EL2805 Reinforcement Learning

Exam – April 2022

Division of Decision and Control Systems
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Re-exam (omtentamen), April 22, 2022, kl 14.00 - 19.00

Aids. Slides of the lectures (**not exercises**), lecture notes (summary.pdf), mathematical tables.

Observe. Do not treat more than one problem on each page. Each step in your solutions must be motivated. Write a clear answer to each question. Write name and personal number on each page. Please only use one side of each sheet. Mark the total number of pages on the cover.

The exam consists in 5 problems. The distribution of points among these problems is indicated below.

Grading.
Grade A: ≥ 43 Grade B: ≥ 38
Grade C: ≥ 33 Grade D: ≥ 28
Grade E: ≥ 23 Grade Fx: ≥ 21

Responsible. Alexandre Proutiere 087906351

Results. Posted no later than May 5, 2022

Good luck!

Problem 1 - Quiz

(a) Why can't we use the Policy Gradient approach for off-policy learning? [1 pt]

(b) Consider the following problem: in each round you choose a scalar θ and you observe a random variable $f(\theta)$ such that $\mathbb{E}[f(\theta)] = g(\theta)$. Which technique would you use to solve in θ the equation $g(\theta) - \alpha = 0$? (for some given α) [1 pt]

(c) What is the complexity (number of operations) of solving the Bellman's equations in a finite time-horizon MDP with S states, A actions, and time-horizon T? [1 pt]

(d) Is the SARSA algorithm based on the stochastic approximation algorithm or the stochastic gradient algorithm? [1 pt]

(e) Is the Q-learning algorithm with function approximation based on the stochastic approximation algorithm or the stochastic gradient algorithm? [1 pt]

(f) In SARSA, we propose to use ϵ-greedy policy with a value of ϵ decreasing over time. More precisely, we select $\epsilon_t = 1/t^2$. The algorithm does not seem to converge. Can you explain why? [1 pt]

(g) In actor-critic algorithms, how many parameters do we need to update? What do they correspond to? [1 pt]

(h) Suppose we take the step-size $\alpha_t = 1/\log(t)$ in the Q-learning algorithm. Are the iterates guaranteed to converge almost surely to the true Q-function? [1 pt]

(i) Let $X_1, X_2, \ldots$ be an homogenous Markov chain with finite state space. Is the reverse process starting at time N also a Markov chain? (The reverse process is $(X_N, X_{N-1}, \ldots, X_1)$) [2 pts]

Problem 2

Financial Options. Consider a financial market with a single stock which trades at a certain (market) price. Each day, the stock, with probability p either gains a fraction $\rho_+ \in (0,1)$ of its current value or with probability $1-p$ loses a fraction $\rho_- \in (0,1)$ of its current value. Suppose the initial value of the stock is 1 (dollar).

(a) Call option. At the beginning of a week, you are given one call option. At the beginning of each day of this week, you may decide to apply this call option, but once the option is used, you cannot apply it anymore. When exercised, the call option gives you the right to buy a stock at price $K \in \mathbb{R}_{\geq 0}$ instead of the market price. We assume that after using the call option, you immediately resell the stock at the market price. You wish to come up with a policy on how to use your call option to maximize the expected gain in the week. Model this problem as an MDP (do not solve the MDP). [4 pts]

(b) Put option. Now assume that at the beginning of the week, you are given one put option. At the beginning of each day of this week, you may decide to apply this put option, but once the option is used, you cannot apply it anymore. When exercised, the put option gives you the right to buy a stock at the market price and to immediately resell it at price $K \in \mathbb{R}_{\geq 0}$. You wish to come up with a policy on how to use your put option to maximize the expected gain in the week. Model this problem as an MDP (do not solve the MDP). [1 pt]

Trading in the Stock Market. You are now entrusted with a portfolio of your own – provided that you can model it of course. Consider again a financial market with a single stock which trades at a certain (market) price. Each day, the stock, with probability p either gains a fraction $\rho_+ \in (0,1)$ of its current value or with probability $1-p$ loses a fraction $\rho_- \in (0,1)$ of its current value. You are a given a starting capital of $1M$ USD and 100 call options at price level $K \in \mathbb{R}_+$. When using a call option, you do not need to resell the acquired stock immediately (as we assumed in (a)). You may trade freely in the stock market: every day you can buy or sell stocks at the market price (depending on how much cash you have available), and to use your options (you can use several options in one day). You seek to maximize the value of your portfolio over the period of 1 year, and need to liquidate all your assets at the end of the year (you have to sell all your remaining stocks the last day).

