Robot Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

Winter Semester 2016, Homework 3 (54 points + 8 bonus)

Prof. Dr. J. Peters, F. Veiga, S. Parisi

Due Date: Tuesday, 31 January 2017 (3 pm)

Problem 3.1 Optimal Control [20 Points]

In this exercise, we consider a finite-horizon discrete time-varying Stochastic Linear Quadratic Regulator with Gaussian noise and time-varying quadratic reward function. Such system is defined as

$$\boldsymbol{s}_{t+1} = \boldsymbol{A}_t \boldsymbol{s}_t + \boldsymbol{B}_t \boldsymbol{a}_t + \boldsymbol{w}_t, \tag{1}$$

where $\boldsymbol{s}_t$ is the state, $\boldsymbol{a}_t$ is the control signal, $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{b}_t, \Sigma_t)$ is Gaussian additive noise with mean $\boldsymbol{b}_t$ and covariance Σ_t and $t = 0, 1, \ldots, T$ is the time horizon. The control signal $\boldsymbol{a}_t$ is computed as

$$\boldsymbol{a}_t = -\boldsymbol{K}_t \boldsymbol{s}_t + \boldsymbol{k}_t \tag{2}$$

and the reward function r_t is

$$r_t = \begin{cases} -(\boldsymbol{s}_t - \boldsymbol{r}_t)^\mathsf{T} \boldsymbol{R}_t (\boldsymbol{s}_t - \boldsymbol{r}_t) - \boldsymbol{a}_t^\mathsf{T} \boldsymbol{H}_t \boldsymbol{a}_t & \text{when} \quad t = 0, 1, \ldots, T-1 \\ -(\boldsymbol{s}_t - \boldsymbol{r}_t)^\mathsf{T} \boldsymbol{R}_t (\boldsymbol{s}_t - \boldsymbol{r}_t) & \text{when} \quad t = T \end{cases} \tag{3}$$

Note: r_t and $\boldsymbol{r}_t$ are different!

Note 2: the notation used in Marc Toussaint's notes "*(Stochastic) Optimal Control*" is different from the one used in the lecture's slides.

a) **Implementation [8 Points]**

Implement the LQR with the following properties

$$\boldsymbol{s}_0 \sim \mathcal{N}(0, 1) \qquad T = 50$$

$$\boldsymbol{A}_t = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{B}_t = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$$

$$\boldsymbol{b}_t = \begin{bmatrix} 5 \\ 0 \end{bmatrix} \qquad \Sigma_t = 0.01$$

$$\boldsymbol{K}_t = \begin{bmatrix} 5 & 0.3 \end{bmatrix} \qquad \boldsymbol{k}_t = 0.3$$

$$\boldsymbol{H}_t = 1 \qquad \boldsymbol{R}_t = \begin{cases} \begin{bmatrix} 100000 & 0 \\ 0 & 0.1 \end{bmatrix} & \text{if} \quad t = 14, 40 \\ \begin{bmatrix} 0.01 & 0 \\ 0 & 0.1 \end{bmatrix} & \text{otherwise} \end{cases} \qquad \boldsymbol{r}_t = \begin{cases} \begin{bmatrix} 10 \\ 0 \end{bmatrix} & \text{if} \quad t = 0, 1, \ldots, 14 \\ \begin{bmatrix} 20 \\ 0 \end{bmatrix} & \text{if} \quad t = 15, 16, \ldots, T \end{cases}$$

Execute the system 20 times. Plot the mean and 95% confidence over the different experiments of the state $\boldsymbol{s}_t$ and of the control signal $\boldsymbol{a}_t$ over time. How does the system behave? Compute and write down the mean and the standard deviation of the cumulative reward over the experiments. Attach a snippet of your code.

