CSE 6th Semester

CONTINUOUS ASSESSMENT TEST - 3 Date: 15th Apr 2024

[19Ze601] - Machine Learning

Time: 1 Hour 30 minutes. Maximum Marks: 35

INSTRUCTIONS:

1. **Answer ALL questions. Each Question carries 25 Marks.**
2. Question No. 1 carries 8 Marks and question No. 2 carries 27 Marks
3. In question No. 1, subdivision **a** carries total of 8 marks (one mark for each question).
4. In question No. 2, subdivision **a** carries total of 7 marks (one mark for each question), subdivisions **b(i)** and **b(ii)** carries 5 marks each and subdivision **c** carries 10 marks.
5. Course Outcome Table : Q1: CO 1-4 and Q2: CO 5

1. **a .** **(8x 1 mark = 8 marks)**

Write the alphabet of your choice answer in the CA test answer book mentioning question number and subdivision number.

i. Inductive learning algorithm that uses both positive and negative examples is / are
A) Find S B) Candidate elimination algorithm C) Both of above D) None of the above

ii. The Delta rule is used to correct
A) Weight values B) Activation function C) Threshold values D) None of the above

iii. 'Bias' and 'Variance' are
A) the same B) opposite C) Unconnected D) synonyms

iv. A discriminant technique that maximizes separation between classes and compacts a class is
A) Least squares B) Perceptron C) Fishers Linear discriminant D) All of the above

v. Naive Bayes requires independence between _____

vi. Describe and differentiate the use of Entropy and the use of Information Gain.

vii. Regularization in Neural Networks is used to handle _____

viii. Entropy is used for measuring _____ while Information Gain is used for measuring _____

2. **a** **(7 x 1 mark = 7 marks)**

Write the alphabet of your choice answer in the CA test answer book mentioning question number and subdivision number.

i. Two major activities performed in Reinforcement learning are
A) Deduce and Infer B) Exploit and Reuse C) Explore and Exploit D) Explore and Deduce

ii. In Reinforcement learning, 'Rewards' are discounted based on
A) Defect in task B) Time of occurrence C) Effort of the task D) Frequency of occurrence

iii. Reinforcement learning differs from other learning because, in Reinforcement learning because

A) Both 'policy' and 'value' have to be learned B) Feedback is delayed C) Actions are non-deterministic D) All of the above

iv. Temporal difference learning solution is characterized by
A) Temporary values B) difference between results C) n step time difference D) All of the above

Write the answer for the following. Fill in the blanks questions in the CA test answer book mentioning question number and subdivision number.

v. Markov Decision process assumes that current state represents all _____ states

vi. Reinforcement learning learns _____ strategies for autonomous agents

vii. _____ assignment is the major problem for reinforcement learning situations

b. **(2 x 5 marks = 10 marks)**

i. Describe the Markov decision process used in Reinforcement learning

ii. Differentiate the Value Function and the Policy in the context of Reinforcement learning. Describe how the Value function can be learned even though the there are no training examples of the form $< s, a >$

c. ***(1 x 10 marks = 10 marks)***

i. Describe the algorithm for Q -learning. How can Q-learning be used for non-deterministic worlds. Show convergence for Q- learning in a deterministic world.. Demonstrate with an example

(OR)

ii. Describe the algorithm for Temporal Difference Learning. Demonstrate with an example

b ii)

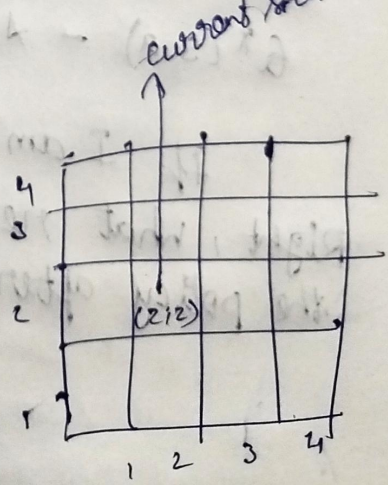
Aspect	Policy (π)	Value Function (V or Q)
Purpose	Tells the agent what to do.	Tells the agent how good a state is
Input	current state	State is a (state, action).
Output	Action (or probability of each action).	A number: expected future reward.
Focus	Behaviour (decision making)	Evaluation (how good is a choice)

These value function ($V(s)$ and $Q(s,a)$ will give you values at each state, By using that we can obtain a best policy π.