(c) Model the problem as an MDP (do not solve the MDP). [5 pts]

Problem 3. Inflation.

You will move to Stockholm in N weeks, and you need to buy a one-room apartment before you move. The real-estate prices are constantly increasing in Stockholm, with an average increase rate $\alpha > 1$ per week. Each week, say the t-th week, you have the opportunity to buy an apartment whose price is p_t, and may decide to buy it or not (after observing its price of course). This price p_t can be written as $p_t = \alpha^t w_t$, where the random variables $w_1, w_2, \ldots, w_N$ are independent and identically distributed over a finite set $\mathcal{W}$; let $f(w) = \mathbb{P}[w_1 = w]$ for $w \in \mathcal{W}$. You are forced to buy an apartment the N-th week if you haven't bought one earlier. Your objective is to minimize the expected price you pay for the apartment.

(a) Model the problem as an MDP. [2 pts]

(b) Establish that the optimal strategy is threshold-based, i.e., at the t-th week (if you haven't bought an apartment earlier) you should buy the apartment if its price is below a threshold β_t. [2 pts]

(c) Provide a recursive expression for the thresholds (express β_t as a function of β_{t+1}). [2 pts]

(d) Imagine now that when you bought an apartment before moving, you can resell it. In the t-th week, you can sell it at a price $\alpha^t z_t$ where $z_1, z_2, \ldots, z_N$ are independent and identically distributed, with the same distribution as w_1. We assume that you can only perform one action per week (buy or sell, but not both). Note that if you decide to resell your apartment, you need to buy a new one before moving. Reformulate the problem as a new MDP. Is the optimal policy threshold-based (if you do not own an apartment in the t-th week, you buy one if the price is lower than some threshold γ_t, and if you do own an apartment in the t-th week, you resell it if the offered price is greater than some threshold ζ_t)? Can you relate γ_t and ζ_t? [4 pts]

Problem 4

Exercise 1 Consider a discounted MDP with state space $\mathcal{S} = \{\text{A, B, C}\}$ and action space $\mathcal{A} = \{u_1, u_2, u_3\}$. We plan to use the SARSA algorithm to learn to control the system. We initialize the following Q-function

$$Q^{(0)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} .$$

where $x > 0$ is an unknown value. The discount factor is λ and the learning rate is fixed to α. The observed trajectory is as follows (for these transitions, we are imposed a policy):

$$(B, u_1, 1); (B, u_2, 2); (C, u_1, 3); (A, u_3, -1); (C, u_2, 10); (A, u_2, -5); (A, u_1, \dots)$$

where each triplet represents the state, the selected action, and the corresponding reward.

(a) Provide the updated Q-values, using the SARSA algorithm, at the 6th iteration. [4 pts]

(b) Assume that $\lambda x > 1$ and that $\alpha < 1/10$. What is the greedy policy w.r.t. the estimated Q-function at the 6th iteration? [1 pt]

Exercise 2 Consider an MDP with a continuous state space S. Assume that for any given policy π, its discounted state value function V^π (for some discount factor λ) can be approximated using linear methods, i.e., that there exists $\theta \in \mathbb{R}^m$ such that $V^\pi(s) = V_\theta(s) = \theta^T \phi(s)$ for some function $\phi : S \to \mathbb{R}^m$, and for any $s \in S$. You wish to estimate V^π or equivalently θ. Let θ_t be your estimate of θ at time t, and suppose that you observe a trajectory

$$(s_t, a_t, r_t, s_{t+1}, \dots),$$

sampled according to π. Assume that you are given a learning rate α and the discount factor λ).

(a) Write the TD(0) algorithm with function approximation (the way you update θ_t). [3 pts]

(b) Your update rule should resemble something like $\theta_{t+1} = \theta_t + \alpha(b_t + A_t\theta_t)$. What are A_t and b_t? [1 pt]

(c) Suppose the algorithm has converged to some $\theta^\star$. Further assume that the Markov chain of the state under π is stationary ergodic. Compute $\theta^\star$ (remember that for an algorithm with updates $\theta_{t+1} = \theta_t + \alpha Y_t(\theta_t)$, a convergence point θ is such that $\mathbb{E}[Y_t(\theta)] = 0$ where $\mathbb{E}$ denotes the expectation w.r.t. to stationary distribution of $Y_t(\theta)$ – assuming it exists). [1 pt]

