Code No: 156BN R18

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B. Tech III Year II Semester Examinations, February/March - 2022

MACHINE LEARNING

(Computer science and Engineering)

Time: 3 Hours **Max. Marks: 75**

Answer any five questions

All questions carry equal marks

- - -

1. a) Discuss about the basic Decision Tree Learning algorithm. [7+8]
b) Briefly explain the need for Inductive Bias in Decision Tree Learning.

2. a) Explain the Find-S: Finding a Maximally Specific Hypothesis in detail.
b) Explain the issues in decision tree learning. [10+5]

3. a) Present the Backpropagation algorithm for feedforward networks and explain each step in it.
b) Explain how to estimate hypothesis accuracy. [9+6]

4. a) Define the terms estimation bias and confidence intervals.
b) Discuss the central limit theorem for deriving confidence intervals in detail.
c) Explain the representation of neural networks. [5+5+5]

5. a) Design the Brute Force Bayesian concept learning algorithm and elaborate.
b) Explain the Mistake Bound for the Halving Algorithm. [8+7]

6. a) Explain the Maximum Likelihood Hypotheses for predicting probabilities.
b) Elaborate the Locally Weighted Linear Regression. [8+7]

7. a) Explain the Q–learning with a suitable example.
b) Explain about the hypothesis space search. [8+7]

8. a) Discuss about the Explanation-based Learning of Search Control Knowledge. [8+7]
b) Explain how to initialize the hypothesis by using prior knowledge.

---ooOoo---

Answer Key

1. a) Discuss about the basic Decision Tree Learning algorithm.

1. Decision Trees: Decision Tree Learning is a supervised machine learning technique used for classification and regression. It involves constructing a tree-like structure to make decisions.
2. Root Node: The tree starts with a root node that represents the entire dataset. It selects the best attribute to split the data based on criteria like information gain or Gini impurity.
3. Internal Nodes: Internal nodes of the tree represent feature attributes. Each internal node is associated with a decision rule based on a feature.
4. Leaf Nodes: Leaf nodes represent the class labels or regression values. Each leaf node corresponds to a final decision or prediction.
5. Splitting: The algorithm recursively splits the data into subsets based on the selected attributes until a stopping criterion is met. This criterion can be the maximum depth of the tree or the minimum number of data points per leaf.
6. Attribute Selection: The choice of attribute for splitting is crucial. Common methods include Information Gain, Gini Impurity, and Gain Ratio.
7. Pruning: Decision trees can be pruned to prevent overfitting. Pruning involves removing branches that do not significantly improve predictive accuracy.
8. Decision Making: To classify a new instance, it traverses the tree from the root to a leaf node following the decision rules at each node.
9. Advantages: Decision Trees are interpretable, easy to visualize, and can handle both categorical and numerical data.
10. Disadvantages: They can be prone to overfitting, especially with deep trees. Ensuring a proper balance is essential.

1. b) Briefly explain the need for Inductive Bias in Decision Tree Learning.

1. Generalization: Decision trees aim to generalize from the training data to make accurate predictions on unseen data.
2. Inductive Bias: Inductive bias is a set of assumptions that guide the learning process. In decision tree learning, it is crucial to have a suitable inductive bias to avoid overfitting.
3. Overfitting: Without an appropriate inductive bias, decision trees can become too complex and fit the noise in the training data, leading to poor generalization of new data.

4. Simplicity: Inductive bias helps in selecting simpler and more interpretable trees, which are less likely to overfit.

5. Pruning: Pruning is one way to introduce inductive bias by favoring smaller trees during tree construction.
6. Occam's Razor: The principle of Occam's Razor suggests that among competing hypotheses, the simplest one should be chosen. Inductive bias aligns with this principle.
7. Domain Knowledge: Inductive bias can also incorporate domain-specific knowledge to guide the learning process.
8. Balance: It strikes a balance between fitting the training data well and making predictions that generalize to new, unseen data.
9. Improved Performance: An appropriate inductive bias enhances the decision tree's performance by preventing it from becoming too complex and overfitting the training data.
10. Need for Customization: The choice of inductive bias may vary depending on the problem, and it should be customized to achieve the best results in Decision Tree Learning.

2. a) Explain the Find-S: Finding a Maximally Specific Hypothesis in detail.

