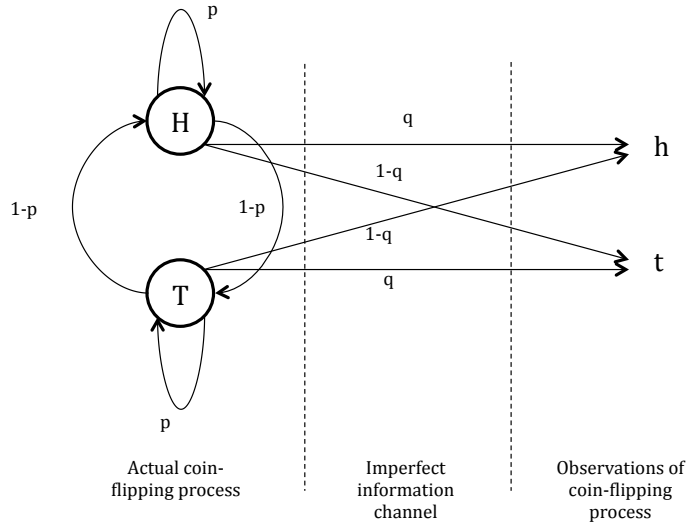

Figure 1: Schematic figure of the foggy coin-flipping situation.

- **Transitions:** Equal to P as specified above (independent of the chosen action).

However, *note that the state x_k is not directly observed*. Rather, we obtain a noisy observation y_k at each time instant (which is equal to x_k with probability q). This means that we do not know the reward we will aquire (since x_k is unknown). To solve the MDP, we would need to compute a state estimate $\hat{x}_k$ (given the observations $y_1, \ldots, y_k$) and base our decision on that estimate.

Note: This is called a *partially observed MDP* (POMDP). (Essentially LQG for linear systems, if you have taken linear control theory.)

e) We model the system as follows

- **State space:** Let the state z_k of the Markov chain be a pair $z_k = (x_k, y_k)$, where the first element x_k denotes the actual outcome of the coin flip, and the second element y_k denotes what we observe it to be.

  Moreover, let H and T denote the actual outcome of the coin flip (heads or tails, respectively), and h and t what we observe it to be. Then,

$$z_k = (x_k, y_k) \in \mathcal{S} = \{H, T\} \times \{h, t\} = \{(H, h), (H, t), (T, h), (T, t)\}.$$

- **Transitions:** Consider that the current state is $z_k = (H, h)$. To end up in, for example, $z_{k+1} = (T, h)$, we need to first transition to H from T (probability $1 - p$) and then make an incorrect observation (probability $1 - q$) – see Figure 7.

  In general, by continuing this reasoning, we obtain the transition matrix:

$$P = \begin{array}{c} \\ (H,h) \\ (H,t) \\ (T,h) \\ (T,t) \end{array} \begin{array}{c} \begin{array}{cccc} (H,h) & (H,t) & (T,h) & (T,t) \end{array} \\ \begin{pmatrix} pq & p(1-q) & (1-p)(1-q) & (1-p)q \\ pq & p(1-q) & (1-p)(1-q) & (1-p)q \\ (1-p)q & (1-p)(1-q) & p(1-q) & pq \\ (1-p)q & (1-p)(1-q) & p(1-q) & pq \end{pmatrix} \end{array}.$$

The possible sequences of our observations can be modeled via this Markov chain, by ignoring the first component of the state vector.

Solution to Problem 6.3

Part a)

We define the state as the number of unfilled orders at the beginning of each period, and hence the state space is $S = \{0, 1, \ldots, n\}$. For states $s = 1, \ldots, n-1$, we have $A_s = \{P, \overline{P}\}$, where P (resp. $\overline{P}$) corresponds to processing unfilled orders (resp. processing no order). We have $A_0 = \{\overline{P}\}$ and $A_n = \{P\}$.

The rewards are given by:

$$\begin{aligned} &r(i, P) = K, \quad r(i, \overline{P}) = ci, \quad i = 1, \ldots, n-1, \\ &r(0, \overline{P}) = 0, \quad r(n, P) = K. \end{aligned}$$

Transition probabilities are given by:

$$\begin{aligned} &p(0|i, P) = 1 - p, \quad p(1|i, P) = p, \quad i = 1, 2, \ldots, n-1, \\ &p(i|i, \overline{P}) = 1 - p, \quad p(i+1|i, \overline{P}) = p, \quad i = 1, 2, \ldots, n-1, \\ &p(0|n, P) = 1 - p, \quad p(1|n, P) = p, \\ &p(0|0, \overline{P}) = 1 - p, \quad p(1|0, \overline{P}) = p. \end{aligned}$$

Part b)

The Bellman's equation takes the form:

$$\begin{aligned} V(i) &= \min\{K + \lambda(1-p)V(0) + \lambda pV(1), ci + \lambda(1-p)V(i) + \lambda pV(i+1)\}, \quad i = 0, 1, \ldots, n-1, \\ V(n) &= K + \lambda(1-p)V(0) + \lambda pV(1), \quad i = n. \end{aligned}$$

Part c)

We show below through induction that the optimal cost $V(i)$ is monotonically nondecreasing in i. Hence, if processing a batch of m orders is optimal, that is,

$$K + \lambda(1-p)V(0) + \lambda V(1) \leq cm + \lambda(1-p)V(m) + \lambda pV(m+1),$$

then processing a batch of $m+1$ orders is also optimal. Therefore, the optimal policy is a threshold policy, which decides to process the orders if their number exceeds some threshold integer $m^\star$ which satisfies:

$$K + \lambda(1-p)V(0) + \lambda V(1) \leq cm^\star + \lambda(1-p)V(m^\star) + \lambda pV(m^\star + 1).$$

