USN | | | | | | | | | | |

Internal Assessment Test 3 – Jan 2024

Sub:	Artificial Intelligence and Machine Learning					Sub Code:	18CS71	Branch :	CSE
Date:	2.01.2024	Duration:	90 mins	Max Marks:	50	Sem/Sec :	7 A		OBE

Answer any FIVE FULL Questions		MARKS	CO	RBT
1 (a)	Explain what is Bayesian Belief Networks and the following terms i)joint space ii) conditional Independence iii)Joint probability.	5M	CO1	L2

- describes the probability distribution governing a set of variables specifying a set of conditional independence assumptions with a a set of conditional probabilities.
- allow stating conditional independence assumptions that apply to subsets of the variables.
- provide a less contraining approach than the global assumption of conditional independence made by NB classifier.
- in general, Bayesian belief network describes probability distribution over a set of variables.

• joint space: set of variables Y which is the cross product $V(Y_1) \times V(Y_2) \times \dots V(Y_n)$.

- probability distribution over this joint space: joint probability distribution.

• Conditional Independence

X is conditionally independent of Y given Z if the probability distribution governing X is the independent of the value of Y given a value for Z.

$$\forall (x_i, y_j, z_k)\ P(X=x_i \mid Y=y_j, Z=z_k) = P(X=x_i \mid Z=z_k)$$

where $x_i \in V(X)$, $y_j \in V(Y)$ and $z_k \in V(Z)$

$$P(X \mid Y, Z) = P(X \mid Z).$$

this can be entended over a set of variables also.

$$P(X_1 \dots X_l \mid Y_1 \dots Y_m, Z_1 \dots Z_n) = P(X_1 \dots X_l \mid Z_1 \dots Z_n)$$

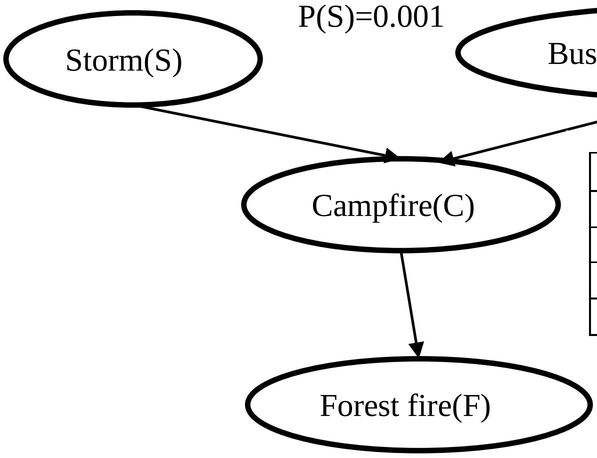
Naive Bayes classifier assumes that instance attribute A_1 is conditionally innendent of A_2 given target value V.

$$P(A_1, A_2 \mid V) = P(A_1 \mid A_2, V)\, P(A_2 \mid V)$$
$$= P(A_1 \mid V)\, P(A_2 \mid V)$$

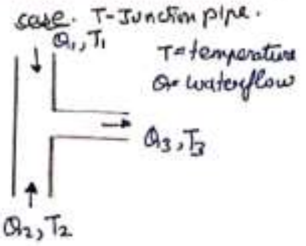
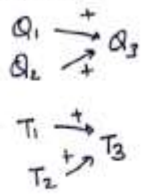
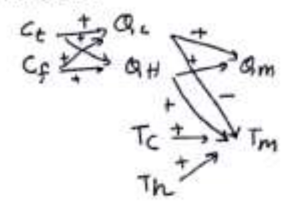
general form of product rule.

