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

Internal Assessment Test 3 – July 2024

Sub:	**Machine Learning**					Sub Code:	**21AI63**	Branch:	**AInDS**
Date:		Duration:	**90 minutes**	Max Marks:	**50**	Sem	**VI**		**OBE**

		Answer any FIVE Questions	**MARKS**	**CO**	**RBT**
1	a)	Discuss Minimum Description Length algorithm	07	CO5	L1
	b)	Write Gibb's algorithm	03	CO5	L1
2		Explain Bayscian Belief Network with Example	10	CO5	L2
3		Define Following a) Conditional Probability b) Joint Probability c) Marginal Probability d) Bayesian Probability e) Prior Probability	10	CO5	L1
4		What is Ensemble Technique? Explain Baggging and Boosting Technique in detail	10	CO4	L2
5			10	CO5	L3

Sl. No.	Color	Legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

Dataset For Naive Bayes Classification
Using the above data, to identify the species of an entity with the following attributes.

X={Color=Green, Legs=2, Height=Tall, Smelly=No}

6		Explain Expectation – Maximization Algorithm, its Advantages and Drawbacks	10	CO4	L2

CI **CCI** **HOD**

Q 1 a) Discuss Minimum Description Length algorithm [7M]

Occam's razor: prefer the shortest hypothesis
MDL: prefer the hypothesis h that minimizes
where $L_C(x)$ is the description length of x under encoding C
Example:

- H = decision trees, D = training data labels
- $L_{C1}(h)$ is # bits to describe tree h
- $L_{C2}(D|h)$ is #bits to describe D given h
  - Note $L_{C2}(D|h) = 0$ if examples classified perfectly by h. Need only describe exceptions
- Hence h_{MDL} trades off tree size for training errors

$$h_{MAP} = \arg\max_{h \in H} P(D \mid h)P(h)$$

$$= \arg\max_{h \in H} \log_2 P(D \mid h) + \log_2 P(h)$$

$$= \arg\min_{h \in H} -\log_2 P(D \mid h) - \log_2 P(h) \quad (1)$$

Interesting fact from information theory:
The optimal (shortest expected length) code for an event with probability p is $\log_2 p$ bits.
So interpret (1):
$-\log_2 P(h)$ is the length of h under optimal code
$-\log_2 P(D|h)$ is length of D given h in optimal code
$\rightarrow$ prefer the hypothesis that minimizes
length(h)+length(misclassifications)

Q 1 b) Write Gibb's algorithm [3M]

Bayes optimal classifier provides best result, but can be expensive if many hypotheses.
Gibbs algorithm:
1. Choose one hypothesis at random, according to $P(h|D)$
2. Use this to classify new instance
Surprising fact: assume target concepts are drawn at random from H according to priors on H. Then:
$E[error_{Gibbs}] \leq 2E[error_{BayesOptimal}]$
Suppose correct, uniform prior distribution over H, then

- Pick any hypothesis from VS, with uniform probability
- Its expected error no worse than twice Bayes optimal

Q 2) Explain Bayesian Belief Network with Example [10M, Concept Explanation 5M and example 5M]

A **Bayesian Belief Network (BBN)**, also known as a **Bayesian Network** or **Belief Network**, is a probabilistic graphical model that represents a set of variables and their conditional dependencies using a directed acyclic graph (DAG). Each node in the graph represents a random variable, and the edges represent conditional dependencies between these variables.

Key Concepts of Bayesian Belief Networks:

1. **Nodes**: Each node in the network represents a random variable, which could be discrete or continuous.
2. **Edges**: Directed edges between nodes represent conditional dependencies. If there is an edge from node AAA to node BBB, AAA is called the parent of BBB, and BBB is conditionally dependent on AAA.
3. **Conditional Probability Tables (CPTs)**: Each node has an associated CPT that quantifies the effects of the parents on the node. For a node BBB with parent AAA, the CPT specifies the probability distribution of BBB given different states of AAA.
4. **D-separation**: A concept used to determine whether two nodes are independent, given a set of observed variables.
5. **Inference**: The process of computing the probability distribution of certain variables given evidence about others.

Example of a Bayesian Belief Network:

Consider a simple Bayesian Belief Network involving three variables:

- **Rain (R)**: Whether it is raining or not.
- **Sprinkler (S)**: Whether a sprinkler is turned on or not.
- **Wet Grass (W)**: Whether the grass is wet or not.

The relationships between these variables can be represented as:

- **R → W**: If it's raining, it might cause the grass to be wet.
- **S → W**: If the sprinkler is on, it might also cause the grass to be wet.

This forms a network where **Rain (R)** and **Sprinkler (S)** are the parent nodes, and **Wet Grass (W)** is the child node.

