QUIZ #3 – COMP 3106 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

NOVEMBER 29, 2023
8:35AM – 9:55AM (80 MINUTES)
SOUTHAM HALL 416

Instructions

The quiz is open-book, and you may consult your notes and the textbook during quizzes. You may not use electronic devices (except non-programmable scientific calculators) during quizzes.

You must complete the quiz individually.

The quiz includes five multiple-choice questions and four written answer questions. Multiple-choice questions are worth 2 marks each. Written answer questions are worth 10 marks each.

For the multiple-choice questions, circle the correct answer. Justification of your answers is not required and will not be considered.

For the written answer questions, write your answer in the space provided below the question. You are not required to provide justification (unless specifically requested). However, if your final answer is incorrect, partial credit may be awarded for justification. If your final answer is incorrect and you do not provide justification, you may receive a grade of zero for that question.

Name:
Student ID: Quiz #3 – COMP 3106

Question 1 [2 marks]

What is the primary objective of reinforcement learning?

a) Learn actions that maximize long-term reward
b) Learn actions that maximize immediate reward
c) Learn structures in data
d) Learn to approximate a function
e) Learn actions to reach a terminal state

Question 2 [2 marks]

Consider an active reinforcement learning agent working in an environment using adaptive dynamic programming with exploration and exploitation. Suppose we encounter a state, call it s, in which the agent could take actions a_1, a_2, a_3, a_4. Suppose the actions a_1, a_2 have both been taken in the state s at least N_E times already ($N_E > 1$). Suppose the actions a_3, a_4 have not yet been taken in the state s. What action should the agent take in the state s?

a) $\underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)V(s')$
b) a_1 with probability 1/2; a_2 with probability 1/2
c) a_3 with probability 1/2; a_4 with probability 1/2
d) a_1 with probability 1/4; a_2 with probability 1/4; a_3 with probability 1/4; a_4 with probability 1/4
e) a_1 with probability $1/N_E$; a_2 with probability $1/N_E$; a_3 with probability $1/N_E$; a_4 with probability $1/N_E$

Question 3 [2 marks]

Consider a reinforcement learning-based vacuum cleaner agent to clean a very large grid environment (e.g. 100 by 100 grid) using a large state space approximation for the value function. Each square in the environment can be dirty or clean. The environment reaches a terminal state T when all squares in the grid are clean. In this environment, each non-terminal state has a reward -0.1; the terminal state has reward +100. Which of the following is a reasonable basis function for this large state space approximation?

a) $f(s) = max\big(V(s)\big)$
b) $f(s) = +100$
c) $f(s) = -1$
d) $f(s) = \#\ of\ squares\ in\ the\ grid$
e) $f(s) = \#\ of\ dirty\ squares\ in\ the\ state\ s$

Question 4 [2 marks]

Under direct policy search, which of the following are computed?

a) Q-function
b) Value function
c) Q-function and Value function
d) Q-function or Value function
e) Neither Q-function nor Value function

Question 5 [2 marks]

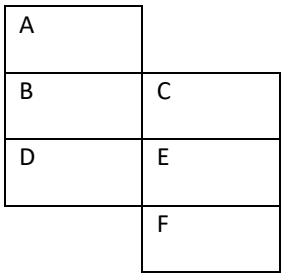
Consider the training sentence “otolaryngology” (it is fourteen letters long). Using a 1-gram model, compute the probability of the sequence of characters “lo”.

a) 0.0000
b) 0.0051
c) 0.0408
d) 0.0714
e) 0.1429

Name:
Student ID: Quiz #3 – COMP 3106

Question 6 [10 marks]

Imagine an agent operating in the grid world illustrated below. The start state is A and the goal state is F.



Suppose at each state, the agent can take one of the following actions:
Move to a horizontally or vertically adjacent square.

Note that with some small random chance, taking an action will not change the state.

The reward associated with each state is:

$$r(s) = \begin{cases} 0 \; if \; s = A \\ -1 \; if \; s = B \; or \; s = C \\ -2 \; if \; s = D \; or \; s = E \\ +10 \; if \; s = F \end{cases}$$

Suppose we simulate one trajectory through the state space, with the following state-action pairs:
(A, Down) -> (B, Down) -> (B, Right) -> (C, Down) -> (E, Down) -> (F, None)

Estimate the value function V(s) associated with each state under fixed policy using one of the three methods listed below (your choice, but you must indicate which method you choose).