- joint probability for any assignment of values $\langle y_1, \dots y_n \rangle$ to $\langle Y_1, \dots Y_n \rangle$

$$P(y_1, \dots, y_n) = \prod_{i=1}^{n} P(y_i \mid Parents(Y_i))$$

(b)	 S | B | P(C|S,B) | $P(\bar{C}\|S,B)$ T | T | 0.3 | T | F | 0.2 | F | T | 0.6 | F | F | 0.4 | C | P(F|C) T | 0.1 F | 0.99 i) Fill up table for $P(\bar{C}\|S,B)$, $P(\bar{S}), P(\bar{B})$ **0.7,.8,.4,.6,** $P(\bar{S}) = 0.999, P(\bar{B}) = 0.95$ ii) What is the probability that there is a campfire but there is no storm and no bus tour group?0.4 iii) Calculate the probability that a Forest fire occurs. Show steps 0.6253  P(F) Prob that camp fire occurs = 0.3 × 0.001 × 0.005 + 0.2 × 0.001 × 0.95 + 0.6 × 0.999 × 0.05 + 0.4 × 0.999 × 0.95 = 0.000015 + 0.00019 + 0.02994 + 0.37962 = 0.4098 Prob that campfire does not occur 1 - 0.4098 = 0.5902 P(F) = 0.1 × 0.4098 + 0.99 × 0.5902 = 0.04098 + 0.584298 = 0.62528	**5M**	**CO2**	L3
2 (a)	Explain K-Nearest Neighbor algorithm for approximating discrete valued target function and continuous valued target function	**8M**	**CO2**	L2

	• K-Nearest Neighbor Learning - assumes all instances correspond to points in the n-dimensional space $\mathbb{R}^n$. - nearest neighbors are defined in terms of standard Euclidean distance. - Let x be described by a feature vector, $\langle a_1(x), a_2(x), \ldots, a_n(x) \rangle$ where $a_r(x)$ denotes the r^{th} attribute of instance x. - The distance between the two instances are: $d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$ - target function may be discrete-valued or real-valued. k-nearest neighbor for approximating discrete-valued function $f: \mathbb{R}^n \to V$ where V is a finite set $\{v_1, \ldots v_s\}$ ▸ Training Algorithm. • For each training example, $(x, f(x))$, add the example to the list training-examples. ▸ Classification Algorithm: • Given a query instance x_q to be classified, → Let $x_1 \ldots x_k$ denote the k instances from training-examples that are nearest to x_q. • Return $\hat{f}(x_q) \leftarrow \text{argmax} \sum_{i=1}^{k} \delta(v, f(x_i))$ where $\delta(a,b)=1$ if $a=b$ and where $\delta(a,b)=0$ otherwise. • Approximating Continuous-valued target function. - calculate the mean value of k nearest training examples. - for $f: \mathbb{R}^n \to \mathbb{R}$, $\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$			
(b)	i) How is distance usually weighted in weighted-KNN $w_i \equiv \frac{1}{d(x_q, x_i)^2}$. Inverse of the square of distance ii) if distance weighting is added, all examples can be used for classification. Why? Instance that are far will have less weightage.	**2M**	**CO3**	L3
3 (a)	How is case-based reasoning similar and different from KNN&LWR.	**3M**	**CO3**	L3

	KNN & LWR. ① lazy learning ② classify new instances by analyzing similar instances ③ instances are represented by real-valued points in n-dimensional Euclidean space. Case-based Reasoning (CBR) - first 2 principles. Applications - conceptual design of mechanical devices based on previous designs. - reasoning about new legal cases based on previous rulings. - solving planning & scheduling problems by reusing / combining portions of previous solutions			
(b)	Explain CADET system with a an example. • The CADET System - uses CBR for conceptual design of simple mechanical devices such as water faucets. - library: 75 previous designs, - each instance - structure and its qualitative function. New desired function - requested. Stored case. T-Junction pipe. Structure:  Function:  Problem Specification Structure: ? Function:  	**7M**	**CO3**	L2

