Homework 5

March 1, 2021

In this homework assignment you are going to implement the Bellman equation in the policy evaluation algorithm and develop the test for your function.

Task 1 - Bellman Equation

The formula is

$$V(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$$

You are supposed to complete the **Bellman** function in the *policyEvaluation.py* file. In this function you calculate the value of a single state in a single iteration step.

There are six input parameters of the function **Bellman**.

(1) s - a tuple representing the state you are interested in.

(2) policy - a function that takes a dictionary as input and returns a dictionary. The keys of the input dictionary are actions. The values of the input dictionary are the corresponding expected utilities. The keys of the ouput dictionary are actions. The values of the output dictionary are the probability you choose the actions.

(3) V - a dictionary showing the value of each state. The keys are states. The values are the corresponding values calculated from the last step of iteration.

(4) transitionTable - a dictionary showing the possible results of actions. The keys of this dictionary are all possible current states. The values of this dictionary are action dictionaries whose keys represent all possible actions. The values of the action dictionaries are probability dictionaries whose keys are all possible results from the action given the current state. The values of the probability dictionaries are the probability of the results given the action and the current state.($\{s : \{a : \{s' : P(Result(s, a) = s'|a)\}\}\}$)

(5) getSPrimeRDistribution - a function to get the joint distribution of s' and R. The input arguments of this function are state (s), and action (a). The return value of this function is $p(s', r|s, a)$.

(6) gamma - a scalar γ.

The return value of the function **Bellman** is a scalar. It represents updated value of a **single state** in the current **single step** of iteration.

In the main function, these pieces of information are provided:

(1) transitionTable - $\{s : \{a : \{s' : P(Result(s, a) = s'|a)\}\}\}$. In this assignment we are using a deterministic transitionTable. The environment is a 3×5 grid, from (0,0) to (2,4). The actions are: (1,0) - move right, (0,1) - move down, (-1,0) - move left, (0, -1) - move up. If you take the action of moving to another state, you will have the probability of 1 to get there. If you move to out of the grid, you will be bounced back and stay in the grid you are in right now.

(2) rewardTable - A dictionary showing the reward of each current state, action and next state combination (s, a, s'). The structure is $\{s : \{a : \{s' : reward\}\}\}$. In our grid setting, we have a trap (1,2) and a bonus (2,4). All moves have the cost 1 (reward=-1) except for the moves to the trap and the bonus. Any move whose destination is the trap will have the cost of 1000 (reward=-1000). Any move whose destination is the bonus will have the no cost and a reward of 100 (reward=100).

You can uncomment and run the commented lines in the main function to take a look at the transitionTable and rewardTable.

(3) getSPrimeRDistribution - a function to get the joint distribution of s' and r. The input arguments of this function are state (s), and action (a). The return value of this function is $p(s', r|s, a)$.

(4) policy - In our assignment we are using the ϵ-greedy policy based on the expected utility of each action from last iteration. We can compare when ϵ is small, large or when the policy is completely random.

Task 2 - Unit test

You are supposed to develop a unit test for your **Bellman** function. Create your own *testBellman_YourLastName_YourFirstName.py* file and use at least two sets of data to test your function.

Submission

Please submit a completed *policyEvaluation_YourLastName_YourFirstName.py* file and a *testBellman_YourLastName_YourFirstName.py* on CCLE before due. **Please submit two seperate files. Do not zip them! The due date and time of this homework assignment is Monday, 03/08/2021 11:59pm**.