1. Direct estimation
2. Adaptive Dynamic Programming (perform one iteration)
3. Temporal Difference Learning (perform one iteration)

If necessary, use the following:
Discount factor: $\gamma = 0.5$
Learning rate: $\alpha = 0.5$
Initial estimate for the value function: $V_{initial}(s) = r(s)$

Direct estimation:

$$V(A) = r(A) + \gamma r(B) + \gamma^2 r(B) + \gamma^3 r(C) + \gamma^4 r(E) + \gamma^5 r(F)$$
$$V(A) = -0.6875$$

$$V(B) = \frac{[r(B) + \gamma r(B) + \gamma^2 r(C) + \gamma^3 r(E) + \gamma^4 r(F)] + [r(B) + \gamma r(C) + \gamma^2 r(E) + \gamma^3 r(F)]}{2}$$
$$V(B) = -1.0625$$

$$V(C) = -1 + \gamma r(E) + \gamma^2 r(F)$$
$$V(C) = 0.5$$

$$V(E) = r(E) + \gamma r(F)$$
$$V(E) = 3$$

$$V(F) = r(F)$$

$$V(F) = 10$$

All other values remain the same as the initial estimate.

Adaptive Dynamic Programming:

Transition probabilities:

$$P(B|B,\pi) = 0.5$$
$$P(C|B,\pi) = 0.5$$

Observe that all transition probabilities are zero or one because no state-action pair is encountered multiple times.

$$V(A) = r(A) + \gamma \sum P(s'|A,\pi)V(s')$$
$$V(A) = r(A) + \gamma(1 \times V(B))$$
$$V(A) = -0.5$$

$$V(B) = r(B) + \gamma \sum P(s'|B,\pi)V(s')$$
$$V(B) = r(B) + \gamma\big(0.5 \times V(B) + 0.5 \times V(C)\big)$$
$$V(B) = -1.5$$

$$V(C) = r(C) + \gamma \sum P(s'|A,\pi)V(s')$$
$$V(C) = r(C) + \gamma(1 \times V(E))$$
$$V(C) = -2$$

$$V(E) = r(E) + \gamma \sum P(s'|A,\pi)V(s')$$
$$V(E) = r(E) + \gamma(1 \times V(F))$$
$$V(E) = 3$$

All other values remain the same as the initial estimate.

Temporal Difference Learning:

$$V(A) = V(A) + \alpha(r(A) + \gamma V(B) - V(A))$$
$$V(A) = -0.25$$

$$V(B) = V(B) + \alpha(r(B) + \gamma V(B) - V(B))$$
$$V(B) = -1.25$$

$$V(B) = V(B) + \alpha(r(B) + \gamma V(C) - V(B))$$
$$V(B) = -1.375$$

$$V(C) = V(C) + \alpha(r(C) + \gamma V(E) - V(C))$$
$$V(C) = -1.5$$

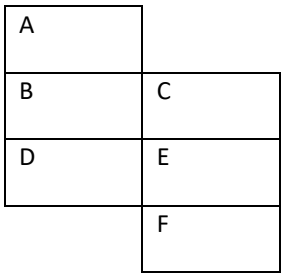
$$V(E) = V(E) + \alpha(r(E) + \gamma V(F) - V(E))$$
$$V(E) = 0.5$$

All other values remain the same as the initial estimate.

Name:
Student ID: Quiz #3 – COMP 3106

Question 7 [10 marks]

Imagine an agent operating in the grid world illustrated below. The start state is A and the goal state is F.



Suppose at each state, the agent can take one of the following actions:
Move to a horizontally or vertically adjacent square.

Note that with some small random chance, taking an action will not change the state.

The reward associated with each state is:

$$r(s) = \begin{cases} 0 \; if \; s = A \\ -1 \; if \; s = B \; or \; s = C \\ -2 \; if \; s = D \; or \; s = E \\ +10 \; if \; s = F \end{cases}$$

Suppose we simulate one trajectory through the state space, with the following state-action pairs:
(A, Down) -> (B, Down) -> (B, Right) -> (C, Down) -> (E, Down) -> (F, None)

Estimate the Q-value function Q(s, a) associated with each state-action pair using temporal difference Q-learning. Perform one iteration.

If necessary, use the following:
Discount factor: $\gamma = 0.5$
Learning rate: $\alpha = 0.5$
Initial estimate for the value function: $Q_{initial}(s, a) = r(s)$

$$Q(A, Down) = Q(A, Down) + \alpha(r(A) + \gamma \max(Q(B, a')) - Q(A, Down))$$
$$Q(A, Down) = -0.25$$

$$Q(B, Down) = Q(B, Down) + \alpha(r(B) + \gamma \max(Q(B, a')) - Q(B, Down))$$
$$Q(B, Down) = -1.25$$

$$Q(B, Right) = Q(B, Right) + \alpha(r(B) + \gamma \max(Q(C, a')) - Q(B, Right))$$
$$Q(B, Right) = -1.25$$

$$Q(C, Down) = Q(C, Down) + \alpha(r(C) + \gamma \max(Q(E, a')) - Q(C, Down))$$
$$Q(C, Down) = -1.5$$

$$Q(E, Down) = Q(E, Down) + \alpha(r(E) + \gamma \max(Q(F, a')) - Q(E, Down))$$
$$Q(E, Down) = 0.5$$

All other Q-values remain the same as the initial estimate.