Step 1: Structure of the Network

```
markdown
Copy code
   R         S
    \       /
     \     /
       W
```

- **R → W**: The edge from **Rain** to **Wet Grass** indicates that rain influences whether the grass is wet.
- **S → W**: The edge from **Sprinkler** to **Wet Grass** indicates that the sprinkler also influences whether the grass is wet.

Step 2: Conditional Probability Tables (CPTs)

We define the CPTs for each node. Assume the following probabilities:

- **P(R)**: Probability of rain.
  - P(R = True) = 0.2
  - P(R = False) = 0.8
- **P(S)**: Probability of the sprinkler being on.
  - P(S = True) = 0.5
  - P(S = False) = 0.5
- **P(W | R, S)**: Probability of the grass being wet given the states of rain and sprinkler.
  - P(W = True | R = True, S = True) = 0.99
  - P(W = True | R = True, S = False) = 0.8

- P(W = True | R = False, S = True) = 0.9
- P(W = True | R = False, S = False) = 0.0

Step 3: Inference

Now, suppose we observe that the grass is wet (W = True). We want to infer the probability that it rained (P(R = True | W = True)).

Using Bayes' theorem and the CPTs, we can compute:

$$P(R = True|W = True) = \frac{P(W=True|R=True)\cdot P(R=True)}{P(W=True)}$$

Where $P(W = True)$ is computed by summing over all possible combinations of R and S:

$$P(W = True) = P(W = True|R = True, S = True) \cdot P(R = True) \cdot P(S = True) + (\text{other combinations})$$

By calculating these probabilities, we can determine how likely it is that it rained given that the grass is wet.

Step 4: Updating Beliefs

If we later observe that the sprinkler was off (S = False), we can update our belief about whether it rained:

$$P(R = True|W = True, S = False) = \frac{P(W=True|R=True,S=False)\cdot P(R=True)}{P(W=True|S=False)}$$

This process of updating beliefs based on new evidence is central to Bayesian networks.

Q 3) Define Following [10M, each 2M]

a) Conditional Probability
b) Joint Probability
c) Marginal Probability
d) Bayesian Probability
e) Prior Probability

Conditional probability

Conditional probability is the probability of an event occurring given that another event has already occurred. It helps us update our understanding of the likelihood of an event based on new information.

The conditional probability of an event A occurring given that event B has occurred is denoted as $P(A \mid B)$. It is defined as:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Where:

- $P(A \mid B)$ is the conditional probability of A given B.
- $P(A \cap B)$ is the joint probability that both A and B occur.
- $P(B)$ is the probability that B occurs.

Joint probability

Joint probability is the probability of two or more events happening at the same time. If you have two events, AAA and BBB, the joint probability of AAA and BBB is the probability that both events occur simultaneously. It is denoted as P(A∩B)P(A \cap B)P(A∩B) or P(A and B)P(A \text{ and } B)P(A and B).

For two events A and B, the joint probability $P(A \cap B)$ is defined as:

$$P(A \cap B) = P(A) \times P(B \mid A)$$

This formula can be interpreted as the probability of event A occurring multiplied by the probability of event B occurring given that A has already occurred.

If A and B are independent events (meaning the occurrence of one does not affect the occurrence of the other), then the joint probability is simply the product of their individual probabilities:

$$P(A \cap B) = P(A) \times P(B)$$

Marginal Probability

Marginal probability refers to the probability of a single event occurring without consideration of any other events. It is obtained by summing or integrating the joint probabilities over all possible outcomes of the other events. Marginal probability gives the overall likelihood of an event happening, irrespective of the outcomes of other related events.

For two events A and B, the marginal probability of event A, denoted as $P(A)$, is calculated by summing the joint probabilities of A with all possible outcomes of B:

$$P(A) = \sum_{b \in B} P(A \cap B = b)$$

If the variables are continuous, integration would be used instead of summation.

Bayesian Probability

Bayesian probability is an interpretation of probability that represents a degree of belief or certainty about the occurrence of an event, given the available evidence. It contrasts with the frequentist interpretation, which views probability as the long-run frequency of an event occurring. Bayesian probability is foundational to Bayesian inference, where prior knowledge is updated with new evidence to form a posterior probability.

Key Concepts in Bayesian Probability:

1. **Prior Probability (P(A))**: This represents the initial degree of belief in an event before new evidence is taken into account. It is often based on prior knowledge or subjective judgment.
2. **Likelihood (P(B|A))**: This is the probability of observing the evidence B, given that the event A is true.
3. **Posterior Probability (P(A|B))**: This is the updated probability of the event A after considering the new evidence B. It represents the revised degree of belief in the event.
4. **Bayes' Theorem**: The relationship between these quantities is formalized by Bayes' theorem, which is used to update the probability of a hypothesis based on new evidence:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Where:

- $P(A \mid B)$ is the posterior probability.
- $P(B \mid A)$ is the likelihood.
- $P(A)$ is the prior probability.
- $P(B)$ is the marginal likelihood or the total probability of the evidence.