b) **LQR as a P controller [4 Points]**

The LQR can also be seen as a simple P controller of the form

$$\boldsymbol{a}_t = \boldsymbol{K}_t \left(\boldsymbol{s}_t^{\text{des}} - \boldsymbol{s}_t \right) + \boldsymbol{k}_t \,, \tag{4}$$

which corresponds to the controller used in the canonical LQR system with the introduction of the target $\boldsymbol{s}_t^{\text{des}}$. Assume as target

$$\boldsymbol{s}_t^{\text{des}} = \boldsymbol{r}_t = \begin{cases} \begin{bmatrix} 10 \\ 0 \end{bmatrix} & \text{if} \quad t = 0, 1, \ldots, 14 \\ \begin{bmatrix} 20 \\ 0 \end{bmatrix} & \text{if} \quad t = 15, 16, \ldots, T \end{cases} \tag{5}$$

Use the same LQR system as in the previous exercise and run 20 experiments. Plot in one figure the mean and 95% confidence of the first state, for both $\boldsymbol{s}_t^{\text{des}} = \boldsymbol{r}_t$ and $\boldsymbol{s}_t^{\text{des}} = \boldsymbol{0}$.

c) **Optimal LQR [8 Points]**

To compute the optimal gains $\boldsymbol{K}_t$ and $\boldsymbol{k}_t$, which maximize the cumulative reward, we can use an analytic optimal solution. This controller recursively computes the optimal action by

$$\boldsymbol{a}^* = -\left(\boldsymbol{H}_t + \boldsymbol{B}_t^T \boldsymbol{V}_{t+1} \boldsymbol{B}_t \right)^{-1} \boldsymbol{B}_t^T \left(\boldsymbol{V}_{t+1} \left(\boldsymbol{A}_t \boldsymbol{s}_t + \boldsymbol{b}_t \right) - \boldsymbol{v}_{t+1} \right), \tag{6}$$

which can be decomposed into

$$\boldsymbol{K}_t = -\left(\boldsymbol{H}_t + \boldsymbol{B}_t^T \boldsymbol{V}_{t+1} \boldsymbol{B}_t \right)^{-1} \boldsymbol{B}_t^T \boldsymbol{V}_{t+1} \boldsymbol{A}_t, \tag{7}$$

$$\boldsymbol{k}_t = -\left(\boldsymbol{H}_t + \boldsymbol{B}_t^T \boldsymbol{V}_{t+1} \boldsymbol{B}_t \right)^{-1} \boldsymbol{B}_t^T \left(\boldsymbol{V}_{t+1} \boldsymbol{b}_t - \boldsymbol{v}_{t+1} \right). \tag{8}$$

where

$$\boldsymbol{M}_t = \boldsymbol{B}_t \left(\boldsymbol{H}_t + \boldsymbol{B}_t^T \boldsymbol{V}_{t+1} \boldsymbol{B}_t \right)^{-1} \boldsymbol{B}_t^T \boldsymbol{V}_{t+1} \boldsymbol{A}_t \tag{9}$$

$$\boldsymbol{V}_t = \begin{cases} \boldsymbol{R}_t + \left(\boldsymbol{A}_t - \boldsymbol{M}_t \right)^T \boldsymbol{V}_{t+1} \boldsymbol{A}_t & \text{when} \quad t = 1 \ldots T-1 \\ \boldsymbol{R}_t & \text{when} \quad t = T \end{cases} \tag{10}$$

$$\boldsymbol{v}_t = \begin{cases} \boldsymbol{R}_t \boldsymbol{r}_t + \left(\boldsymbol{A}_t - \boldsymbol{M}_t \right)^T \left(\boldsymbol{v}_{t+1} - \boldsymbol{V}_{t+1} \boldsymbol{b}_t \right) & \text{when} \quad t = 1 \ldots T-1 \\ \boldsymbol{R}_t \boldsymbol{r}_t & \text{when} \quad t = T \end{cases} \tag{11}$$

Run 20 experiments with $\boldsymbol{s}_t^{\text{des}} = \boldsymbol{0}$ computing the optimal gains $\boldsymbol{K}_t$ and $\boldsymbol{k}_t$. Plot the mean and 95% confidence of both states for all three different controllers used so far. Use one figure per state. Report the mean and std of the cumulative reward for each controller and comment the results. Attach a snippet of your code.

