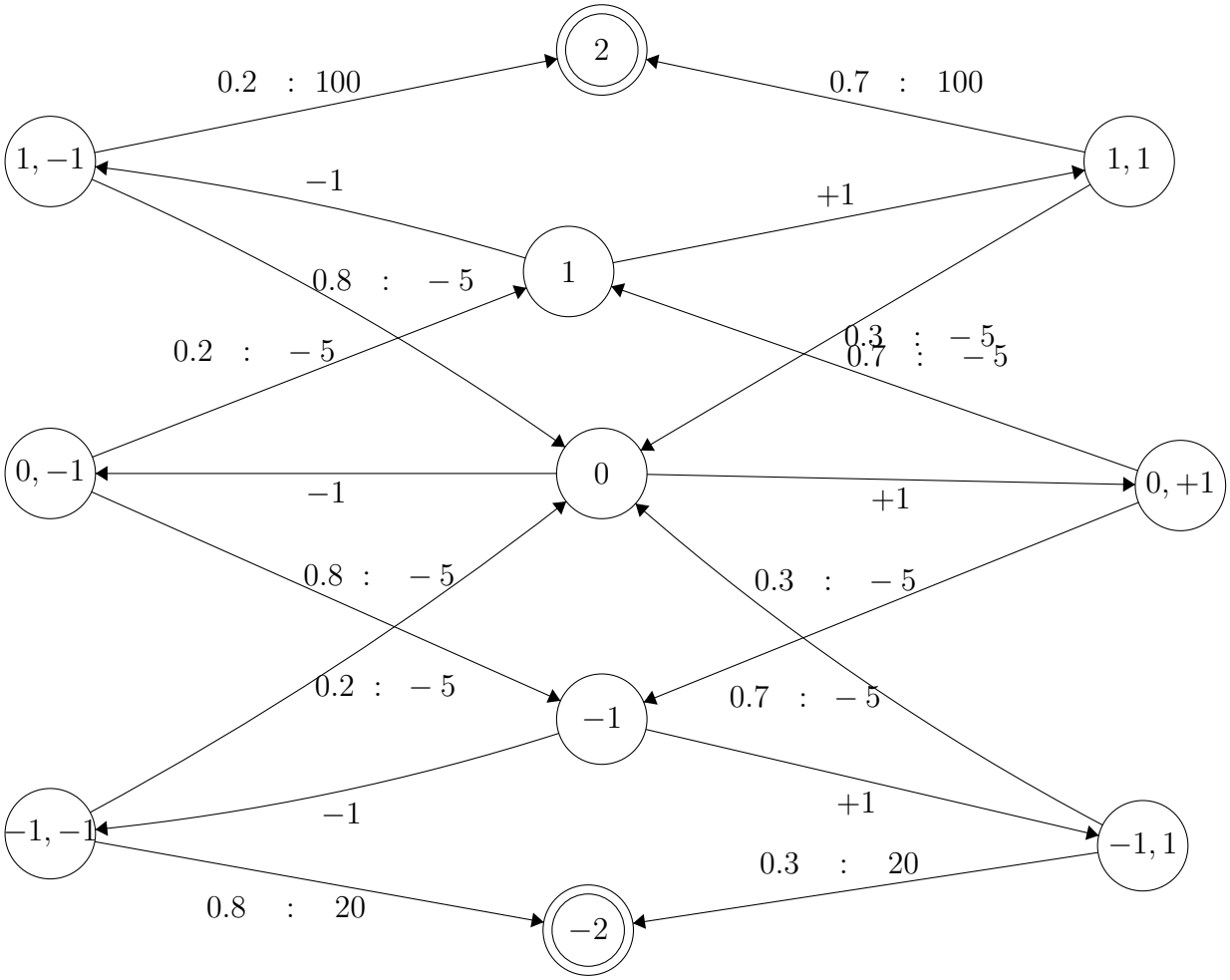
CS221 Assignment 4 Blackjack

Hassan Fahmy

April 2019

Problem 1



(a) Zero Iteration:

$V_{opt}^{(0)}(2) = 0$

$V_{opt}^{(0)}(1) = 0$

$V_{opt}^{(0)}(0) = 0$

$V_{opt}^{(0)}(-1) = 0$

$V_{opt}^{(0)}(-2) = 0$

First Iteration

$V_{opt}^{(1)}(2) = 0$

$V_{opt}^{(1)}(1) = 0.7(100) + 1(0)) + 0.3(-5 + 1(0)) = 68.5$

$V_{opt}^{(1)}(0) = -5$ //all choices have -5 reward and all previous values are zero

$V_{opt}^{(1)}(-1) = 0.8(20 + 1(0)) + 0.2(-5 + 1(0)) = 15$

$V_{opt}^{(1)}(-2) = 0$

Second Iteration

$V_{opt}^{(2)}(2) = 0$

$V_{opt}^{(2)}(1) = 0.7(100 + 1(0)) + 0.3(-5 + 1(0)) = 68.5$

$V_{opt}^{(2)}(0) = 0.7(-5 + 1(68.5)) + 0.3(-5 + 1(15)) = 47.45$

$V_{opt}^{(2)}(-1) = 0.8(20 + 1(0)) + 0.2(-5 + 1(-5))) = 14$

$V_{opt}^{(2)}(-2) = 0$

Third Iteration

$V_{opt}^{(2)}(2) = 0$

$V_{opt}^{(2)}(1) = 0.7(100 + 1(0)) + 0.3(-5 + 1(0)) = 68.5$

$V_{opt}^{(2)}(0) = 0.7(-5 + 1(68.5)) + 0.3(-5 + 1(15)) = 47.45$

$V_{opt}^{(2)}(-1) = 0.7(-5 + 1(47.45)) + 0.3(20 + 1(0))) = 35.715$

$V_{opt}^{(2)}(-2) = 0$

(b) The final policy would be to go to the s+1 node for all non terminal nodes because from the third iteration onward all values reach their maximum when the +1 action is taken.

Problem 2

(a)

(b) We can either topologically sort the states in the problem from end state to start state then start calculating the value starting from the end node or we can simply use the same algorithm but instead of getting the $V_{opt}s'$ from the previous iteration we get it by recursively calling the function we are using and use dynamic programming to calculate each value only once.

(c) $T'(s, a, s') = \Upsilon T(s, a, s')$ for all s' in $States$, $T'(s, a, o) = (1 - \Upsilon)$
$Rewards'(s, a, s') = Rewards(s, a, s')$ for all s' in $States$, $Rewards'(s, a, o) = 0$
It works because with probability $(1 - \gamma)$ it terminates at each state so the probability of each state in the future is multiplied by a factor of gamma for each step away.

Problem 4

(a)

(b) For small MDP approximately 84 percent of the states had the same actions with value iteration and Qlearning. For large MDP this is the case for only 65 percent of the states. This could be because it is possible that Qlearning does not visit all the states and actions like value iteration. and because of that b=value iteration might get better results with the large mdp where Qlearning cant cover as much.

(c)

(d) With q learning the expected reward is approximately none while with the fixed RL it is roughly seven. Qlearnong worked better becasue it can modify weights while fixed rl can not. So qlearnong is better with novel data and when the threshold changes it can chenge the weights