Last name: Li First name: Minghan SID#: 1561234

Collaborators:

CMPUT 366/609 Assignment 6: Prediction with linear function approximation

Due: Tuesday Nov 21st by gradescope

There are a total of 100 points available on this assignment, as well as 15 bonus points

The first question is an exercise from the Sutton and Barto textbook, 2nd edition:

Question 1. [10 points] Exercise 9.4 (*tile coding two dimensions*)

Question 2. **Programming Exercise [90 points, three parts].**

Part 1 [20 points]. Implement the 1000 state random walk described in Example 9.1. You will make an RL-glue environment program that implements the 1000 state random walk described in chapter 9, example 9.1.

Please submit your environment program for this part.

Part 2 [50 points]. Implement three prediction agents based on TD(0); each using a different function approximation schemes in RL-glue.

Tabular feature encoding: agent one will implement Semi-gradient TD(0) (described on page 166) with a tabular or "one-hot" state encoding. A tabular encoding of the current state uses a unique feature vector for each state, where the number of components in the feature vector is equal the number of states. For example imagine an MDP with 4 states. The feature vectors corresponding to each state would be: [1,0,0,0], [0,1,0,0],[0,0,1,0],[0,0,0,1] for states 1,2,3,4 respectively. Test your semi-gradient TD(0) agent with alpha=0.5, tabular features on your 1000 state random walk environment.

Tile coding features: agent two will implement Semi-gradient TD(0) with tile coding. You will use the supplied tiling coding software (tiles3.py). In this case we treat the state number as a continuous number and tile code it producing a list of active features equal to the number of tilings. You will use number of tilings = 50, and tile width equal to 0.2. For TD(0) use alpha = 0.01/50.

Tile coder documentation can be found here: **http://www.incompleteideas.net/sutton/tiles/tiles3.html**

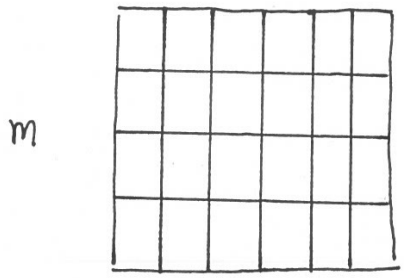
State aggregation: agent three will implement Semi-gradient TD(0) with state aggregation. You can implement state aggregation, as described in Example 9.1. "For the state aggregation, the 1000 states were partitioned into 10 groups of 100 states each (i.e., states 1–100 were one group, states 101–200 were another, and so on)." Or you can use the tile coder with num tilings equal to one and a tile width of 0.1 to achieve state aggregation. For semi-gradient TD(0) use alpha = 0.1.

Please submit your three agents for this part. They can be in one agent program with programatic way to switch between them, or three different agent program files. Either is fine.

Question 1

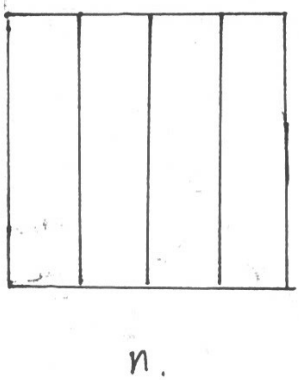
Coarse tiles with many tilings can give good Generalization, since we want to have generalization to be primarily across one dimension, there are two schemes we can use:

Scheme 1: Rectangular Tiling



$m < n$ such that the tile would be rectangular (suppose the space is a square), in this way we are allowing more generalization over y-axis

Scheme 2: Stripe Tiling



It would be similar as in scheme 1 where the generalization is across y-axis but instead has much broader generalization along y-axis for each tile than in scheme 1.