Problem 5

Policy gradient method. Consider an episodic RL problem with finite state-space $\mathcal{S}$, and finite action space $\mathcal{A} = \{1, \ldots, n+1\}$. For all $s \in \mathcal{S}$, $i \in \mathcal{A}$, let $\phi(s, i)$ be a feature vector in $\mathbb{R}^d$. We parameterize the policy using a vector $\theta \in \mathbb{R}^d$ via the following recursion: For $i \in \{1, \ldots, n\}$, initialize $i = 1$ and draw an independent random variable Z_i from $[0, 1]$. If $Z_i \leq \cos^2(\theta^\top \phi(s, i))$, then choose action $a = i$, otherwise, set $i \leftarrow i + 1$ and repeat. At the last step of the recursion if $Z_n > \cos^2(\theta^\top \phi(s, n))$, choose a = n+1.

(a) Compute in state s, the probability $\pi_\theta(s, i)$ of choosing action i. [2 pts]

(b) Compute the score $\nabla_\theta \log \pi_\theta(s, i)$. [2 pts]

(c) Write the REINFORCE algorithm update of θ upon observing an episode $\tau = (s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$. *Precise the update using explicitly the function ϕ, θ, and τ only.* [1 pt]

Variance reduction for policy gradient methods. Upon observing an epsiode $\tau = (s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$, we consider the following gradient estimate

$$\hat{g}(\theta) = \sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t) \sum_{u=t}^{T} r_u(s_u, a_u)$$

(d) Is $\hat{g}(\theta)$ an unbiased estimator of the gradient? If so, would you prefer to use this estimator or the one provided by the policy gradient theorem in the REINFORCE algorithm update? *Explain why!* [2 pts]

(e) Show that one can a add state-dependent baseline $b(s)$ in the expression of $\hat{g}(\theta)$ without changing its expected value. More precisely, prove that

$$\mathbb{E}[\hat{g}] = \mathbb{E}\left[\left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right)\left(\sum_{u=t}^{T} r_u(s_u, a_u) - b(s_t)\right)\right]$$

[2 pts]

(d) What could be the reason of adding such bias term? How would you chose the bias term? [1 pt]

Solution Problem 1.

(a) Because the samples are sampled according to the behavior policy, and not the policy that we actually want to improve. An idea to solve this problem is to use Importance sampling.

(b) To solve this problem we can resort to stochastic approximation, using the following scheme $\theta_{t+1} = \theta_t - \alpha_t(f(\theta_t) - \alpha)$.

(c) The complexity scales as AS^2T.

(d) Robbins-Monroe (stochastic approximation).

(e) Stochastic gradient.

(f) Because with this choice, we do not explore all actions infinitely often (hence the convergence of the underlying stochastic approximation algorithm is not guaranteed). This is due to the fact that $\sum_t \epsilon_t < \infty$.

(g) We update a parameter for the (state, action) value function of the current policy (critic), and a parameter for the policy (actor).

(h) No, because $\sum_t 1/\log(t)^2 = \infty$.

(i) Yes, you can easily verify the Markov property.

Solution – Problem 2

a) The time horizon is $T = 7$ and the state space is

$$S = \underbrace{\{(1+\rho_+)^i(1-\rho_-)^j; i+j \le T, i,j \in \mathbb{Z}_+\}}_{s\text{: market price}} \times \underbrace{\{0,1\}}_{o\text{: options left}}$$

and the action space is

$$A = \{0,1\}$$

where 1 represents using the option and 0 represents doing nothing.

The transition probabilities for the first coordinate, s, of S are given by a geometric random walk and is independent of your actions:

$$p(s' = s(1+\rho_+)) = p$$
$$p(s' = s(1-\rho-)) = 1-p$$

The transition probabilities of the second coordinate o of S depend only on the current state of o and your action:

$$p(o' = 1 | o = 1, a = 0) = 1 \qquad p(o' = 0 | o = 0, a = \cdot) = 1$$

and the remaining transitions occur with probability zero.

