CS2180 Artificial Intelligence Lab (Jan-May 2023)

Department of Computer Science and Engineering

Indian Institute of Technology Palakkad

Assignment 2: Agents and Environments (Given: 31 Jan 2023, Due: 14 Feb 2023 2pm)

General instructions

- Solutions are to be typed in the `.ipynb` file provided and uploaded in the lab course page in Moodle before the due date.
- Your code should be well commented and should be compatible with python3.
- For this assignment, you are allowed to import the libraries `pandas`, `numpy`, `random` and `re` of python3. No other libraries may be imported.

1 Text Generation (Static Prediction Task)

(a) Read the text file "speeches.txt" and find the list of unique words with their frequency of occurrence.

(b) Let the number of unique words be n. Construct an $n \times n$ matrix M where the ith row corresponds to the ith word, and the (i, j)th entry stands for the frequency of occurrence of jth word after the ith word. Write a function, which accepts a given word, and returns the frequencies of occurrences of the next words (i.e., the row corresponding to the given word).

(c) Use M to produce a random text T of length 5000 where the first word in T is a random word from the input file and the subsequent words are sampled according to M.

2 Room Cleaning Robot (Dynamic Control Task)

Consider a Room Cleaner Robot which cleans a room containing dirt. A room can be treated as a $m \times n$ grid with walls on all sides.

(a) Implement an environment which takes as inputs the values m and n. Initially, 10 random cells in the grid contain dirt. The list d_t with dimensions $m \times n$ contains the information on the amount of dirt in each cell of the grid. At any point of time, a unit dirt is added at a random location, i.e., a location (x, y) is picked uniformly at random and $d_t(x, y)$ is updated to $d_t(x, y) + 1$.

(b) Implement an agent which at each time instant t observes its position $o_t = (x_t, y_t)$ (but not the dirt information in this location) and performs a random action $a_t \in \{up, down, right, left, pickDirt\}$. On hitting a wall or picking dirt from the current location

(x_t, y_t), the agent stays in the same position (x_t, y_t), otherwise, its position changes according to the action. Every action is associated with a reward (or penalty) defined as follows.

$$r_t = R((x_t, y_t), a_t) = \begin{cases} -1, & \text{if the agent tries to pick dirt and } (x_t, y_t) \text{ is a clean cell} \\ -10, & \text{if the agent hits a wall} \\ d, & \text{if } (x_t, y_t) \text{ has } d \text{ units of dirt and the agent picks it} \\ 0, & \text{otherwise} \end{cases}$$

(c) Print out the activity at each time instant t for $t = 1, \ldots, 1000$. That is, for each t, display the grid (with dirt values in each cell), the location of the agent, the action of the agent and the reward obtained.

3 Cricket Match Simulation (Stochastic Control Task)

Consider the following simplified variant of the first innings of a one-day cricket match. The innings consists of 300 balls and at any point of time, treat the pair of batsmen playing as a single player. That is, there are 10 players: $(1, 2), (2, 3), \ldots, (10, 11)$ with exactly one of them batting. In each ball, there are 5 possible shots $A = \{1, 2, 3, 4, 6\}$ and each of these shots is associated with a risk of the player getting out. This risk varies from player to player. The probabilities of getting out for Player $(1, 2)$ is $pOutMin = [0.01, 0.02, 0.03, 0.1, 0.3]$ and for Player $(10, 11)$ is $pOutMax = [0.1, 0.2, 0.3, 0.5, 0.7]$, where the ith entry is for the ith action. If there are w wickets in hand, then use the formula $pOut(a, w) = pOutMax(a) - (pOutMax(a) - pOutMin(a)) \times (w - 1)/9)$ to calculate the probabilities of getting out for Player $(11 - w, 11 - w + 1)$ where $w \in [10]$. Note that $pOut(a, 1) = pOutMin(a)$ and $pOut(a, 10) = pOutMax(a)$. Taking $pRunMin = 0.5$, and $pRunMax = 0.8$, when the player is not getting out, the probability of successfully obtaining the runs for that shot is given by $pRun(w) = pRunMin + (pRunMax - pRunMin) \times ((w - 1)/9)$.

(a) Implement an environment that maintains the state $s_t = (b_t, w_t)$ where b_t is the balls left and w_t is the wickets left at time t. Initialize the start state to $s_1 = (300, 10)$. Write a function that accepts input as $a_t \in A$ and returns r_t (the runs scored on that shot) and s_t, and also updates s_{t+1}.

(b) Play $a_t = 1$ for each t and find the average balls played and average runs scored by Player $(1, 2)$ and Player $(10, 11)$.

(c) Play $a_t = 6$ for each t and find the average balls played and average runs scored by Player $(1, 2)$ and Player $(10, 11)$.

(d) Simulate 10 matches, for different constant strategies like $a_t = 1$ for each t, $a_t = 2$ for each t, etc. and and find the average runs obtained in each of the strategies.