Robot Learning - Homework 3

Name, Vorname: _____ Matrikelnummer: ⌊_⌋⌊_⌋⌊_⌋⌊_⌋⌊_⌋⌊_⌋⌊_⌋

Problem 3.2 Reinforcement Learning [34 Points + 8 Bonus]

You recently acquired a robot for cleaning you apartment but you are not happy with its performance and you decide to reprogram it using the latest AI algorithms. As a consequence the robot became self-aware and, whenever you are away, it prefers to play with toys rather than cleaning the apartment. Only the cat has noticed the strange behavior and attacks the robot. The robot is about to start its day and its current perception of the environment is as following

Your graphic could be here. It just wasn't included.

The black squares denote extremely dangerous states that the robot must avoid to protect its valuable sensors. The reward of such states is set to $r_{\text{danger}} = -10^5$ (NB: the robot can still go through these states!). Moreover, despite being waterproof, the robot developed a phobia of water (W), imitating the cat. The reward of states with water is $r_{\text{water}} = -100$. The robot is also afraid of the cat (C) and tries to avoid it at any cost. The reward when encountering the cat is $r_{\text{cat}} = -3000$. The state containing the toy (T) has a reward of $r_{\text{toy}} = 1000$, as the robot enjoys playing with them. Some of the initial specification still remain, therefore the robot receives $r_{\text{dirt}} = 35$ in states with dirt (D).

State rewards can be collected at every time the robot is at that state. The robot can perform the following actions: *down, right, up, left* and *stay*.

In our system we represent the actions with the an ID (0, 1, 2, 3, 4), while the grid is indexed as $\{\texttt{row}, \texttt{column}\}$. The robot can't leave the grid as it is surrounded with walls. A skeleton of the gridworld code and some plotting functions are available at the webpage. For all the following questions, always attach a snippet of your code.

a) **Finite Horizon Problem [14 Points]**

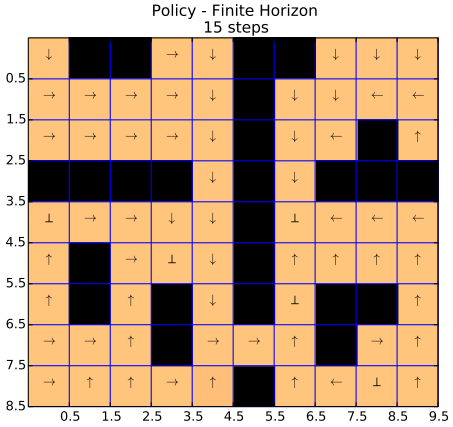
In the first exercise we consider the finite horizon problem, with horizon $T = 15$ steps. The goal of the robot is to maximize the expected return

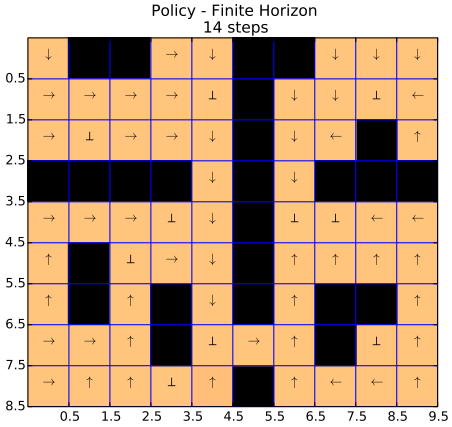
$$J_\pi = \mathbb{E}_\pi \left[\sum_{t=1}^{T-1} r_t(s_t, a_t) + r_T(s_T) \right], \tag{12}$$