Prior Probability

Prior Probability (P(A)): This represents the initial degree of belief in an event before new evidence is taken into account. It is often based on prior knowledge or subjective judgment.

Q 4) What is Ensemble Technique? Explain Baggging and Boosting Technique in detail [10M, Concept=2M,Bagging=4M Boosting=4M]

Ensemble techniques in machine learning refer to methods that combine the predictions of multiple models (often referred to as "weak learners" or "base models") to create a more robust and accurate model. The idea behind ensemble methods is that by aggregating the predictions of multiple models, the combined model can reduce the risk of errors and improve generalization to new data.

Types of Ensemble Techniques:

There are several ensemble techniques, but two of the most common ones are **Bagging** and **Boosting**.

Bagging (Bootstrap Aggregating):

Bagging is an ensemble technique designed to improve the stability and accuracy of machine learning algorithms by reducing variance. The key idea is to create multiple versions of a model by training them on different subsets of the data, and then combining their predictions.

Steps in Bagging:

1. **Bootstrap Sampling**:
    - Create multiple subsets of the original training dataset by sampling with replacement (this is known as bootstrapping). Each subset is the same size as the original dataset but may have duplicate instances.
2. **Model Training**:
    - Train a base model (e.g., decision trees) on each subset independently. Because the training data differs for each model, the individual models may capture different patterns in the data.
3. **Aggregation**:
    - For classification problems, predictions are typically aggregated using majority voting: each model in the ensemble votes on the class label, and the label with the most votes is chosen.
    - For regression problems, the predictions are averaged.

Example:

Imagine we have a dataset of 100 instances and we're using decision trees as the base model. Bagging would create, say, 10 different datasets (each of size 100) by randomly sampling with replacement. Each decision tree is trained on one of these datasets, and their predictions are combined using majority voting or averaging.

Popular Bagging Algorithms:

- **Random Forest**: A popular bagging technique where the base models are decision trees, and in addition to bootstrap sampling, random feature selection is applied during tree construction.

Advantages:

- Reduces variance, making the model less prone to overfitting.
- Works well with high-variance models like decision trees.

Disadvantages:

- Doesn’t significantly reduce bias, so if the base model is biased, the ensemble might also be biased.

---

2. Boosting:

Boosting is an ensemble technique that focuses on reducing bias and variance by training a sequence of models, where each subsequent model attempts to correct the errors made by the previous ones. Unlike bagging, where models are trained independently, boosting builds models sequentially.

Steps in Boosting:

1. **Initialize Weights**:
    - Start with equal weights assigned to each instance in the training data. These weights reflect the importance of each instance in training.

2. **Model Training**:
    - Train a base model on the weighted data. After the model is trained, evaluate its performance on the training set.
    - Increase the weights of the incorrectly predicted instances, so that the next model in the sequence focuses more on these difficult cases.
3. **Update and Combine**:
    - The process is repeated for a specified number of iterations or until no further improvement is observed.
    - The final prediction is a weighted combination of the predictions of all models in the sequence.

Popular Boosting Algorithms:

- **AdaBoost (Adaptive Boosting)**: Adjusts the weights of misclassified instances and combines the weak learners' outputs with a weighted majority vote.
- **Gradient Boosting**: Focuses on optimizing a loss function by adding models that predict the residual errors of previous models. Variants include **XGBoost** and **LightGBM**.

Advantages:

- Reduces both bias and variance, making the model more accurate.
- Particularly effective for complex datasets.

Disadvantages:

- Prone to overfitting if not properly regularized.
- Typically requires careful tuning of parameters.
- Slower to train compared to bagging, as models are built sequentially.

Q 5) Dataset For Naive Bayes Classification Using the above data, to identify the species of an entity with the following attributes.

X={Color=Green, Legs=2, Height=Tall, Smelly=No}

To predict the class label for the above attribute set, we will first calculate the probability of the species being M or H in total.

P(Species=M)=4/8=0.5
P(Species=H)=4/8=0.5

Next, we will calculate the conditional probability of each attribute value for each class label.

P(Color=White/Species=M)=2/4=0.5
P(Color=White/Species=H)=¾=0.75
P(Color=Green/Species=M)=2/4=0.5
P(Color=Green/Species=H)=¼=0.25
P(Legs=2/Species=M)=1/4=0.25
P(Legs=2/Species=H)=4/4=1
P(Legs=3/Species=M)=3/4=0.75
P(Legs=3/Species=H)=0/4=0
P(Height=Tall/Species=M)=3/4=0.75
P(Height=Tall/Species=H)=2/4=0.5
P(Height=Short/Species=M)=1/4=0.25
P(Height=Short/Species=H)=2/4=0.5
P(Smelly=Yes/Species=M)=3/4=0.75

P(Smelly=Yes/Species=H)=1/4=0.25
P(Smelly=No/Species=M)=1/4=0.25
P(Smelly=No/Species=H)=3/4=0.75

We can tabulate the above calculations in the tables for better visualization.