Suppose that $V_k(i+1) \geq V_k(i)$ for all i. We will show that $V_{k+1}(i+1) \geq V_{k+1}(i)$ for all i. Consider first the case $i + 1 < n$. Then by induction hypothesis, we have that

$$c(i+1) + \lambda(1-p)V_k(i+1) + \lambda pV_k(i+2) \geq ci + \lambda(1-p)V_k(i) + \lambda pV_k(i+1).$$

Define for any scalar γ,

$$F_k(\gamma) = \min\{K + \lambda(1-p)V_k(0) + \lambda pV_k(1), \gamma\}.$$

Since $F_k(\gamma)$ is monotonically increasing in γ, from the above equations we have that

$$\begin{aligned} V_{k+1}(i+1) &= F_k\Big(c(i+1) + \lambda(1-p)V_k(i+1) + \lambda pV_k(i+2)\Big) \\ &\geq F_k\Big(ci + \lambda(1-p)V_k(i) + \lambda pV_k(i+1)\Big) \\ &= V_{k+1}(i). \end{aligned}$$

Finally consider the case $i + 1 = n$. It follows that

$$\begin{aligned} V_{k+1}(n) &= K + \lambda(1-p)V_k(0) + \lambda pV_k(1) \\ &\geq F_k\Big(ci + \lambda(1-p)V_k(i) + \lambda pV_k(i+1)\Big) \\ &= V_{k+1}(n-1) \end{aligned}$$

and hence the induction is complete.

Solution to Problem 6.4

Part a)

$$Q^{(5737)} = \begin{matrix} \\ A \\ B \\ C \end{matrix} \begin{matrix} \begin{matrix} \varphi & \beta & \gamma \end{matrix} \\ \begin{bmatrix} 157 & 200 & 120 \\ 145 & 50 & 140 \\ 90 & 200 & 100 \end{bmatrix} \end{matrix} .$$

Part b)

$$\pi^{(5737)}(s) = \begin{cases} \beta \text{ if } s = A, \\ \varphi \text{ if } s = B, \\ \beta \text{ if } s = C. \end{cases}$$

Part c)

$$Q^{(5737)} = \begin{matrix} \\ A \\ B \\ C \end{matrix} \begin{matrix} \begin{matrix} \varphi & \beta & \gamma \end{matrix} \\ \begin{bmatrix} 156 & 200 & 115 \\ 139 & 50 & 140 \\ 90 & 200 & 100 \end{bmatrix} \end{matrix} .$$

Part d)

$$\pi^{(5737)}(s) = \begin{cases} \beta \text{ if } s = A, \\ \gamma \text{ if } s = B, \\ \beta \text{ if } s = C. \end{cases}$$

Part e)

Q-learning will converge to the optimal policy. SARSA will converge to an optimal policy that takes into account that we explore with probability ε. By letting $\varepsilon \to 0$, we can make SARSA tend to the optimal policy.

Solution to Problem 6.5

Part b)
The variance has to be non-negative. This is enforced by the exponential function.
Part c)
We have that

$$\nabla \log \pi_\theta(s,a) = \begin{bmatrix} \nabla_{\theta_\mu} \\ \nabla_{\theta_\sigma} \end{bmatrix} \log \pi_\theta(s,a) = \begin{bmatrix} \nabla_{\theta_\mu} \{\log \pi_\theta(s,a)\} \\ \nabla_{\theta_\sigma} \{\log \pi_\theta(s,a)\} \end{bmatrix}$$

with

$$\nabla_{\theta_\mu} \{\log \pi_\theta(s,a)\} = \frac{1}{\sigma(s,\theta)^2} \left(a - \mu(s,\theta)\right) x_\mu(s)$$

and

$$\nabla_{\theta_\sigma} \{\log \pi_\theta(s,a)\} = \left(\frac{(a - \mu(s,\theta))^2}{\sigma(s,\theta)^2} - 1 \right) x_\sigma(s).$$

Part d)

$$
\begin{aligned}
\mathbb{E}_{\pi_\theta}\left\{\nabla \log \pi_\theta(s_t, a_t)\right\} &= \sum_{s,a} \Pr\{s_t = s, a_t = a\} \nabla \log \pi_\theta(s, a) \\
&= \sum_{s,a} \Pr\{a_t = a | s_t = s\} \Pr\{s_t = s\} \nabla \log \pi_\theta(s, a) \\
&= \sum_{s} \Pr\{s_t = s\} \sum_{a} \Pr\{a_t = a | s_t = s\} \nabla \log \pi_\theta(s, a) \\
&= \sum_{s} \Pr\{s_t = s\} \sum_{a} \pi_\theta(s, a) \nabla \log \pi_\theta(s, a) \\
&= \sum_{s} \Pr\{s_t = s\} \sum_{a} \pi_\theta(s, a) \frac{\nabla \pi_\theta(s, a)}{\pi_\theta(s, a)} \\
&= \sum_{s} \Pr\{s_t = s\} \sum_{a} \nabla \pi_\theta(s, a) \\
&= \sum_{s} \Pr\{s_t = s\} \nabla \sum_{a} \pi_\theta(s, a) \\
&= \sum_{s} \Pr\{s_t = s\} \nabla 1 \\
&= \sum_{s} \Pr\{s_t = s\} \cdot 0 \\
&= 0.
\end{aligned}
$$