Name:
Student ID: Quiz #3 – COMP 3106

Question 8 [10 marks]

Suppose we have a set S of n positive real numbers $\{x_1, \dots, x_n\}$. We wish to find two non-empty subsets A, B of S such that the sum of the elements in A is as close as possible to the sum of the elements in B. Note that there may be some elements of S that are in neither A nor B.

Consider using a genetic algorithm to solve the problem of computing the optimal solution to this problem. State a representation, fitness function, genetic operator, and mutation operator.

You do not have to execute the genetic algorithm.

Represent a solution as a bit string $y = b_1, \dots, b_n$.

$$b_i = \begin{cases} 1 \; if \; x_i \in A \\ -1 \; if \; x_i \in B \\ 0 \; otherwise \end{cases}$$

Fitness function:

$$f(y) = \frac{1}{\left|\sum_{i=1}^{n}(x_i \times y_i)\right|}$$

Genetic operator:

$$g(y, z) = y_1, \dots, y_i, z_{i+1}, \dots, z_n$$
For random $0 \leq i \leq n$ such that $1 \in g(y, z)$ and $-1 \in g(y, z)$

Mutation operator:

$$m(y) = y_1, \dots, y_{j-1}, w, y_{j+1}, \dots, y_n$$
For random $1 \leq j \leq n$ and random $w \in \{-1, 0, 1\}$ such that $1 \in m(y)$ and $-1 \in m(y)$

Name:
Student ID: Quiz #3 – COMP 3106

Question 9 [10 marks]

Consider the follow set of documents:

Document 1: “the cat takes the mouse”
Document 2: “the mouse takes the cheese”
Document 3: “the cheese stands alone”

Using a bag of words model, compute the term frequency-inverse document frequency vectors for each of these documents. In your answer, explicitly state your vocabulary.

Vocabulary:
['the' 'cat' 'takes' 'mouse' 'cheese' 'stands' 'alone']

Occurrence vectors:
Document 1: [1, 1, 1, 1, 0, 0, 0]
Document 2: [1, 0, 1, 1, 1, 0, 0]
Document 3: [1, 0, 0, 0, 1, 1, 1]

Count vectors:
Document 1: [2, 1, 1, 1, 0, 0, 0]
Document 2: [2, 0, 1, 1, 1, 0, 0]
Document 3: [1, 0, 0, 0, 1, 1, 1]

Term frequency vectors:
Document 1: [0.4, 0.2, 0.2, 0.2, 0.0, 0.0, 0.0]
Document 2: [0.4, 0.0, 0.2, 0.2, 0.2, 0.0, 0.0]
Document 3: [0.25, 0.0, 0.0, 0.0, 0.25, 0.25, 0.25]

Inverse document frequency vector (using log2):
[0.0, 1.584962500721156, 0.5849625007211562, 0.5849625007211562, 0.5849625007211562, 1.584962500721156, 1.584962500721156]

Term frequency-inverse document frequency vectors (using log2):
Document 1: [0.0, 0.31699250014423125, 0.11699250014423124, 0.11699250014423124, 0.0, 0.0, 0.0]
Document 2: [0.0, 0.0, 0.11699250014423124, 0.11699250014423124, 0.11699250014423124, 0.0, 0.0]
Document 3: [0.0, 0.0, 0.0, 0.0, 0.14624062518028905, 0.396240625180289, 0.396240625180289]

Inverse document frequency vector (using log10):
[0.0, 0.47712125471966244, 0.17609125905568124, 0.17609125905568124, 0.17609125905568124, 0.47712125471966244, 0.47712125471966244]

Term frequency-inverse document frequency vectors (using log2):
Document 1: [0.0, 0.09542425094393249, 0.03521825181113625, 0.03521825181113625, 0.0, 0.0, 0.0]
Document 2: [0.0, 0.0, 0.03521825181113625, 0.03521825181113625, 0.03521825181113625, 0.0, 0.0]
Document 3: [0.0, 0.0, 0.0, 0.0, 0.04402281476392031, 0.11928031367991561, 0.11928031367991561]

Inverse document frequency vector (using ln):

[0.0, 1.0986122886681098, 0.4054651081081644, 0.4054651081081644, 0.4054651081081644, 1.0986122886681098, 1.0986122886681098]

Term frequency-inverse document frequency vectors (using ln):
Document 1: [0.0, 0.21972245773362198, 0.08109302162163289, 0.08109302162163289, 0.0, 0.0, 0.0]
Document 2: [0.0, 0.0, 0.08109302162163289, 0.08109302162163289, 0.08109302162163289, 0.0, 0.0]
Document 3: [0.0, 0.0, 0.0, 0.0, 0.1013662770270411, 0.27465307216702745, 0.27465307216702745]