NCTU Pattern Recognition, Homework 3

0886004 周芝妤

Part. 1, Coding (60%):

In this coding assignment, you are required to implement the decision tree, and random forest algorithm by using only NumPy, then train your model on the provided dataset and evaluate the performance on testing data. Find the sample code and data on the GitHub page https://github.com/NCTU-VRDL/CS_ILE5065/tree/main/HW3

Please note that only NumPy can be used to implement your model. You will get zero points by simply calling sklearn.tree.DecsionTreeClassifier. And note that all of the model accuracy scores should be over 0.9

1. (5%) Please compute the Entropy and Gini Index of the given array by the formula on the slides.
   Ans:
   Gini of data is 0.4628099173553719
   Entropy of data is 0.9456603046006402

2. (20%) Implement the Decision Tree algorithm (CART, Classification and Regression Trees) and train the model by the given arguments, and print the accuracy score on the test data. You should implement **two arguments** for the Decision Tree algorithm, 1) **Criterion**: The function to measure the quality of a split. Your model should support "gini" for the Gini impurity and "entropy" for the information gain.
   2) **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 to split data once
   - **2.1.** Using Criterion='gini' to train the model and show the accuracy score of test data by Max_depth=3 and Max_depth=10, respectively.
   - **2.2.** Using Max_depth=3 to train the model and show the accuracy score of test data by Criterion='gini' and Criterion='entropy', respectively.

   Note: You should get the same results when re-building the model with the same arguments, ***no need to prune the trees***

   Note: You can find the best split threshold by two methods. First one: 1) Try N-1 threshold values, where the i-th threshold is the average of the i-th and (i+1)-th sorted values. Second one: Use the unique sorted value of the feature as the threshold to split the data.

   Hint: You can use the recursive method to build the nodes

Ans 2.1 :

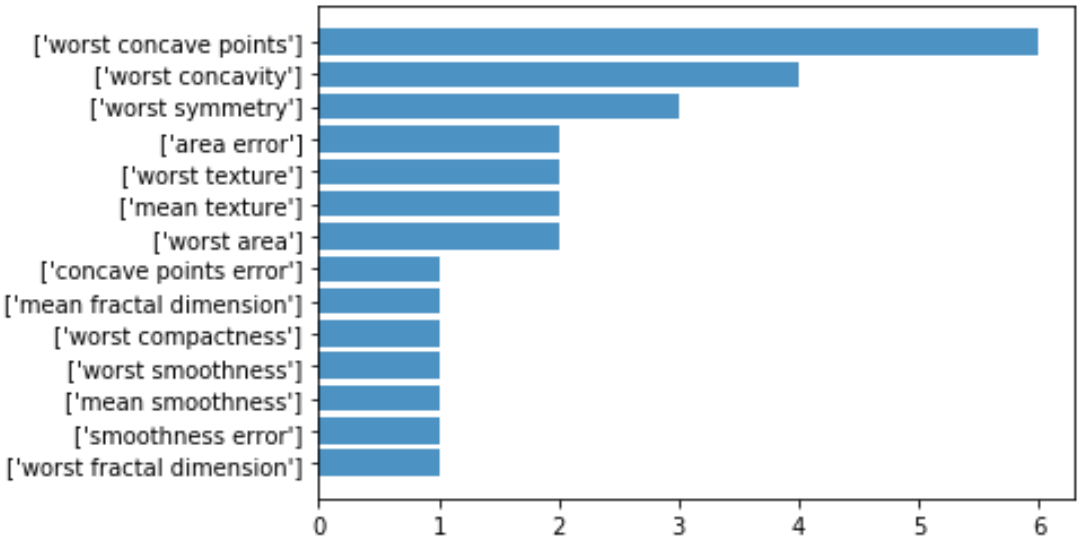
clf_depth3, criterion='gini', max_depth=3	clf_depth10 ,criterion='gini', max_depth=10
0.916083916083916	0.96875

Ans 2.2 :

clf_gini ,criterion='gini', max_depth=3	clf_entropy, criterion='entropy', max_depth=3
0.916083916083916	0.916083916083916

3. (15%) Plot the feature importance of your Decision Tree model. You can use the model for Question 2.1, max_depth=10. (You can simply count the number of a feature used in the tree, instead of the formula in the reference. Find more details on the sample code. (matplotlib is allowed to use)

   Ans: I just simply count the features used in Question 2.1, max_depth=10 tree.



4. (20%) Implement the random forest algorithm by using the CART you just implemented for Question 2. You should implement **three arguments** for the random forest model.
   1) **N_estimators**: The number of trees in the forest.
   2) **Max_features**: The number of features to consider when looking for the best split
   3) **Bootstrap**: Whether bootstrap samples are used when building trees

   4.1. Using Criterion='gini', Max_depth=None, Max_features=sqrt(n_features), Bootstrap=True to train the model and show the accuracy score of test data by n_estimators=10 and n_estimators=100, respectively.

