Homework#4

CS576 Machine Learning, Fall 2023

Due Date: December 8

Instruction

- Compile your work into a single file, naming it "*YourLastName_FirstName*_CS576_HW4.zip"
- Structure your submission with individual directories (or files) for each part, namely Part I, Part II, Part III, etc.
- Label your answer clearly with the corresponding question number such as Part I Q1 (1), Part I Q2, Part II Q3, and so on.

Part I. Fundamental Concepts of Reinforcement Learning

1. Provide your answer for each of the following questions:

(1) Define reinforcement learning. How does it differ from supervised and unsupervised learning?

(2) What is a Markov Decision Process (MDP)?

(3) Explain the Markov property. Why is it important in the context of MDPs?

(4) What is the difference between a value function and a Q-function (or action-value function)?

(5) Describe the exploration-exploitation trade-off in reinforcement learning.

(6) What is the ε-greedy strategy, and how does it balance exploration and exploitation?

2. Describe the pseudocode of **Q-Learning** algorithm

Part II. GridWorld Reinforcement Learning

In part II, you will gain hands-on experience with a basic grid world problem, using a Q-learning agent to navigate through the grid to reach a goal while avoiding obstacles.

3. The following describes the simple grid world reinforcement learning problem. To assist with your implementation, a template file named GridWorldRN_template.ipynb is provided. .
Complete the program file and present the output result.

i. Grid World Environment

- **Grid Dimensions**: Work with a 4x4 grid.
- **Start & Goal**: Set fixed locations for the start (0,0) and goal (3.3)
- **Obstacles**: Set two obstacles in (1,1) and (2.2)
- **Agent Interaction**: The agent can move one cell at a time, either up, down, left, or right.

ii. State and Action Space

- **State Space**: Each cell represents a state, totaling 16 states in a 4x4 grid.
- **Action Space**: Four possible actions - up, down, left, right.

iii. Reward Structure

- **Rewards/Penalties**: A high positive reward (+10) for reaching the goal, a penalty (-10) for hitting an obstacle, and a small penalty (-1) per move to promote efficiency.

iv. Q-Learning Algorithm Setup

- **Learning Rate** (α): Set at α=0.1 to determine how much new information overrides old information.
- **Discount Factor** (γ): Set γ=0.9 to balances immediate and future rewards.
- **ϵ-greedy strategy**: Start with ϵ =1.0, decreasing over time to shift from exploration to exploitation (max_exploration_rate=1.0, min_exploration_rate=0.01 and exploration_decay_rate=0.001).

v. Initialization and Updates

- **Q-table Initialization**: Start with all zeros, indicating no prior knowledge.
- **Q-value Update**: Adjust Q-values after each step based on the Q-Learning algorithm

vi. Training the Agent

- **Training Procedure**: Conduct multiple episodes (10), with max_steps_per_episode parameter (20) to avoid infinite loops.

Deliverables:

(1) The complete program code (named with GridWorldRN_ver1.ipynb)

(2) The program output – Show the grid per each 5 steps and present the Q-table after each episode.

4. For assessing the algorithm performance, we will use three key metrics.

vii. Performance Metrics

- **Metrics used**: Average steps per episode, the success rate (the probability of reaching the goal), and the learning curve (showing improvement across episodes).

To effective measure these metrics, you will need to modify your program to generate three specific pots:

(a) **Steps per Episode**: This plot will illustrate the average number of steps taken in each episode

(b) **Success Rate per Episode**: This plot will display the frequency of successfully reaching the goal in each episode.

(c) **Learning Curve** (**Total Reward per Episode**): This plot will demonstrate the total reward accumulated in each episode, showcasing the agent's learning process over time.

Start by adjusting the number of episodes to 500 and setting the maximum number of steps per episode to 50. This expansion will provide a more comprehensive view of the agent's performance and learning trajectory.

Deliverables:

(1) Modified program code (named with GridWorldRN_ver2.ipynb)

(2) The program output – The three performance plots (a), (b) and (c)

Part III. Clustering

5. The table below presents a small dataset comprising measurements from three sensors. These measurements were recorded when a valve in an oil well was opened, capturing data on PRESSURE, TEMPERATURE, and VOLUME - the characteristics of the oil flow through the valve.

We will apply the ***k*-means** clustering technique to this dataset, setting k=3 and using Euclidean distance as the metric. The initial cluster centroids for the three clusters C1, C2 , and C3 are defined as follows: $c_1 = \langle -0.929, -1.040, -0.831 \rangle$, $c_2 = \langle -0.329, -1.099,\ 0377 \rangle$, $c_3 = \langle -0.672, -0.505, 0.110 \rangle$,

The table also includes distances to these three cluster centers for each data instance.

				Cluster Distances Iter. 1		
ID	PRESSURE	TEMPERATURE	VOLUME	$Dist(\mathbf{d}_i, \mathbf{c}_1)$	$Dist(\mathbf{d}_i, \mathbf{c}_2)$	$Dist(\mathbf{d}_i, \mathbf{c}_3)$
1	-0.392	-1.258	-0.666	0.603	1.057	1.117
2	-0.251	-1.781	-1.495	1.204	1.994	2.093
3	-0.823	-0.042	1.254	2.314	1.460	1.243
4	0.917	-0.961	0.055	2.049	1.294	1.654
5	-0.736	-1.694	-0.686	0.697	1.284	1.432
6	1.204	-0.605	0.351	2.477	1.611	1.894
7	0.778	-0.436	-0.220	1.911	1.422	1.489
8	1.075	-1.199	-0.141	2.125	1.500	1.896
9	-0.854	-0.654	0.771	1.650	0.793	0.702
10	-1.027	-0.269	0.893	1.891	1.201	0.892
11	-0.288	-2.116	-1.165	1.296	1.848	2.090
12	-0.597	-1.577	-0.618	0.666	1.136	1.298
13	-1.113	-0.271	0.930	1.930	1.267	0.960
14	-0.849	-0.430	0.612	1.569	0.879	0.538
15	1.280	-1.188	0.053	2.384	1.644	2.069

(1) Given the initial cluster centroids and the distances provided, identify which cluster each observation belongs to. Justify your answer with the distance values.

(2) After assigning each observation to the nearest cluster based on the provided distances, calculate the new centroids for each cluster. Present the new centroid values.

(3) Evaluate the homogeneity of the clusters formed in the first iteration using (i) the Within-Cluster Sum of Squares and (ii) the average Silhouette Coefficient. Which cluster appears to be the most homogeneous, and why?

(4) Assuming the new centroids have been calculated, describe how you would perform the next iteration of k-means clustering. What changes would you expect in the cluster memberships?

(5) Based on the distances to the centroids, identify any potential outliers in the dataset. Discuss how outliers can influence the formation of clusters and the centroid calculation.