NYCU Pattern Recognition, Homework 2

Deadline: Apr. 26, 23:59

Part. 1, Coding (80%):

For this coding assignment, you are required to implement the Decision Tree and Random Forest algorithms using only NumPy. Afterward, you will need to train your model on the provided dataset and evaluate its performance on the validation data.

(30%) Decision Tree

Requirements:

- Implement the **Gini** and **Entropy** for measuring the "best" splitting of the data.
- Implement the Decision Tree algorithm (CART, Classification and Regression Trees) with the following 3 arguments:
- **Criterion**: The function to measure the quality of a split of the data. Your model should support "gini" for the Gini impurity and "entropy" for the information gain.
- **Max_depth**: The maximum depth of the tree. If Max_depth=None, then nodes are expanded until all leaves are pure. Max_depth=1 equals splitting data once.
- **Max_features**: The number of features to consider when looking for the best split. If None, then max_features=n_features.
- For more detailed descriptions of the arguments, please refer to Scikit-learn.
- Your model should produce the same results when rebuilt with the same arguments, and there is no need to prune the trees.
- You can use the recursive method to build the nodes.

Criteria:

1. (5%) Compute the Entropy and Gini index of the array provided in the sample code, using the formulas on page 6 of the HW3 slide.
2. (10%) Show the accuracy score of the validation data using criterion='gini' and max_features=None for max_depth=3 and max_depth=10, respectively.
3. (10%) Show the accuracy score of the validation data using max_depth=3 and max_features=None, for criterion='gini' and criterion='entropy', respectively.
4. (5%) Train your model using criterion='gini', max_depth=10 and max_features=None. Plot the feature importance of your decision tree model by simply counting the number of times each feature is used to split the data.

(20%) Random Forest

Requirements:

- Fix the random seed.
- Implement the Random Forest algorithm by using the CART you just implemented.
- The Random Forest model should include the following three arguments:
- **N_estimators**: The number of trees in the forest.
- **Max_features**: The number of features to consider when looking for the best split using the decision tree.
- **Bootstrap**: Whether to use bootstrap samples when building trees.
- For more detailed descriptions of the arguments, please refer to Scikit-learn.
- Use majority voting to obtain the final prediction.

Criteria:

5. (10%) Show the accuracy score of the validation data using criterion='gini', max_depth=None, max_features=sqrt(n_features), and bootstrap=True, for n_estimators=10 and n_estimators=50, respectively.
6. (10%) Show the accuracy score of the validation data using criterion='gini', max_depth=None, n_estimators=10, and bootstrap=True, for max_features=sqrt(n_features) and max_features=n_features, respectively.

(20%) Train your own model

Requirements:

- Train your model (either Decision Tree or Random Forest).
- Try different parameters and feature engineering to beat the baseline.
- Save your test predictions in a CSV file.

Criteria:

7. (20%) Explain how you chose/design your model and what feature processing you have done in detail. Otherwise, no points will be given.

Points	**Testing Accuracy**
20 points	**acc > 0.915**
15 points	**acc > 0.9**
10 points	**acc > 0.88**
5 points	**acc > 0.8**
0 points	**acc <= 0.8**

Part. 2, Questions (30%):

1. Answer the following questions in detail:
   a. Why does a decision tree tend to overfit the training set?
   b. Is it possible for a decision tree to achieve 100% accuracy on the training set?
   c. List and describe at least three strategies we can use to reduce the risk of overfitting in a decision tree.

2. For each statement, answer True or False and provide a detailed explanation:
   a. In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor.
   b. In AdaBoost, weighted training error ε_t of the t_{th} weak classifier on training data with weights D_t tends to increase as a function of t.
   c. AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

3. Consider a data set comprising 400 data points from class C_1 and 400 data points from class C_2. Suppose that a tree model A splits these into (200, 400) at the first leaf node and (200, 0) at the second leaf node, where (n, m) denotes that n points are assigned to C_1 and m points are assigned to C_2. Similarly, suppose that a second tree model B splits them into (300, 100) and (100, 300). **Evaluate the <u>misclassification rates</u> for the two trees and hence show that they are equal**. Similarly, **evaluate the cross-entropy** $Entropy = -\sum_{k=1}^{K} p_k \log_2 p_k$ and **Gini index** $Gini = 1 - \sum_{k=1}^{K} p_k^2$ **for the two trees**. Define p_k to be the proportion of data points in region R assigned to class k, where k = 1, . . . , K.