	+ indicates variable at arrowhead increases with variable at tail. e.g. Q_3 increases with Q_1. (quantity) – indicates variable at arrowhead decreases with variable at tail. New design problem Q_c: cold water T_c : temperature of cold water Q_h: hot water T_h: " " hot water T_m: " " mixed " C_t: control signal for temperature C_f: control signal for quantity / waterflow. → for new specification, CADET searches library for stored cases. – if exact match is found, it returns suggested solution – otherwise, it searches for match in various sub graphs. ↳ it searches for subgraph isomorphisms between two function graphs. Isomorphic graphs if $G_1 \equiv G_2$, then – $\|V(G_1)\| = \|V(G_2)\|$ $\|E(G_1)\| = \|E(G_2)\|$. degree sequences are the same. - system may also elaborate the original function specification graph to create functionally equivalent graphs. $A \xrightarrow{+} B$ rewrite rule as $A \xrightarrow{+} x \xrightarrow{+} B$ - if B must increase with A, it is sufficient to find some quantity x such that B increases with x and x increases with A. - x is a universally quantified variable whose value is bound when matching the function graph against the case library.			
4 (a)	Mention the type of learning most apt for the following statements/techniques (eager learning / lazy learning) i) Approximation function is chosen before query is observed-eager ii) Implicit local functions for each query instance-lazy iii) Requires more computation time during training - eager iv) Requires less computation time during prediction - eager v) KNN-lazy vi) LWR-lazy vii) Backpropation-eager	**3M**	**CO1**	L3
(b)	How does Radial Basis Function combine both local and eager methods? Draw RBF networks and explain 2-stage process of training.	**7M**	**CO2**	L2

Radial Basis Functions

$$\hat{f}(x) = w_0 + \sum_{u=1}^{k} w_u K_u(d(x_u, x)). \quad \text{—①}$$

x_u : an instance from X

$K_u(d(x_u, x))$: kernel function such that it decreases as the distance $d(x_u, x)$ increases.

k: user provided constant that specifies the number of kernel functions to be included.

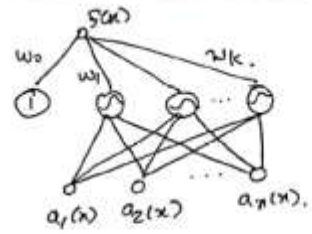
- even though $\hat{f}(x)$ is global approximation to $f(x)$ contribution from each of the $K_u(d(x_u, x))$ is localized to a region nearby point x_u.

* common to choose $K_u(d(x_u, x))$ to be a Gaussian function centered at x_u with some variance σ_u^2.

$$K_u(d(x_u, x)) = e^{\frac{1}{2\sigma_u^2}(x_u, x)}$$

$\hat{f}(x)$ - 2-layer network.

- first layer: computes values of the various $K_u(d(x_u, x))$.
- second layer: computes linear combination of these first layer values.



- each hidden unit produces an activation determined by Gaussian function centered at some instance x_u.
- activation will be close to zero unless input x is near x_u.
- output unit: produces a linear combination of the hidden unit activations.

• RBF networks.

- trained in 2-stage process.
- first the number of k hidden units is determined
- each hidden unit u is defined. by choosing the values of x_u and σ_u^2. that define its kernel function $K_u(d(x_u, x))$
- w_u : trained to maximize the fit of the network to the training data using the global error criterion
- because kernel functions are held fixed during the second stage, linear weight values w_u can be trained. very efficiently.

...number of hidden units / kernel functions ...

5 (a)	With a neat diagram explain reinforcement learning	7M	CO1	L2

<table>
<tr>
<td></td>
<td>

Reinforcement Learning:

- addresses question of how can autonomous agent senses and acts to choose optimal actions to achieve its goals.
- reward or penalty is given to train the agent.
- agent has to learn from indirect delayed reward.
- choose actions with highest cumulative reward.
- Q learning algorithm : acquires optimal control strategies from delayed rewards.
- RL - related to dynamic programming algorithms
  - used to solve optimization problems.

- set of states S
- set of actions A
- a_t on s_t at time t results in r_t : real-valued reward.
- produces a sequence of states

$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2}$

Task: learn the control policy $\pi : S \rightarrow A$
- maximizes the expected sum of these rewards
- future rewards are discounted exponentially by the delay.

