CMPUT 466/566
University of Alberta **Fall 2022**

Assignment 3

Due: December 8, 2022, 11:59 pm on eClass
466: 55 marks; 566: 60 marks

This assignment contains a written portion, and a programming portion. Please hand in a pdf for your written answers, and a zip of your code for the programming portions. Your pdf can contain scans of handwritten answers, so long as they are legible.

Questions

1) [Language Modeling] (466: 15 marks, 566: 20 marks)

You will use the language_modeling.ipynb notebook to complete the following section. Include the answers to all questions in your pdf.

(a) Complete the code in part 1 of the notebook. What accuracy do you obtain?

(b) Use the code in part 2.1 of the notebook to help you answer this question

i. What do you notice about the generated text?

ii. How can this be avoided?

(c) (*CMPUT 566 Students Only*) Implement the Nucleus Sampling method described in section 3.1 of `https://arxiv.org/pdf/1904.09751.pdf`.

You can use any *torch* or *torch.nn* functions

Hint: Use torch.multinomial for sampling

(d) Use the code in part 2.2 of the notebook to help you answer this question

Use the *predict_mask* function and the *[MASK]* token to extract a fact from the language model (similar to the example provided). Include your input and the model's prediction in your pdf report.

(e) Use code code in part 2.3 of the notebook to help you answer this question

i. Find a short piece of text (article, poem, section of a paper, etc.) and get the model to summarize it. Include the summary in your report.

ii. Is the summary accurate? If so, explain why the summary is accurate. If not, explain how the summary could be improved.

2) [Unsupervised Learning] (10 marks)

For this question, please make a copy of this notebook and save it on your Google Drive. The dataset for the first part of this question is linked here. We highly recommend using Google Colaboratory for this question to avoid longer training times and compatibility issues on your local machine. Moreover, to access the datasets in Colab, it is recommended that you link your Google Drive to your notebook.

For submitting the solutions, please download the .ipynb version of your file with the cell outputs by clicking *File* > *Download*. You should also submit a .pdf file of our notebook after running all the cells. Make sure the pdf has outputs of all the cells including your answers to theoretical questions. To download the notebook as a pdf go to *File* > *Print*. Please keep the same name for your notebook and pdf file, for example Unsupervised.ipynb and Unsupervised.pdf.

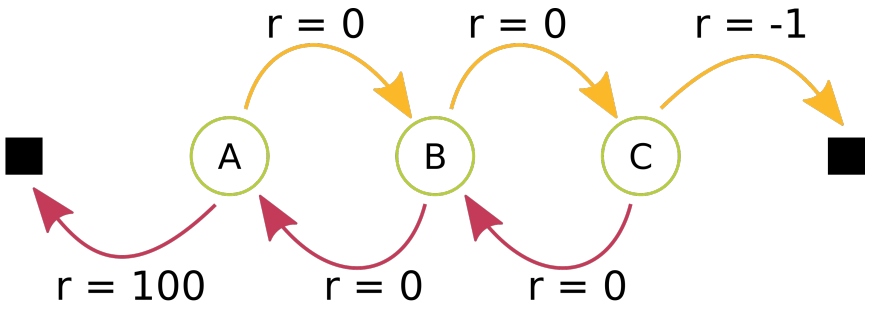
You cannot use any libraries other than the ones already imported in the notebook. You cannot use code completion tools like GitHub Copilot.

3) [Reinforcement Learning] (10 marks)

(a) Given the deterministic Markov Decision Process in Figure 1, compute the value function of the states A, B, and C using a discount factor of 1. Remember that the value function for a state s is defined as

$$v_\pi(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s') \right],$$

where $\pi(a|s)$ is the probability of choosing action a in state s, $p(s', r|s, a)$ is the probability of observing state s' and reward r after executing action a in state s, $\gamma \in [0, 1]$ is a discount factor, and $v_\pi(s')$ is the value function of s'. **Hint 1:** Use the fact that the MDP is deterministic to get rid of one of the summations in the equation above. **Hint 2:** The value function of terminal states is always zero.