The goal of the agent is to maximize the expected cumulative reward

$$\mathbb{E} \sum_{t=1}^{T} r_t((s_t, o_t), a_t)$$

where

$$r_t((s_t, o_t), a_t) = \mathbf{1}_{o_t=1, a_t=1} \max(s_t - K, 0)$$

b) The rewards change to

$$r_t((s_t, o_t), a_t) = \mathbf{1}_{o_t=1, a_t=1} \max(K - s_t, 0)$$

and everything else remains unchanged.

c) The time horizon is $T = 365$ and the state space is

$$S = \underbrace{\{(1+\rho_+)^i(1-\rho_-)^j; i+j \le T, i,j \in \mathbb{Z}_+\}}_{s\text{: market price of stock 1}} \times \underbrace{\{0,1,\ldots,100\}}_{o\text{: options left}} \times \underbrace{\mathbb{R}_+}_{c\text{: cash}} \times \underbrace{\mathbb{Z}_+}_{f\text{: stocks in portfolio}}$$

and the action space is

$$A = \underbrace{\mathbb{Z}}_{b\text{: Stocks bought/sold}} \times \underbrace{\{0,1,\ldots,100\}}_{a\text{: options used}}$$

The transition probabilities for the first coordinate, s, of S are given by a geometric random walk and is independent of your actions:

$$p(s' = s(1+\rho_+)) = p$$
$$p(s' = s(1-\rho-)) = 1-p$$

The transition probabilities of your inventory (stocks, options, cash) are essentially deterministic and described by

$$p_t(o' = j, c' = q - l - (j - k)Kz, f' = i + l + k | s = z, o = j - k, c = q, f = i, b = l, a = k) = 1$$
$$\text{subject to } k \leq j, f \leq b \leq c/z$$

for $t < T - 1$ and

$$p_{T-1} = (c' = q + iz | s = zc = q, f = i) = 1 \qquad \text{(convert remaining assets to cash)}$$

and otherwise trivial.

The goal of the agent is to maximize the expected cumulative reward

$$\mathbb{E} \sum_{t=1}^{T-1} r_t((s_t, o_t, c_t, f_t), (b_t, l_t, a_t)) + r_T(s_T, c_T, f_T)$$

where

$$r_t = 0 \qquad t < T$$
$$r_T = c_T$$

Solution – Problem 3 (a) A finite time-horizon MDP with horizon $T = N$. The state is (t, w_t) (t is the number of the week) if you did not buy an apartment before week t, and $\emptyset$ if you bought an apartment earlier. The set of actions are 'B' for Buy and 'W' for Wait. The transition probabilities are as follows:

$$\begin{aligned} p((t+1,w)|(t,w'),W) &= f(w), \quad \forall (t,w,w') \\ p(\emptyset|(t,w'),B) &= 1, \quad \forall (t,w') \\ p(\emptyset|\emptyset,a) &= 1, \quad \text{for } a \in \{W,B\} \end{aligned}$$

The rewards are: $r((t,w),W) = 0$ and $r((t,w),B) = -\alpha^t w$.

(b) For $t = N$, you have to buy if you did not earlier, hence it is a threshold-based decision with threshold $\beta_N = \infty$.

For $t < N$, let us write Bellman's equation: let $V_t(w)$ the value at week t if $w_t = w$. Then:

$$V_t(w) = \max\{-\alpha^t w, \sum_x f(x) V_{t+1}(x)\}.$$

In particular, it is optimal to buy if and only if: the price $\alpha^t w$ is below the threshold $\beta_t = -\sum_x f(x) V_{t+1}(x)$.

(c) Note that:

$$V_{t+1}(x) = \max\{-\alpha^{t+1} x, \sum_y f(y) V_{t+2}(y)\} = \max\{-\alpha^{t+1} x, -\beta_{t+1}\}.$$

Hence we have:

$$\beta_t = \sum_x f(x) \min\{\alpha^{t+1} x, \beta_{t+1}\}.$$

(d) For the state, we add a binary variable X equal to 1 if you have an apartment, and 0 otherwise. The state can be $(t, 0, w_t)$ or $(t, 1, z_t)$. When $X = 0$, the actions are B or W. When $X = 1$ the actions are S (Sell) or W (Wait). The transition probabilities are as follows:

$$\begin{aligned} p((t+1,0,w)|(t,0,w'),W) &= f(w), \quad \forall (t,w,w') \\ p((t+1,1,w)|(t,0,w'),B) &= f(w), \quad \forall (t,w,w') \\ p((t+1,1,w)|(t,1,w'),W) &= f(w), \quad \forall (t,w,w') \\ p((t+1,0,w)|(t,1,w'),S) &= f(w), \quad \forall (t,w,w') \end{aligned}$$