1. Find-S is a machine learning algorithm used for concept learning in the context of supervised learning.
2. Its goal is to find the most specific hypothesis that is consistent with the training data.
3. The hypothesis is represented as a conjunction of literals, where each literal corresponds to an attribute-value pair.
4. Initially, the hypothesis is set to the most specific hypothesis, which is a conjunction of all possible attribute-value pairs.
5. It iteratively refines the hypothesis based on the training examples. If an example is positive, it generalizes the hypothesis to include the attribute-value pairs present in the example.
6. The Find-S algorithm continues to refine the hypothesis until it covers all positive examples and no negative examples.
7. It is a top-down search approach, starting with the most specific hypothesis and gradually making it more general.
8. Find-S is guaranteed to find the maximally specific hypothesis if the target concept is in the hypothesis space.
9. It may not always find the correct hypothesis if there is noise in the data.
10. Find-S is a simple and intuitive algorithm but may not be suitable for complex concept spaces.

2. b) Explain the issues in decision tree learning.

1. Overfitting: One of the primary issues in decision tree learning is overfitting. Decision trees can become too complex and fit the training data noise, leading to poor generalization of unseen data.
2. Bias towards features: Decision trees tend to favor features with many values or high cardinality, as they can lead to finer splits. This can result in biased trees.
3. Instability: Small changes in the training data can lead to significantly different decision trees. Decision trees are unstable classifiers.
4. Difficulty in handling continuous attributes: Decision trees are naturally suited for discrete attributes. Handling continuous attributes requires discretization, which can be non-trivial.
5. Greedy nature: Decision tree algorithms use a greedy approach to select the best split at each node. This may not lead to the globally optimal tree.
6. Lack of support for missing data: Traditional decision tree algorithms do not handle missing values well, and missing data can lead to biased splits.
7. Imbalanced data: Decision trees can be biased towards the majority class in imbalanced datasets, leading to poor classification of minority classes.
8. Difficulty in handling irrelevant attributes: Decision trees may include irrelevant attributes in the tree, which can reduce interpretability and efficiency.
9. Limited expressiveness: Decision trees may not be able to represent complex decision boundaries effectively, especially for XOR-like problems.
10. Difficulty in pruning: Pruning decision trees to improve generalization can be a challenging task, requiring validation data.

3. a) Present the Backpropagation algorithm for feedforward networks and explain each step in it.

1. Backpropagation is a supervised learning algorithm used for training feedforward neural networks.
2. It is based on the gradient descent optimization technique and is used to minimize the error between the network's predictions and the actual target values.
3. Backpropagation involves both the forward pass and the backward pass.
4. Forward Pass:
The input data is fed into the network, and the activations of each neuron are calculated layer by layer using the weights and biases.
The output of the network is compared to the actual target values, and the error (usually mean squared error) is computed.
5. Backward Pass (Backpropagation):
The gradients of the error with respect to the weights and biases are computed layer by layer, starting from the output layer and moving backward.
The chain rule is used to calculate these gradients.

Weight updates are performed to minimize the error by adjusting the weights and biases in the opposite direction of the gradient.

6. Gradient Descent:

The weight updates are typically performed using gradient descent, where the learning rate determines the step size for weight adjustments.

The process is repeated for multiple epochs until the error converges or reaches a predefined threshold.

7. The key steps in backpropagation are calculating the gradients, updating the weights, and repeating this process iteratively.

8. Activation functions like sigmoid or ReLU are used to introduce non-linearity into the network, allowing it to learn complex mappings.

9. Backpropagation is the foundation for training deep neural networks, and variations like stochastic gradient descent (SGD) and mini-batch gradient descent are commonly used for efficiency.

10. Backpropagation requires careful hyperparameter tuning, including the learning rate and network architecture, to achieve good training results.

3. b) Explain how to estimate hypothesis accuracy.

1. Hypothesis accuracy refers to the ability of a machine learning model to make correct predictions on unseen data.

2. To estimate hypothesis accuracy, a common practice is to split the available dataset into two parts: a training set and a test set.

3. The training set is used to train the model, while the test set is used to evaluate its performance on unseen data.

4. Accuracy is typically measured using metrics such as:

Classification Accuracy: For classification tasks, it's the ratio of correctly predicted instances to the total number of instances in the test set.

Mean Squared Error (MSE): For regression tasks, it measures the average squared difference between predicted and actual values.

