Stochastic Control and Optimization Course (EL2800)

Computer Lab 2 – The Great Heist

December 7, 2016

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

Please send your answers and your code by email to magur@kth.se and mstms@kth.se before Friday December 16, 5PM. Good luck!

Problem 1: Bank Robbing: Reloaded

You are a bank robber trying to heist the bank of an unknown town. You enter the town at position A in the grid in Figure 1, the police starts from the opposite corner and the bank is at position B. For each round spent in the bank, you receive a reward of 1 SEK. The police walks randomly (i.e. uniformly at random up, down, left or right) across the grid and whenever you are caught (i.e. you are on the same cell as the police) you lose 10 SEK. You are new in town, and hence oblivious to the value of the rewards, the position of the bank, the starting point and the movement strategy of the police. Before you take an action (move up, down, left, right or stand still), you can observe both your position and that of the police (which jointly define the states of the underlying MDP). Your task is to develop an algorithm learning the policy that maximizes your total discounted reward (discount factor $\lambda = 0.8$)

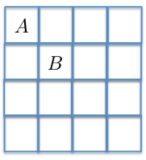

Figure 1: The grid of the town.

(a) Solve the problem by implementing the Q-learning algorithm exploring actions uniformly at random. Create a plot showing the convergence of your learned policy. (**Note:** Expect the policy to converge after roughly 100 000 iterations.)

(b) Solve the problem by implementing the SARSA algorithm using ε-greedy exploration (initially $\varepsilon = 0.1$). Try different values of ε.

Problem 2: The Inverted Pendulum

We consider the problem of balancing a pendulum of length l on the finger. For simplicity, the pendulum can only move on a vertical plane, and the finger can only move on a horizontal line in this plane as shown in the figure below. Denote by g the gravitational acceleration. Let us introduce the state variables $x_1 = y$ and $x_2 = \dot{y}$, then we obtain the following *nonlinear* state space description:

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= \frac{2g}{l}\sin(x_1) - \frac{2}{l}u\cos(x_1) \\ y &= x_1 \end{aligned}$$

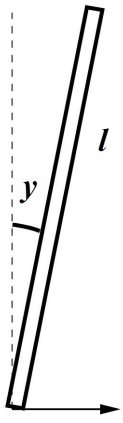

Figure 2: The inverted pendulum.

Since the system is linearized around $x = 0$ and $u = 0$, we can approximate the above description by the following *linear* state space model:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ \frac{2g}{l} & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -\frac{2}{l} \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned}$$

Denote by $\alpha = 2g/l$ and $\beta = -2/l$. The discrete-time model is then given by:

$$x(t+1) = Ax(t) + Bu(t) + w(t),$$

where $w(t)$ is a Gaussian noise process with 0 mean and unit covariance matrix and:

$$A = \begin{bmatrix} \frac{e^{\sqrt{\alpha}}}{2} & \frac{e^{\sqrt{\alpha}} - e^{-\sqrt{\alpha}}}{2} \\ \frac{\sqrt{\alpha}}{2}(e^{\sqrt{\alpha}} - e^{-\sqrt{\alpha}}) & \frac{e^{\sqrt{\alpha}} + e^{-\sqrt{\alpha}}}{2} \end{bmatrix}$$

and

$$B = \beta[e^{\sqrt{\alpha}} - e^{-\sqrt{\alpha}}] \begin{bmatrix} \alpha^{-1} \\ (2\sqrt{\alpha})^{-1} \end{bmatrix}.$$

(a) Implement a controller that minimized the cost function $\sum_{t=1}^{T} y_t^2 + \sum_{t=1}^{T-1} \mu u_t^2$ for a time horizon of $T = 20$, for parameters $\mu = 1, 5, 20$, and 100. Plot the evolution of y for each μ.

(b) Consider the task in part (a) with $\mu = 10$, and describe the behavior of the controller when T grows large.