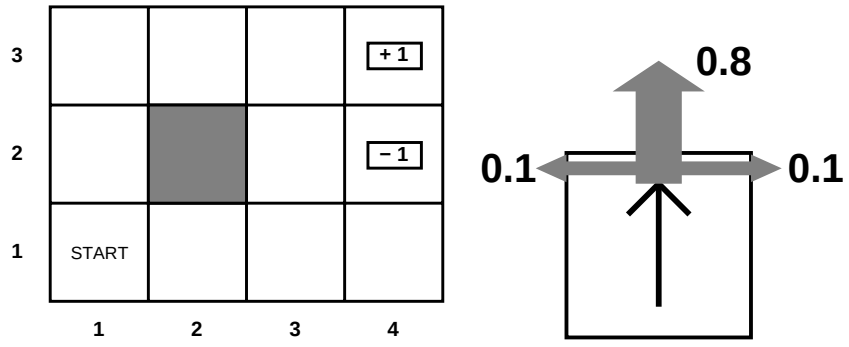
CSCI 5512: Artificial Intelligence II (Fall '19)

Homework 3

(Due Thu, Nov. 14, 11:59 pm central)

1. **(55 points)** [Programming Assignment] Consider the state space and transitions due to actions for a Markov Decision Process (MDP) shown in Figure 1. The goal is to find the optimal policy for a given set of rewards for individual states. Note that the rewards for the terminal states are already given (Figure 1(a)).



(a) States of the MDP (b) Transitions due to action

Figure 1: Markov Decision Process.

For each of the three non-terminal states rewards: (i) $r = -2$, (ii) $r = -0.2$, and $r = -0.01$ do the following:

(a) (20 points) Write a program to compute the optimal policy for the MDP in Figure 1 using **value iteration**.

(b) (25 points) Write a program to compute the optimal policy for the MDP in Figure 1 using **policy iteration**. For solving the linear equation $Ax = b$, you can use standard approaches/code.

For (a) and (b) above, the computed optimal policy and the utility of each state has to be shown in the state space diagram. You have to submit code for mdpVI and mdpPI which each take in one argument: *reward*, which determines the reward for every non-terminal state. The output should be the policy of each state, in separate lines, in the form: *row*, *column*, *policy*, where $row \in \{1, 2, 3\}$, $column \in \{1, 2, 3, 4\}$, and $policy \in \{u, d, l, r\}$, which respectively correspond to up, down, left, and right. Note that the policy has to be stated for 9 non-terminal states, since $(2, 2)$ is not a valid state, and $(4, 2)$ and $(4, 3)$ are terminal states.

Sample input for Python 3.6 for (a) and (b) when $reward = -2$:

```
$python mdpVI.py -2
```

```
$python mdpPI.py -2
```

2. **(45 points)** [Programming Assignment] Consider the Restaurant dataset given in Table 1.

Example	*Attributes*										*Target*
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	WillWait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

   (a) (20 points) Implement a decision tree learning algorithm `dtree4`, and learn a decision tree (of depth at most 4) from the given training data. What is the classification error rate on the training set of the learned decision tree? What is the leave-one-out-cross-validation (LOOCV) classification error rate?

   (b) (20 points) Implement a 2-decision list learning algorithm `dlist2`, and learn a decision list from the given training data. What is the classification error rate on the training set of the learned decision tree? What is the leave-one-out-cross-validation (LOOCV) classification error rate?

3. **Extra Credit (10 points)** Consider the size of the hypothesis space of depth-k decision trees for boolean functions $f(x_1, \ldots, x_n)$ where each input is also boolean. Prove that the hypothesis space of k-decision trees (k-DT(n)) is smaller than the hypothesis space of k-decision lists (k-DL(n)).

Instructions

Please follow these instructions carefully. Code submitted without adhering to these instructions will not receive any credit.

For each programming assignment, you have to submit the code as required by the problem and the algorithm must be implemented using a main file as named in the problem (e.g., `mdpVI.py`). Only Python 3.6 will be accepted, any other language will receive zero credit. The program must run on the CSE labs machines and will not receive credit if it fails this.