5. Cross-validation can be used to provide a more robust estimate of accuracy by splitting the data into multiple folds and evaluating the model on different combinations of training and test sets.

6. Precision, recall, F1-score, ROC curves, and AUC are other metrics used to assess the accuracy of models, especially in classification tasks.

7. Hypothesis accuracy can vary depending on the choice of algorithm, hyperparameters, and preprocessing steps, so it's essential to experiment with different configurations.

8. For hypothesis accuracy estimation, it's crucial to have a well-defined evaluation protocol, including the choice of performance metrics and the handling of imbalanced datasets if applicable.

9. In some cases, accuracy may not be the only relevant metric, and domain-specific metrics or business objectives should also be considered.
10. Model accuracy should be interpreted in the context of the specific problem and domain to ensure it meets the desired level of performance.

4. a) Define the terms estimation bias and confidence intervals.

1. Estimation Bias:

Estimation bias refers to the systematic error or deviation of an estimate from the true or population parameter.

In other words, it represents the tendency of an estimator to consistently overestimate or underestimate the parameter it's trying to estimate.

Estimation bias can arise due to various factors, such as sampling methods, measurement errors, or the choice of estimation technique.

A biased estimator consistently gives incorrect results, and reducing bias is a crucial goal in statistical estimation.

2. Confidence Intervals:

Confidence intervals (CI) are a range of values calculated from sample data that is likely to contain the true population parameter with a certain level of confidence.

CI provides a measure of the uncertainty associated with an estimate.

It consists of two values: an upper bound and a lower bound, defining a range within which the true parameter is expected to lie.

The level of confidence, often denoted as (1 - α), represents the probability that the interval contains the true parameter. Common values for α include 0.05 (95% confidence) and 0.01 (99% confidence).

The width of the confidence interval depends on the sample size and the variability of the data.

Widening the CI increases confidence but decreases precision while narrowing it decreases confidence but increases precision.

4. b) Discuss the central limit theorem for deriving confidence intervals in detail.

1. Central Limit Theorem (CLT):

The Central Limit Theorem is a fundamental concept in statistics that states that, regardless of the population distribution, the sampling distribution of the sample mean (or other sample statistics) approaches a normal distribution as the sample size increases.

This theorem is critical for deriving confidence intervals and making inferences about population parameters.

2. Key Points of the CLT:

For a sufficiently large sample size, the distribution of the sample mean becomes approximately normal, regardless of the shape of the population distribution.

The CLT assumes that the samples are drawn randomly and independently from the population.

The larger the sample size, the closer the sampling distribution of the mean approximates a normal distribution.

The CLT provides a theoretical basis for using the normal distribution as an approximation for the sampling distribution of the mean, even when dealing with non-normally distributed populations.

3. Deriving Confidence Intervals Using CLT:

To derive a confidence interval for a population parameter (e.g., the population mean), the CLT allows us to assume that the sampling distribution of the sample mean is approximately normal.

With this assumption, we can calculate the standard error of the sample mean and use it to construct a confidence interval.

The formula for the confidence interval is typically based on the standard normal distribution (z-distribution) for large sample sizes or the t-distribution for smaller sample sizes.

The level of confidence determines the critical value from the chosen distribution.

4. Practical Use:

The CLT is widely used in hypothesis testing, confidence interval estimation, and various statistical analyses.

It provides a foundation for making statistical inferences about population parameters from sample data.

Understanding the CLT is crucial for statisticians and data analysts when working with real-world data, as it enables the application of common statistical techniques.

4. c) Explain the representation of neural networks.

1. Neural networks are composed of layers of interconnected nodes or neurons and are used for various machine learning tasks, including deep learning.

2. Representation of Neural Networks:

Input Layer: The first layer of a neural network receives input features or data. Each neuron in this layer corresponds to a feature.

Hidden Layers: Between the input and output layers, there can be one or more hidden layers. These layers process the input data through a series of weighted connections and apply activation functions to produce intermediate representations.

Output Layer: The final layer, known as the output layer, produces the network's predictions or outputs. The number of neurons in this layer depends on the task (e.g., one neuron for binary classification, multiple neurons for multi-class classification or regression).

Connections (Weights): Each connection between neurons has an associated weight that determines the strength of the connection. Weights are adjusted during training to minimize error.