The conditional probability table for the Color attribute is as follows.

Color	M	H
White	0.5	0.75
Green	0.5	0.25

Conditional Probabilities for Color Attribute

The conditional probability table for the Legs attribute is as follows.

Legs	M	H
2	0.25	1
3	0.75	0

Conditional Probabilities for Legs Attribute

The conditional probability table for the Height attribute is as follows.

Height	M	H
Tall	0.75	0.5
Short	0.25	0.5

Conditional Probabilities for Height Attribute

The conditional probability table for the Smelly attribute is as follows.

Smelly	M	H
Yes	0.75	0.25
No	0.25	0.75

Conditional Probabilities for Smelly Attribute

Now that we have calculated the conditional probabilities, we will use them to calculate the probability of the new attribute set belonging to a single class.

Let us consider X= {Color=Green, Legs=2, Height=Tall, Smelly=No}.

Then, the probability of X belonging to Species M will be as follows.

P(M/X)=P(Species=M)*P(Color=Green/Species=M)*P(Legs=2/Species=M)*P(Height=Tall/Species=M)*P(Smelly=No/Species=M)
=0.5*0.5*0.25*0.75*0.25
=0.0117

Similarly, the probability of X belonging to Species H will be calculated as follows.

P(H/X)=P(Species=H)*P(Color=Green/Species=H)*P(Legs=2/Species=H)*P(Height=Tall/Species=H)*P(Smelly=No/Species=H)
=0.5*0.25*1*0.5*0.75
=0.0468

So, the probability of X belonging to Species M is 0.0117 and that to Species H is 0.0468. Hence, we will assign the entity X with attributes {Color=Green, Legs=2, Height=Tall, Smelly=No} to species H.

Q 6) Explain Expectation – Maximization Algorithm, its Advantages and Drawbacks [Algorithm=2 M, Advantages=4M, Drawbacks=4M]

Steps of the EM Algorithm:

1. **Initialization**:
   - Start by initializing the parameters of the model (e.g., means, variances, and mixing coefficients in a Gaussian mixture model).
2. **Expectation Step (E-step)**:
   - Given the current parameter estimates, compute the expected value of the latent variables. This involves calculating the posterior probabilities of the latent variables given the observed data and the current parameter estimates.
   - Essentially, the E-step computes the expected value of the complete data log-likelihood.
3. **Maximization Step (M-step)**:
   - Update the model parameters by maximizing the expected complete data log-likelihood obtained from the E-step.
   - The new parameters are chosen to maximize the likelihood of the observed data, given the expected values of the latent variables from the E-step.
4. **Iterate**:
   - Repeat the E-step and M-step until the parameter estimates converge, meaning that subsequent iterations do not significantly change the parameters.
5. **Convergence**:
   - The algorithm converges when the change in the log-likelihood (or another convergence criterion) between iterations falls below a predefined threshold.

Advantages of the EM Algorithm:

1. **Handling Incomplete Data**:
   - EM is particularly useful when dealing with datasets that have missing or incomplete data. It can still provide parameter estimates in such situations.
2. **Convergence Properties**:
   - The EM algorithm is guaranteed to converge to a local maximum of the likelihood function, making it a reliable method in many practical applications.
3. **Flexibility**:
   - EM can be applied to a wide range of models, not just Gaussian Mixture Models. It is a general framework for maximum likelihood estimation with latent variables.
4. **Simplicity**:
   - The algorithm is conceptually straightforward, involving just two steps (E-step and M-step) that are repeated iteratively.

Drawbacks of the EM Algorithm:

1. **Convergence to Local Optima**:
   - EM is not guaranteed to find the global maximum of the likelihood function. It may converge to a local maximum, especially in complex models with many parameters or in multimodal likelihood surfaces.
2. **Initialization Sensitivity**:
   - The quality of the final solution can be highly dependent on the initial parameter values. Poor initialization can lead to convergence to suboptimal solutions.
3. **Computational Complexity**:
   - The E-step can be computationally intensive, especially when dealing with large datasets or complex models. Additionally, the M-step might require solving complex optimization problems.
4. **Slow Convergence**:
   - The EM algorithm can converge slowly, especially near the optimum. The algorithm may require many iterations to achieve convergence, which can be computationally expensive.
5. **Assumption of Model Correctness**:
   - EM assumes that the model structure is correct (e.g., that the data indeed comes from a mixture of Gaussians). If the model is misspecified, the algorithm may not yield meaningful results.