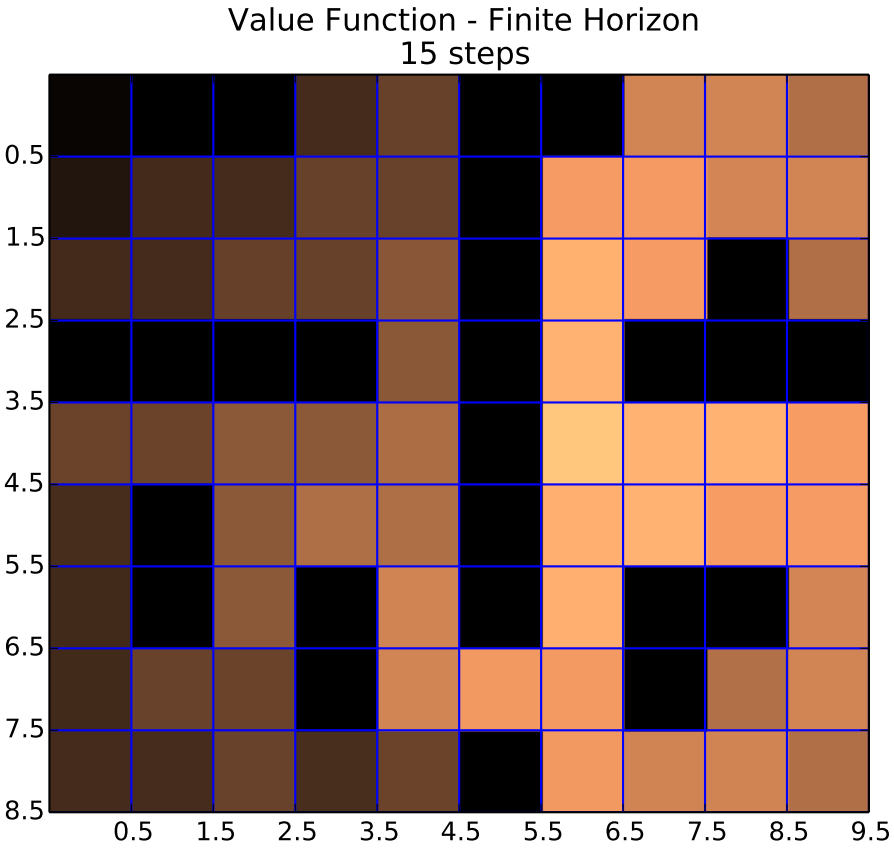
according to policy π, state s, action a, reward r, and horizon T. Since rewards in our case are independent of the action and the actions are deterministic, Equation (12) becomes

$$J_\pi = \sum_{t=1}^{T} r_t(s_t). \tag{13}$$

Using the Value Iteration algorithm, determine the optimal action for each state when the robot has 15 steps left. Attach the plot of the policy to your answer and a mesh plot for the value function. Describe and comment the policy: is the robot avoiding the cat and the water? Is it collecting dirt and playing with the toy? Which would be the time horizon that makes the robot acts differently in state $(9, 4)$?







Our policy seems appropriate given our explanation. The robot does collect dirt if it is far enough away from the toy (5,1) but otherwise will prioritize getting towards the toy. It seems so reckless that it even walks over the cat space to get to it when starting in (9,4). This however changes once the robot is limited to 14 steps forcing our robot to simply collect dirt beside the cat in (9,4).

b) **Infinite Horizon Problem - Part 1 [4 Points]**

We now consider the infinite horizon problem, where $T = \infty$. Rewrite Equation (12) for the infinite horizon case adding a discount factor γ. Explain briefly why the discount factor is needed.

c) **Infinite Horizon Problem - Part 2 [6 Points]**

Calculate the optimal actions with the infinite horizon formulation. Use a discount factor of $\gamma = 0.8$ and attach the new policy and value function plots. What can we say about the new policy? Is it different from the finite horizon scenario? Why?

d) **Finite Horizon Problem with Probabilistic Transition Function [10 Points]**

After a fight with the cat, the robot experiences control problems. For each of the actions *up, left, down, right*, the robot has now a probability 0.7 of correctly performing it and a probability of 0.1 of performing another action according to the following rule: if the action is *left* or *right*, the robot could perform *up* or *down*. If the action is *up* or *down*, the robot could perform *left* or *right*. Additionally, the action can fail causing the robot to remain on the same state with probability 0.1. Using the finite horizon formulation, calculate the optimal policy and the value function. Use a time horizon of $T = 15$ steps as before. Attach your plots and comment them: what is the most common action and why does the learned policy select it?

e) **Reinforcement Learning - Other Approaches [8 Bonus Points]**

What are the two assumptions that let us use the Value Iteration algorithm? What if they would have been not satisfied? Which other algorithm would you have used? Explain it with your own words and write down its fundamental equation.