Activation Functions: Neurons apply activation functions to their weighted inputs to introduce non-linearity into the model. Common activation functions include sigmoid, ReLU (Rectified Linear Unit), and tanh.

Feedforward: Information flows from the input layer through the hidden layers to the output layer in a feedforward manner.

Backpropagation: During training, the network uses backpropagation to update weights and minimize the error between predicted and actual values.

3. Representation Complexity:

The architecture of a neural network, including the number of layers and neurons, determines its capacity to model complex relationships in data.

Deep neural networks with many hidden layers are capable of representing highly complex functions and are used for deep learning tasks.

The choice of activation functions and regularization techniques also influences the network's representation.

4. Neural networks can be designed for various tasks, such as image classification, natural language processing, and reinforcement learning, by adapting the network architecture and training process to the specific problem.

5. a) Design the Brute Force Bayesian concept learning algorithm and elaborate.

1. The Brute Force Bayesian concept learning algorithm is a method to learn and classify data based on Bayesian probability theory.
2. It considers all possible hypotheses for classification and calculates their posterior probabilities.
3. For each hypothesis, it computes the likelihood of the data given that hypothesis and the prior probability of the hypothesis itself.
4. The algorithm then normalizes these probabilities to obtain the posterior probabilities for each hypothesis.
5. Finally, it selects the hypothesis with the highest posterior probability as the classification result.
6. This algorithm is considered "brute force" because it exhaustively evaluates all possible hypotheses.
7. It is useful when dealing with small datasets and simple hypothesis spaces.
8. However, it becomes impractical for large datasets or complex hypothesis spaces due to the combinatorial explosion of possibilities.

9. Despite its computational intensity, it provides an accurate and theoretically sound way of classifying data.
10. In summary, the Brute Force Bayesian concept learning algorithm is a straightforward approach that considers all hypotheses to make data classifications but may not be efficient for large or complex problems.

5. b) Explain the Mistake Bound for the Halving Algorithm.

1. The Halving Algorithm is a concept learning algorithm that iteratively eliminates half of the remaining hypotheses based on their performance on labeled examples.
2. The Mistake Bound for the Halving Algorithm represents the maximum number of mistakes it can make during the learning process.
3. It is related to the number of hypotheses and the number of training examples.
4. The Mistake Bound is calculated using the formula: Mistake Bound = log2(N), where N is the number of hypotheses.
5. This bound signifies that the Halving Algorithm is guaranteed to make at most log2(N) mistakes before converging to a correct hypothesis.
6. It demonstrates the efficiency and reliability of the algorithm in finding the correct concept.
7. As the number of hypotheses decreases (i.e., when there are fewer possible concepts), the Mistake Bound reduces, indicating faster convergence.
8. The Mistake Bound is a fundamental theoretical concept in the analysis of online learning algorithms.
9. It assures that with a sufficient number of examples, the algorithm will eventually identify the correct concept.
10. In summary, the Mistake Bound for the Halving Algorithm provides a theoretical guarantee of the maximum number of mistakes the algorithm can make during the learning process.

6. a) Explain the Maximum Likelihood Hypotheses for predicting probabilities.

1. Maximum Likelihood Hypotheses is a method used for estimating probabilities in statistical modeling.
2. It assumes that the best estimate for the probability of an event is the relative frequency of that event in the given data.
3. In the context of predicting probabilities, it is used to find the parameters of a probabilistic model that maximizes the likelihood of the observed data.
4. Mathematically, it involves finding the parameter values that maximize the likelihood function, which measures how well the model explains the observed data.

5. Maximum Likelihood Estimation (MLE) is widely used in various fields, including machine learning and statistics.
6. It is particularly useful when dealing with parametric models, such as the normal distribution.
7. MLE provides a point estimate of the parameters that are most likely to have generated the observed data.
8. However, it does not provide information about the uncertainty or variability of the estimates.
9. MLE can be sensitive to outliers in the data, leading to biased parameter estimates in some cases.
10. In summary, the Maximum Likelihood Hypothesis is a method for estimating probabilities by maximizing the likelihood of the data under a given model, commonly used in statistical modeling.

6. b) Elaborate the Locally Weighted Linear Regression.

