Problem Set VII: Value & Policy Iteration

Aim The purpose of this workshop is to help you get a better understanding of MDPs, value iteration, and policy iteration.

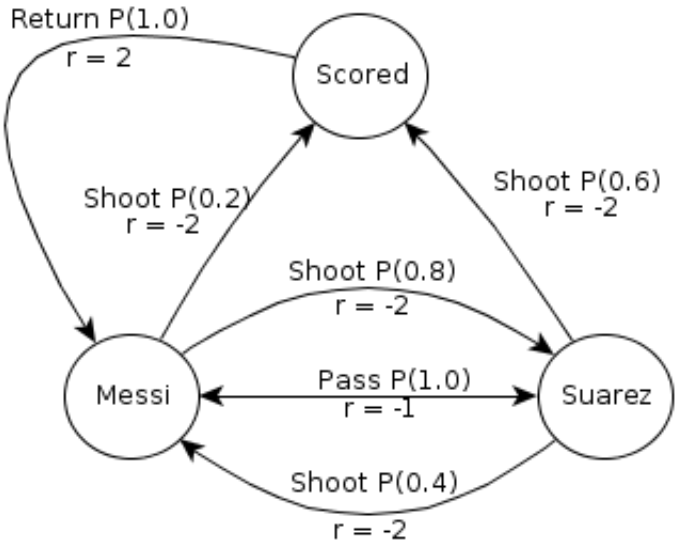
Consider two football-playing robots: Messi and Suarez.

They play a simple two-player cooperate game of football, and you need to write a controller for them. Each player can pass the ball or can shoot at goal.

The football game can be modelled as a discounted-reward MDP with three states: $Messi$, $Suarez$ (denoting who has the ball), and $Scored$ (denoting that a goal has been scored); and the following action descriptions:

- If Messi shoots, he has 0.2 chance of scoring a goal and a 0.8 chance of the ball going to Suarez. Shooting towards the goal incurs a cost of 2 (or a reward of -2).
- If Suarez shoots, he has 0.6 chance of scoring a goal and a 0.4 chance of the ball going to Messi. Shooting towards the goal incurs a cost of 2 (or a reward of -2).
- If either player passes, the ball will reach its intended target with a probability of 1.0. Passing the ball incurs a cost 1 (or a reward of -1).
- If a goal is scored, the only action is to return the ball to Messi, which has a probability of 1.0 and has a reward of 2. Thus the reward for scoring is modelled by giving a reward of 2 when *leaving* the goal state.

The following diagram shows the transition probabilities and rewards:



 State(Messi)

Tasks

1. Assume that we have calculated the following *non-optimal* value function V for this problem using value iteration with $\gamma = 1.0$, after iteration 2 we arrive at the following:

Iteration		0	1	2	3
V(Messi)	=	0.0	-1.0	-2.0	
V(Suarez)	=	0.0	-1.0	-1.2	
V(Scored)	=	0.0	2.0	1.0	

   If Messi has the ball (the system is in the Messi state), what action should we choose to maximise our reward in the next state: pass or shoot? Assume we are using the values for V after three iterations.

2. Complete the values of these states for iteration 3 using value iteration. Show your working.

3. Consider the following policy update table and policy evaluation table, with discount factor $\gamma = 0.8$:

Iter	$Q^\pi(Messi, P)$	$Q^\pi(Messi, S)$	$Q^\pi(Suarez, P)$	$Q^\pi(Suarez, S)$	$Q^\pi(Scored)$
0	0	0	0	0	0
1	-4	-5.52	-4	-4.56	-2
2	-4.194	-5.465	-4.355	-3.993	-1.355

   Apply two iterations of policy iteration. Finish both tables and show the working for the policy evaluation and policy update.

   What is the policy after two iterations?

Iter	$\pi(Messi)$	$\pi(Suarez)$	$\pi(Scored)$
0	Pass	Pass	Return
1	Pass	Shoot	Return
2	Pass	Shoot	Return

$V(Me, S) = 0.8 \cdot [-2 + 0.8 \cdot V^\pi(Su)] + 0.2 \times [-2 + V^\pi(S)]$

$= 0.8 \times (-2 + (-4)) + 0.2 \times (-2 + -1.6)$

$= -4.8 + (0.72)$

$= -4.12$

a

$V^\pi(Messi) = 1 \cdot [-1 + 0.8 \cdot V^\pi(Suarez)] = -1 + 0.8 \cdot V^\pi(Su)$

b

$V^\pi(Suarez) = 1 \cdot [-1 + 0.8 \cdot V^\pi(Messi)] = -1 + 0.8 \cdot V^\pi(Me)$

c

$V^\pi(Score) = 1 \cdot [2 + 0.8 \times V^\pi(Messi)] = 2 + 0.8 \times V^\pi(Me)$

$a = -1 + 0.8b$

$b = -1 + 0.8a$

$c = 2 + 0.8a$

$0.8a = -0.8 + 0.64b$

$-1 + 0.8a = b$

$-1 = 0.8 + 0.36b$

$b = -5 \quad a = -5 \quad c = -2$

Additional Tasks for Personal Study

To improve your understanding of value iteration, try completing the first question of Project 3 at `http://ai.berkeley.edu/reinforcement.html#Q1`. You can download all the necessary files to complete this task.

Hints In order to help you complete the task during the workshop, here are some useful hints:

1. The functions that you need to change:
   (a) _init_
   (b) computeQValueFromValue
   (c) computeActionFromValue
2. The files you need to take a look:
   (a) util.py (Counter())
   (b) mdp.py (isTerminal(), getStates() ,getPossibleActions(), etc.)
   (c) other files: (gridworld.py, learningAgent.py) not directly related
3. How to test your code:
   (a) python autograder.py -q q1 (testing by autograde)
   (b) python gridworld.py -a value -i 5 (result after 5 iteration)
   (c) python gridworld.py -a value -i 100 -k 10 (how value iteration works)