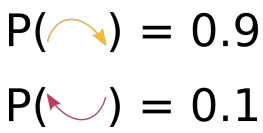
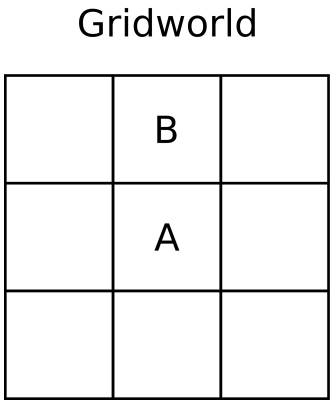


Figure 1: MDP for Question 3)a. The green circles labeled A, B, and C represent states of the environment. The solid black squares represent terminal states. There are two actions move right (yellow arrows) and move left (red arrows). The probability of each action is 0.9 for moving right and 0.1 for moving left. The reward for executing each action in each state is written next to each arrow.



State	Approximate Action Values Q(s,a)			
	Up	Down	Left	Right
A	-1.2	-2.5	2	0.5
B	-0.2	0.1	-1.1	1.2

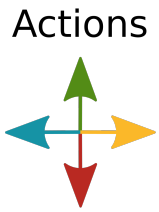


Figure 2: MDP for Questions **3**)b, **3**)c, and **3**)d. Each cell in the gridworld shown on the left is a different state. The actions available to the agent are moving up, down, left, or right. On every transition, the agent receives a reward of -1. The current approximate action-values for states A and B are shown in the table on the right side.

(b) Assume the agent is choosing actions using an ϵ-greedy policy with respect to the action-value function with $\epsilon = 0.25$. Make a table with the probability of choosing each action for states A and B in the gridworld shown in Figure 2.

(c) **[CMPUT 466]** Assume the agent moves from state A to state B in the gridworld shown in Figure 2 and receives a reward of -1. Once in state B, the agent chooses to move right using an ϵ-greedy policy, with $\epsilon = 0.25$. What is the new value of $Q(A, \text{up})$ if we use the Sarsa update rule shown below?

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

Use a discount factor of 0.99 and $\alpha = 0.1$.

(d) **[CMPUT 566]** Assume the agent moves from state A to state B in the gridworld shown in Figure 2 and receives a reward of -1. Once in state B, the agent chooses to move right using an ϵ-greedy policy, with $\epsilon = 0.25$. What is the new value of $Q(A, \text{up})$ if we use the Expected Sarsa update rule shown below?

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \left(\sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) \right) - Q(S_t, A_t) \right]$$

Use a discount factor of 0.99 and $\alpha = 0.1$.

4) [Reinforcement Learning Coding Problem] (15 marks)

Make a copy and follow the instructions in this notebook notebook to complete this portion of the assignment. When submitting your assignment, also include the `ipynb` file generated by Google Colab. **You cannot use any libraries other than *numpy* and *matplotlib*. You cannot use code completion tools like *GitHub Copilot*.**

4.1 Complete the implementation of the `update` method in the `QLearningAgent` class.

4.2 Complete the implementation of the `greedy_policy` function.

4.3 Using the Q-Learning agent and the greedy policy implemented in questions 4.1 and 4.2, respectively, find a learning rate that achieves a higher average reward per episode than the default learning rate of 0.1. Report the value of the learning rate, its corresponding average reward per episode, and the resulting learning curve.

4.4 In the Hard Gridworld environment, try different values for the learning rate parameter until you find a value that achieves a higher average reward per episode than the default learning rate value of 0.1. Report the learning rate and the corresponding average reward per step. Did the agent reach the blue square?

4.5 Complete the implementation of the `epsilon_greedy_policy`.

4.6 In the Hard Gridworld environment, find a value of $\epsilon \in [0,1]$ that achieves an average reward per episode greater than 0. Report the value of ϵ and the corresponding average reward per episode.

5) **[Ethical considerations] (5 points)** Recall in class we discussed possible sources of harm in ML algorithms. These harms are summarized in this paper: https://arxiv.org/abs/1901.10002. For this question, consider the problem of predicive policing: https://en.wikipedia.org/wiki/Predictive_policing.

(a) Give an example of historical bias in predictive policing. Explain your answer.

(b) Give an example of measurement bias in predictive policing. Explain your answer.