Goal: learn to choose actions that maximize $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$ where $0 \le \gamma < 1$

eg. scheduling problem of choosing taxis.
- maximize function of wait time of passengers and total fuel costs of the taxi fleet

</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td>i) Which yields higher rewards? (Exploration / Exploitation)
ii)rewards received i time steps in future are discounted (uniformly / logarithmically / exponentially)
iii) if $\gamma = 0$ (only future rewards are considered / immediate rewards are considered)</td>
<td>3M</td>
<td>CO2</td>
<td>L3</td>
</tr>
<tr>
<td>6 (a)</td>
<td>Discuss Q learning and algorithm for deterministic Markov decision process.</td>
<td>5M</td>
<td>CO2</td>
<td>L2</td>
</tr>
</table>

Q Learning

V^*: can learn optimal policy only when agent has perfect knowledge of δ and r.

- if δ or r is unknown as is the case in practical situations, then V^* fails in choosing optimal policy.

Solution: Q Learning

$$Q(s,a) \equiv r(s,a) + \gamma V^*(\delta(s,a))$$

Q : reward received upon executing action a from state s, plus the value (discounted by γ)

$$\pi^*(s) = \underset{a}{\arg\max}\ Q(s,a)$$

- if agent learns the Q function instead of V^*, it will be able to select optimal actions even when it has no knowledge of r and δ

- agent can choose an optimal action without doing a lookahead search.

→ $Q(s,a)$ - r value plus V^* value of resulting state discounted by γ.

→ optimal policy - corresponds to selecting actions with maximal Q values

Q Learning Algorithm for deterministic Markov decision process

For each s,a initialize the table entry $\hat{Q}(s,a)$ to zero

Observe the current state s

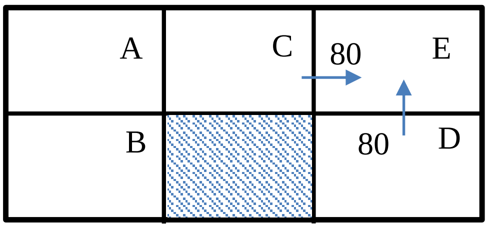
Do forever:

- 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:

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

- $s \leftarrow s'$

(b)	For the following problem where again can move within a grid and the goal is E, fin the Q matrix. Show steps. Discount factor = 0.8	5M	**CO2**	L3



State/Action	**A**	**B**	**C**	**D**	**E**
A	-1	0	0	-1	-1
B	0	-1	-1	-1	-1
C	0	-1	-1	-1	80
D	-1	-1	-1	-1	80
E	-1	-1	0	0	80

Q	**A**	**B**	**C**	**D**	**E**
A					
B					
C					
D					
E					

	A	**B**	**C**	**D**	**E**
A	0	40.96	64	0	0
B	51.2	0	0	0	0
C	51.2	0	0	0	80
D	0	0	0	0	80
E	0	0	64	64	80

CI **CCI** **HOD**

Course Outcomes		Blooms Level	Modules covered	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	Appraise the theory of Artificial intelligence and Machine Learning.	**L2**	**1,2**	3	3	2	2	0	2	2	0	0	0	0	0	0	2	0	3
CO2	Illustrate the working of AI and ML Algorithms.	**L3**	**2,3,4**	3	3	3	3	3	3	0	0	0	0	0	0	0	2	0	3
CO3	Demonstrate the applications of AI and ML.	**L2**	**4,5**	3	3	3	3	3	3	0	0	0	0	0	0	0	2	0	3

CO PO Mapping

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and team work	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		
PSO1	Develop applications using different stacks of web and programming technologies				
PSO2	Design and develop secure, parallel, distributed, networked, and digital systems				
PSO3	Apply software engineering methods to design, develop, test and manage software systems.				
PSO4	Develop intelligent applications for business and industry				