CS-233 Theoretical Exercise

Mar 2025

1 Understanding Decision Tree

Question 1: Are the following statements true?

1. Decision trees are prone to overfitting, especially when they are too deep.
2. Decision trees intrinsically perform feature selection during the training process.
3. A decision tree can only be used for classification problems.
4. Decision trees have linear decision boundaries, similar to logistic regression.
5. Decision trees make predictions based on the majority class or average value in leaf nodes, while KNN makes predictions based on the majority class or average value of nearest neighbors.

Solution: 1. True. 2. True. 3. False. 4. False. 5. True.

2 Building Decision Tree

Question 2: Considering the data in Table 2, a dataset of 8 students about whether they like the famous movie Gladiator. Our goal is to build a decision tree classifier using *Gender, Major* as features and whether or not the students like the movie.

Gender	Major	Like
Male	Math	Yes
Female	History	No
Male	CS	Yes
Female	Math	No
Female	Math	No
Male	CS	Yes
Male	History	No
Female	Math	Yes

Throughout this question, we will use Entropy as the splitting criterion. You may use entropy with base 2: $Q(\mathcal{S}) = -\sum_{k=1}^{K} p^k \log_2 p^k$. Some useful values: $\log(\frac{1}{2}) = -1$, $\log(\frac{1}{4}) = -2$, $\log(\frac{3}{4}) = -0.41$.

Question 2.1: Initial Entropy. To start with, we have all 8 samples on the root node. Then what is the initial entropy of this dataset?

Solution: There are 4 likes and 4 not. Hence, $Q(\mathcal{S}) = -\frac{4}{8}\log\left(\frac{4}{8}\right) - -\frac{4}{8}\log\left(\frac{4}{8}\right) = 1$

Question 2.2: Information Gain. Now, to split the samples (i.e. grow the tree), we need to compare the information gain for the two features (i.e. Gender and Major). Which feature is the best to split the data as per your results?

Solution: We compute the information gain of the two features and choose the higher one. The information gain is $Q(\mathcal{S}) - \sum_\tau \frac{|\mathcal{S}^\tau|}{|\mathcal{S}|} Q\left(S^\tau\right)$.

- For Gender, we have

$$\sum_\tau \frac{|\mathcal{S}^\tau|}{|\mathcal{S}|} Q\left(S^\tau\right) = 0.5 \cdot \left(-\frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{3}{4}\log\left(\frac{3}{4}\right)\right) + 0.5 \cdot \left(-\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right)\right) \quad (1)$$

$$= 0.5 \cdot (-0.25 \cdot -2 - 0.75 \cdot -0.41) + 0.5 \cdot (-0.75 \cdot -0.41 - 0.25 \cdot -2) \quad (2)$$

$$= 0.81 \quad (3)$$

- For Major, we have $\sum_\tau \frac{|\mathcal{S}^\tau|}{|\mathcal{S}|} Q\left(S^\tau\right) = 0.5 * 1 + 0.25 * 0 + 0.25 * 0 = 0.5$ (the entropy term is $1, 0, 0$ respectively for Math, History and CS, which is easy to observe)

The information gain for Gender is $1 - 0.81 = 0.19$, while for Major, it is $1 - 0.5 = 0.5$. Therefore, Major is the better feature to predict the label "like".

Question 2.3: Purity Measures. Apart from the Gini Index and Entropy we have seen in the class, one might argue to use a misclassification error, i.e. $1 - \max(p_k)$, to build the tree. Can you think of the pros and cons of this measure?

Hint: think in terms of its computational cost and its effectiveness as a split criterion.

Solution:

Pros: Compare to Gini Index and Entropy, a misclassification error is more efficient (and easier to compute) because Gini Index involves computing multiplication and entropy involves computing logarithms.

Cons: The problem with misclassification error is that it does not provide as fine-grained differentiation as Gini or entropy, making it rarely used when building the tree. Specifically, it **only considers the most frequent class** in a node and **ignores the overall distribution of class probabilities**.

3 Random Forests

Question 3: You are designing a decision forest for a **multi-class classification problem** with t decision trees trained using bagging. Each tree uses simple weak learners that split the data using a single feature at a time. At each node, the tree randomly selects a subset of k features (out of d total features) and chooses the one that gives the best split according to an information gain criterion. The full dataset has n samples and d real-valued features. For tree i, a new training set $X^{(i)}$ is created using bagging.