1. Locally Weighted Linear Regression (LWLR) is a non-parametric regression technique used for predicting a target variable based on input features.
2. Unlike traditional linear regression, LWLR assigns different weights to data points based on their proximity to the prediction point.
3. It assumes that data points closer to the prediction point are more relevant and should contribute more to the regression.
4. LWLR uses a weighted linear regression model, where the weights are determined by a kernel function.
5. The kernel function assigns higher weights to nearby data points and lower weights to distant ones.
6. The model is trained by fitting a linear regression line to the weighted data points in the local neighborhood of the prediction point.
7. It adapts to the local characteristics of the data, allowing it to capture complex and non-linear relationships.
8. LWLR is often used for tasks where the relationship between inputs and outputs varies across the input space.
9. It is sensitive to the choice of the kernel function and the bandwidth parameter, which controls the influence of data points.
10. In summary, Locally Weighted Linear Regression is a regression technique that assigns different weights to data points based on their proximity, making it suitable for capturing local patterns and non-linear relationships in data.

7. a) Explain Q-learning with a suitable example.

1. Q-learning is a reinforcement learning technique used for training agents to make decisions in an environment.

2. It involves learning a Q-table, where Q-values represent the expected cumulative rewards for taking specific actions in specific states.
3. The Q-value update rule is: $Q(s, a) \leftarrow Q(s, a) + \alpha \cdot [R(s) + \gamma \cdot \max Q(s', a') - Q(s, a)]$.
4. Here, s is the current state, a is the chosen action, α is the learning rate, $R(s)$ is the immediate reward, γ is the discount factor, s' is the next state, and a' is the next action.
5. Let's consider an example of teaching a robot to navigate a maze. The robot explores different paths and learns Q-values for each state-action pair.
6. As it explores, it updates its Q-values based on the rewards it receives, aiming to maximize the cumulative reward.
7. Eventually, the robot learns an optimal policy to reach the maze's goal with the highest reward.

7. b) Explain about the hypothesis space search.

1. Hypothesis space search is a fundamental concept in machine learning and pattern recognition.
2. It involves searching for the best hypothesis (model) that fits a given dataset.
3. The hypothesis space represents all possible models that can explain the data.
4. The search process aims to find the hypothesis that minimizes the error or loss function.
5. One common example is in linear regression, where the hypothesis space includes all possible linear functions.
6. The search involves adjusting the model's parameters to find the best-fit line that minimizes the least-squares error.
7. Hypothesis space search can be guided by various search algorithms, such as gradient descent or genetic algorithms.
8. The choice of hypothesis space and search algorithm depends on the specific machine learning problem.
9. It's essential to balance model complexity (underfitting vs. overfitting) during the search.
10. The goal is to discover a hypothesis that generalizes well to unseen data.

8. a) Discuss Explanation-based Learning of Search Control Knowledge.

1. Explanation-based learning (EBL) is a machine learning approach that focuses on learning control knowledge for problem-solving.
2. It involves learning from explanations or justifications provided for problem-solving decisions.
3. EBL aims to capture the reasoning or heuristic strategies used by experts in problem-solving.

4. The learned control knowledge can then guide search algorithms more effectively.
5. In search problems, EBL can be used to learn rules or heuristics that determine which nodes or states to explore during the search.
6. It reduces the search space by selecting promising paths based on past explanations.
7. EBL is especially useful in domains where human expertise can provide valuable insights.
8. It can improve the efficiency and effectiveness of search algorithms.
9. EBL has applications in various fields, including natural language processing and game playing.
10. EBL systems adapt and refine their control knowledge as they gain more experience.

8. b) Explain how to initialize the hypothesis by using prior knowledge.

1. Initializing hypotheses with prior knowledge is a common practice in machine learning.
2. Prior knowledge includes any existing information or domain expertise about the problem.
3. It provides a starting point for the learning algorithm and can speed up convergence.
4. For example, in Bayesian statistics, we can initialize prior probability distributions that represent our beliefs before observing data.
5. In deep learning, pre-trained neural network weights are often used as initialization.
6. Prior knowledge can be in the form of constraints, rules, or heuristics.
7. Initializing hypotheses with prior knowledge helps regularize the learning process.
8. It reduces the risk of overfitting, especially when the dataset is small.
9. However, careful consideration is needed to ensure that prior knowledge aligns with the problem and doesn't introduce bias.
10. Combining prior knowledge with data-driven learning can lead to more robust and accurate models.