CS221 Fall 2015 Homework 4

SUNet ID: lguan
Name: Leying Guan

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

The results below are acquired after 2 iterations with updating order being -1, 0, 1(it is always 0 for two end-points), and action 1 has probability 0.3 of increasing 1 and action -1 has probability 0.2 of increasing 1.

(a) iter 0:[0, 0, 0, 0, 0]

iter 1:[0.0, 15.0, 7.0, 31.4, 0.0]

iter 2:[0.0, 16.4, 15.90, 37.63, 0.0]

(b) optimal policy after 2 iterations is [0, -1.0, 1.0, 1.0, 0]

Problem 2

(a) Counter example is given in submission.py

(b) When the graph is acyclic, we can update from leafs to roots: first calculate nodes which do not depend on others and then sequentially update nodes whose values depend only on nodes which have already been updated.

(c) Let the reward become 1/discount× original value; after convergence, times everything with discount.

Problem 3

(a) Code

(b) Code

Problem 4

(a) Code.

(b) For the small problem, the learned policy is the same as the actual optimal policy. For the large problem, there are 869 difference state where the learned policy and the actual optimal policy are not consistent. One potential reason is that our feature space is too contrained and could not represent the underlying truth well.

(c) Code.Now using the new feature extractor, there are only a dozen of different states where two policies are not consistent.

(d) We see that when we fix an optimal policy, its good performance depands largely on whether our knowledge about the reward, transition probability is correct, unlike the reinforcement learning approach which does not depend on human's prior. After we change the parameters of MDP problem, reinforcement learninng produced results around 25% better then fixed policy strategy(across 30000 repetitions).