Example:

Policy (π)

This is my strategy - how I decide what move to make in each state.



In state $S = (2,2)$

the policy might be :-

$\pi(a|s) = \begin{bmatrix} \text{right} : 0.6, \\ \text{up} : 0.4, \\ \text{left/down} : 0.0 \end{bmatrix}$

This means:

"If I am at (2,2), I will go right 60% of the time and up 40% of the time."

The best way is maximum probability.

Through policy we can make decision.

Through Value Function - we can have experience.

Value Function (V and Q):

$V^{\pi}(s)$ - state-value Function.

"If I am in state (2,2), and I follow my current strategy (policy), how much reward can I expect in the future?"

$Q^{\pi}(s,a)$ - Action-value Function

"If I am in state (2,2), and I move Right, what reward can I expect if I follow the policy afterward".

How can value Function be Learned without Training Examples $\langle s, a \rangle$?

Trick!! Learn through Interaction (Experience), Not labels!

RL - you we often don't have labeled data. Instead, the agent learns by doing thing and seeing what happens.

Solution: Temporal Difference Learning.

$$V(s) \leftarrow V(s) + \alpha[r + \gamma(V(s') - V(s))]$$

This means:

You update your guess of $V(s)$ based on the actual reward and the estimate of value of the next state $V(s')$.

- This is called bootstrapping - learning from your own estimates.

6))

Q-learning Algorithm:

For each s,a initialize table entry $\hat{Q}(s,a) \leftarrow 0$, Observe current state s

Do forever:

→ select an action a and execute it
→ receive immediate reward r
→ Observe the new state s'
→ Update the table entry for $\hat{Q}(s,a)$ as follows:

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$$

→ $s \leftarrow s'$

How Q-learning Handles Non-Determinism?

In non-deterministic environments, the next state and reward are not fixed for a given (s,a) pair.

Q-learning still works because:

* It uses samples of transitions rather than modeling the environment.

* It uses expected values through repeated sampling, which allows it to converge to the optimal Q-values even when transitions are stochastic.

so as long as:

* All state-action pairs are visited infinitely often,
* The learning rate α decays properly,
* And $\gamma < 1$.

Q-learning converges with probability 1 to the optimal Q-function $Q^*(s,a)$ - even in a non-deterministic world! (Refer pdf for formula).

$V^\pi(s) \equiv E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right]$, $Q(s,a) \equiv E[r(s,a) + \gamma V^*(s(s,a))]$.

Convergence of Q-learning in Deterministic world.

In a deterministic world, each action leads to a known state and reward - This simplifies convergence because there's no randomness - the update is exact each time!

Why it converges?

* Repeated updates reinforce the correct Q-values.
* The $\max_{a'} Q(s',a')$ term always reflects the true best path, since future values don't fluctuate due to randomness.

S3	S4
S2	S1

$$R = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$Q = \begin{array}{c} \\ S1 \\ S2 \\ S3 \\ S4 \end{array} \begin{array}{c} \begin{array}{cccc} UP & down & left & right \end{array} \\ \begin{bmatrix} 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 \\ -1 & 0 & -1 & 0 \\ -1 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

Let us start with S3:

$$Q(S3, RIGHT) = R(S3, RIGHT) + \gamma \max_{a'} Q(S4, R\,a')$$

$$= 1 + 0.8 \max\{0, 0\}$$

$$= 1 + 0$$

$$Q(S3, RIGHT) = 1$$

S2

$$Q(S2, UP) = R(S2, UP) + \gamma \max_{a'} Q(S3, a')$$

$$= 0 + 0.8 \max\{1, 0\}$$

$$= 0.8.$$

Like this go on.

① - Episode

→ $Q =$

$$\begin{bmatrix} 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 \\ -1 & 0 & -1 & 1 \\ -1 & 0 & 0 & -1 \end{bmatrix}$$

Next

$$Q = \begin{bmatrix} 0 & -1 & 0 & -1 \\ 0.8 & -1 & -1 & 0 \\ -1 & 0 & -1 & 1 \\ -1 & 0 & 0 & -1 \end{bmatrix}$$

Convergence (key points)

Non-deterministic

→ Use Expected values of $V\pi(s)$ and $Q^{\pi}(s,a)$ and Convergence.

Deterministic

→ No problem, directly Q values will converge after many iterations

Convergence mean: To put a best value inside the Q-table for each state. This value will decrease the error to optimal Q-value and try to give good policy.