Problem Set 3 for ABE 598 Autonomous Decision Making in the Real World

Total 150 points

Due on May 2 2018

The problems in this problem set are computer implementation problems. You are free to use any programming language of your choice, including MATLAB. You will submit the code with your assignment. Your code will be graded on its style, commenting, and readability. You are suggested to use good programming techniques, such as writing functions for repeating tasks, indenting your code properly, and choosing most efficient operations (e.g. if possible use matrix manipulations in MATLAB instead of for loops).

If you use online resources or collaborate with others to write your software, please make sure you are citing them correctly. Provide enough comments in your code to illustrate your understanding of the algorithmic process.

P1: MDPs and RL on a grid world

Consider a 5 by 5 grid world. We would like to find a path from one end of the grid to the other end, that is from location (1,1) to location (5,5). A reward of 1 is obtained when the agent reaches (5,5), a reward of -0.01 is obtained for every transition. There are 5 actions: stay, go left, go up, go right, go down. Each action results in a stochastic transition, with 90% probability of the intended transition, and 10% probability of a random transition.

What would be the state transition matrix, what would be the reward function?

Assuming the state transition and reward function is known, implement the following algorithms to find this path:

1. Value iteration (Alg 2 from Geramifard et al.)
2. TBVI (Alg 3 from Geramifard et al.)

Assuming that the state transition and the reward matrices are not known, implement the following reinforcement learning algorithm:

1. SARSA (Alg 6 from Geramifard et al.)

For all algorithms, implement the following value function approximations:

1. Tabular
2. Radial Basis Function network, limit the maximum number of basis to 20
3. (Bonus) Gaussian Process (5 points extra for each algorithm you implement GP for)

An implementation of Q learning with Tabular, Radial Basis, and GP function approximator has been provided. The main file is gridworld_QLGP_main.m. It has been commented for your review. You need not use this implementation if you do not wish to.

For implementing the SARSA algorithm, you could change the Q learning update law with the SARSA one in the given code.

For implementing the MDP algorithms, you have you change the code to use the transition and reward models in the computation of the policy. Else, you can write code from scratch, which in this case could be easier since an episodic treatment that is done in the RL code is not needed for solving MDPs when the reward and transition models are known.

Note that the transitions in this grid world are stochastic, as opposed to the deterministic transitions in the grid world in Pset 2.

Present your answer as plots and a discussion. For the dynamic programming problems, you will evaluate the performance of your algorithm over a series of 100 Monte-Carlo runs. The stochasticity comes from the random transitions.

The RL algorithms are set to run in 5 executions of 200 episodes each with 100 evaluations (samples).

The files given to you plot the results with their mean and standard deviations. This is the kind of plot you should be presenting.