Question 3.1: Bagging. Describe how the training set $X^{(i)}$ is constructed using bagging. Why is using *sampling with replacement* important? How should we handle duplicate data points?

Solution: To create $X^{(i)}$, we randomly sample n data points **with replacement** from the original training set X. Sampling with replacement allows some data points to appear multiple times while others may not appear at all. This introduces **variation** in the training data for each tree, helping to **reduce variance** when predictions from multiple trees are averaged.

Duplicate points are treated as independent entries. In split computations (e.g., entropy or Gini index), their influence is **weighted by their frequency** in the dataset. That is, if a point appears multiple times, it contributes proportionally more to impurity measures and decisions.

Question 3.2: Training time complexity. Fill in the blanks to derive the overall running time to construct a random forest using both bagging and random feature selection.

Let h be the depth (or height) of the deepest tree in the forest. You must use the tightest possible bounds in terms of n, d, t, k, h, and n'.

Consider choosing the best split at a tree node that contains n' sample points. We can choose the best split for these n' points in $O(\_\_\_\_)$ time. Therefore, the time per sample point in that node is $O(\_\_\_\_)$.

Each sample point in $X^{(i)}$ participates in at most $O(\_\_\_\_)$ nodes, so it contributes at most $O(\_\_\_\_)$ to the total time.
Therefore, the total time to train a single tree is $O(\_\_\_\_)$.
With t trees, the total time to train the entire forest is $O(\_\_\_\_)$.

Solution: Answer (blanks in order): $O(n'k)$, $O(k)$, $O(h)$, $O(kh)$, $O(nkh)$, $O(nkht)$

Question 3.3: Feature selection effect. Suppose instead of selecting k random features at each split, you always evaluate all d features. How does this affect the diversity of the trees in the forest? What might be a performance consequence?

Solution: Evaluating all d features makes the trees more similar, because the same dominant features are more likely to be chosen for splits across different trees. This **reduces diversity**, which is key to the ensemble's effectiveness. As a result, the forest may **overfit** or have **higher variance**, reducing the benefit of averaging over multiple models.

Question 3.4: Diversity and generalization. Random forests rely on the idea that averaging many uncorrelated models can improve generalization. Why is it important to both **bag the data** and **randomize the feature selection**? What could go wrong if only one of these techniques is used?

Solution:

- **Bagging** introduces variation by changing the training data per tree.
- **Random feature selection** introduces variation in the *splits* made within trees.
- If only bagging is used, trees may still select similar features and produce similar structures, especially if some features dominate.
- If only feature randomness is used without bagging, each tree sees the same data and may overfit in similar ways.
- Using both ensures **low correlation** among trees and maximizes the variance-reduction benefit of ensemble learning.

Question 4:
You are given a training dataset with 3 Boolean features X_1, X_2, and X_3, where $X_i \in \{0, 1\}$. The label is defined by the rule $Y = X_1 \vee X_2$, that is, $Y = 1$ if $X_1 = 1$ or $X_2 = 1$, and $Y = 0$ otherwise. The dataset contains all 8 possible combinations of these features:

X_1	X_2	X_3	Y
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	1

Question 4.1: Tree accuracy. What is the training error rate of a depth-0 decision tree on this dataset?

Solution: A depth-0 tree predicts only a single label for *every* example. Since 6 out of 8 training points have label $Y = 1$, the tree will predict 1 for all examples. This misclassifies exactly the 2 examples where $Y = 0$. Hence, the training error is $\frac{2}{8} = \frac{1}{4}$.

Question 4.2: Splitting by error rate. If your splitting criterion is *training error rate*, which feature (or features) would you choose to split on at the root? Briefly explain your answer.

Solution: All features are equally good. No matter which feature you choose at the root, the tree will still make 2 mistakes: 4 data points will go to the left and 4 to the right. Both child nodes will contain more 1's than 0's, so the best prediction in each case is 1. As a result, the tree will predict all examples as positive, just like the depth-0 decision tree.

Question 4.3: Splitting by information gain. If your splitting criterion is *information gain*, which feature (or features) would you choose to split on at the root? Briefly explain your answer.

Solution: Either X_1 or X_2 would be suitable. Feature X_3 has zero mutual information with the label Y, while both X_1 and X_2 yield nonzero information gain. Their mutual information is:

$$-\left(\frac{1}{4}\log\frac{1}{4}+\frac{3}{4}\log\frac{3}{4}\right)-\frac{1}{2}>0$$