The rewards are all 0 except for $r((t,0,w),B) = -\alpha^t w$ and $r((t,1,w),S) = \alpha^t w$. We can write Bellman's equations:

$$V_t(0,w) = \max\{-\alpha^t w + \sum_x f(x) V_{t+1}(1,x), \sum_x f(x) V_{t+1}(0,x)\}.$$

Hence the buying threshold is $\gamma_t = \sum_x f(x)(V_{t+1}(1,x) - V_{t+1}(0,x))$. Similarly:

$$V_t(1,w) = \max\{\alpha^t w + \sum_x f(x) V_{t+1}(0,x), \sum_x f(x) V_{t+1}(1,x)\}.$$

The selling threshold is $\zeta_t = \sum_x f(x)(V_{t+1}(1,x) - V_{t+1}(0,x))$. Note that $\gamma_t = \zeta_t$.

Solutions - Problem 4 - exercise 1

(A) The trajectory is

$$(B, u_1, 1); (B, u_2, 2); (C, u_1, 3); (A, u_3, -1); (C, u_2, 10); (A, u_2, -5); (A, u_1, \dots)$$

and Q^0 is

$$Q^{(0)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} .$$

Since we use SARSA, the Q-values in the different rounds are:

1. Step 1: the update is

   $$Q^{(1)}(B, u_1) = Q^{(0)}(B, u_1) + \alpha(1 + \lambda Q^{(0)}(B, u_2) - Q^{(0)}(B, u_1)) = \alpha(1 + \lambda x),$$

   thus

   $$Q^{(1)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ \alpha(1 + \lambda x) & x & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} .$$

2. Step 2: the update is

   $$Q^{(2)}(B, u_2) = Q^{(1)}(B, u_2) + \alpha(2 + \lambda Q^{(1)}(C, u_1) - Q^{(1)}(B, u_2)) = x + \alpha(2 - x) = x(1 - \alpha) + 2\alpha,$$

   thus

   $$Q^{(2)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ \alpha(1 + \lambda x) & x(1 - \alpha) + 2\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} .$$

3. Step 3: the update is

   $$Q^{(3)}(C, u_1) = Q^{(2)}(C, u_1) + \alpha(3 + \lambda Q^{(2)}(A, u_3) - Q^{(2)}(C, u_1)) = 3\alpha,$$

   thus

   $$Q^{(3)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ \alpha(1 + \lambda x) & x(1 - \alpha) + 2\alpha & 0 \\ 3\alpha & 0 & 1 \end{bmatrix} \end{array} .$$

4. Step 4: the update is

   $$Q^{(4)}(A, u_3) = Q^{(3)}(A, u_3) + \alpha(-1 + \lambda Q^{(3)}(C, u_2) - Q^{(3)}(A, u_3)) = -\alpha,$$

   thus

   $$Q^{(4)} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \\ \begin{bmatrix} 0 & 0 & -\alpha \\ \alpha(1 + \lambda x) & x(1 - \alpha) + 2\alpha & 0 \\ 3\alpha & 0 & 1 \end{bmatrix} \end{array} .$$

5. Step 5: the update is

$$Q^{(5)}(C,u_2) = Q^{(4)}(C,u_2) + \alpha(10 + \lambda Q^{(4)}(A,u_2) - Q^{(4)}(C,u_2)) = 10\alpha,$$

thus

$$Q^{(5)} = \begin{matrix} \\ A \\ B \\ C \end{matrix} \begin{matrix} \begin{matrix} u_1 \end{matrix} \\ \begin{bmatrix} 0 \\ \alpha(1+\lambda x) \\ 3\alpha \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} u_2 \end{matrix} \\ \begin{bmatrix} 0 \\ x(1-\alpha)+2\alpha \\ 10\alpha \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} u_3 \end{matrix} \\ \begin{bmatrix} -\alpha \\ 0 \\ 1 \end{bmatrix} \end{matrix}.$$

6. Step 6: the update is

$$Q^{(6)}(A,u_2) = Q^{(5)}(A,u_2) + \alpha(-5 + \lambda Q^{(5)}(A,u_1) - Q^{(5)}(A,u_2)) = -5\alpha,$$

thus

$$Q^{(6)} = \begin{matrix} \\ A \\ B \\ C \end{matrix} \begin{matrix} \begin{matrix} u_1 \end{matrix} \\ \begin{bmatrix} -5\alpha \\ \alpha(1+\lambda x) \\ 3\alpha \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} u_2 \end{matrix} \\ \begin{bmatrix} 0 \\ x(1-\alpha)+2\alpha \\ 10\alpha \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} u_3 \end{matrix} \\ \begin{bmatrix} -\alpha \\ 0 \\ 1 \end{bmatrix} \end{matrix}.$$

(B) The greedy policy in A is $\pi(A) = u_2$. In B we have that the greedy action is u_1 since $\lambda x > 1 + x(1/\alpha - 1)$ (note that $(1/\alpha - 1)$ is negative). In C the greedy action is simply u_3.

Solutions - Problem 4 - Exercise 2

(A) Using 1-step bootstrapping the target value is simply $y_t = r_t + \lambda V_{\theta_t}(s_{t+1}) = r_t + \lambda \theta_t^\top \phi(s_{t+1})$.
(B) An MSE loss is a loss of the type $\mathbb{E}[(V^\pi(s) - V_{\theta_t}(s))^2]$. Therefore, an algorithm that minimizes it is given by $\theta_{t+1} = \theta_t + \alpha \delta_t \nabla_\theta V_{\theta_t}(s)$, where $\delta_t = y_t - V_{\theta_t}(s_t)$ and y_t is the target estimate.

Since $\nabla_\theta V_\theta(s) = \phi(s)$, the SGD update simply is

$$\theta_{t+1} = \theta_t + \alpha(y_t - \theta_t^\top \phi(s_t))\phi(s_t) = \theta_t + \alpha(r_t + \lambda \theta_t^\top \phi(s_{t+1}) - \theta_t^\top \phi(s_t))\phi(s_t)$$

(C) In this case $b_t = \alpha r_t \phi(s_t)$ and $A_t = \alpha \phi(s_t)(\lambda \phi(s_{t+1})^\top - \phi(s_t)^\top)$.
(D) At convergence $\theta_{t+1} = \theta_t$. Therefore $\theta^\star = \theta^\star + A\theta^\star + b$, from which follows that the solution is $\theta^\star = -A^{-1}b = -\mathbb{E}[\alpha\phi(s_t)(\lambda\phi(s_{t+1})^\top - \phi(s_t)^\top)]^{-1}\mathbb{E}[\alpha r_t \phi(s_t)]$.

Solution – Problem 5

(a) We can easily compute the probabilities

$$\begin{aligned}
\pi_\theta(s,1) &= \cos^2(\theta^\top \phi(s,1)) \\
\pi_\theta(s,i) &= \prod_{j=1}^{i-1} \sin^2(\theta^\top \phi(s,j)) \cos^2(\theta^\top \phi(s,i)) \quad i = 2,\dots,i \\
\pi_\theta(s,n+1) &= \prod_{j=1}^{n} \sin^2(\theta^\top \phi(s,j))
\end{aligned}$$

(b)

$$\begin{aligned}
\nabla \log \pi_\theta(s,1) &= -2\tan(\theta^\top \phi(s,1))\phi(s,1) \\
\nabla \log \pi_\theta(s,i) &= \sum_{j=1}^{i-1} \frac{2}{\tan(\theta^\top \phi(s,j))}\phi(s,j) - 2\tan(\theta^\top \phi(s,i))\phi(s,i) \quad for \quad i = 2,\dots,n \\
\nabla \log \pi_\theta(s,n+1) &= \sum_{j=1}^{n} \frac{2}{\tan(\theta^\top \phi(s,j))}\phi(s,j)
\end{aligned}$$

(c) We can express the update as follows

$$\theta \leftarrow \theta + \alpha \left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t,a_t) \right) \left(\sum_{t=1}^{T} r_t(s_t,a_t) \right)$$

with $\nabla \log \pi_\theta(s_t,a_t)$ as defined in the solution of (b).

(d) Yes, $\hat{g}(\theta)$ is an ubiased estimator. One should use the estimator $\hat{g}(\theta)$ instead of standard gradient estimator presented in the policy gradient theorem. The reason is $\hat{g}(\theta)$ has a lower variance.

(e) Yes, it is easy to verify that

$$\forall u \geq t, \quad \mathbb{E}[\nabla \log \pi_\theta(s_t,a_t) b(s_u)] = 0$$

(f) The reason is to reduce the variance of the gradient estimator. One possible choice is to use an estimate of the value function $\hat{V}_t^\pi(s_t)$.