   4.2. Using Criterion='gini', Max_depth=None, N_estimators=10, Bootstrap=True, to train the model and show the accuracy score of test data by Max_features=sqrt(n_features) and Max_features=n_features, respectively.

Note: Use majority votes to get the final prediction, you may get different results when re-building the random forest model

Ans 4.1 :

clf_10tree (n_estimators=10)	clf_100tree (n_estimators=100)
0.951048951048951	0.958041958041958

Ans 4.2 :

clf_random_features	clf_all_features
0.951048951048951	0.965034965034965

Part. 2, Questions (40%):

The answer of Part 2 is handed written and showed in following pictures.

1. (15%) By differentiating the error function below with respect to α_m,

$$E = e^{-\alpha_m/2} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in \mathcal{M}_m} w_n^{(m)}$$

$$= (e^{\alpha_m/2} - e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$$

show that the parameters α_m in the AdaBoost algorithm are updated using $\alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}$ in which ϵ_m is defined by $\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$.

2. (15%) 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 (300, 100) assigned to the first leaf node (predicting C_1) and (100, 300) assigned to the second leaf node (predicting C_2), where (n, m) denotes that n points come from class C_1 and m points come from class C_2. Similarly, suppose that a second tree model B splits them into (200, 400) and (200, 0), respectively. Evaluate the misclassification rates for the two trees and hence show that they are equal. Similarly, evaluate the pruning criterion $C(T) = \sum_{m=1}^{|T|} Q_m(T) + \lambda|T|$ for the cross-entropy case $Q_m(T) = \sum_{k=1}^{K} p_{mk} \ln(p_{mk})$ for the two trees and show that tree B is lower than tree A. Leaf nodes are indexed by $\tau = 1, \ldots, |T|$, with leaf node τ represents a region R_τ, and $p_{\tau k}$ is the proportion of data points in region R_τ assigned to class k, where $k = 1, \ldots, K$.
Hint: The answer should contain λ which is the regularization parameter.

3. (10%) Verify that if we minimize the sum-of-squares error between a set of training values $\{x_n\}_{n=1\sim N}$ (N is number of training data) and a single predictive value x, then the optimal solution for x is given by the mean of the $\{x_n\}_{n=1\sim N}$.

Part 2,

1. $E = (e^{\alpha_m/2} - e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$

E : error function w_n : weighting coefficients

I is the "Indicator function", when $y_m(x_n) \neq t_n$, $I = 1$

By differentiating the E with respect to α_m

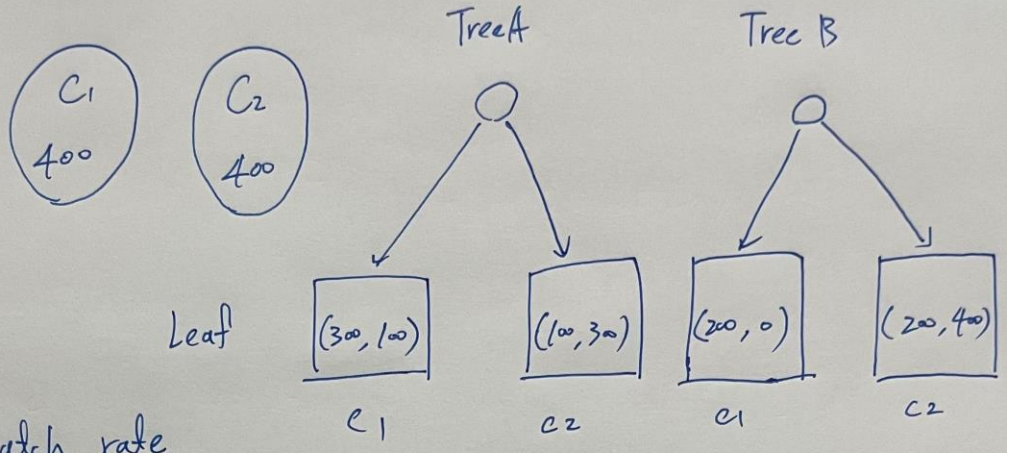
$$\frac{\partial E}{\partial \alpha_m} = \frac{1}{2}\left((e^{\alpha_m/2} + e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) - e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)} \right)$$

Setting to 0, $\frac{\sum_n w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_n w_n^{(m)}} = \frac{e^{-\alpha_m/2}}{e^{\alpha_m/2} + e^{-\alpha_m/2}} = \frac{1}{e^{\alpha_m} + 1}$

Cause we know $\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} \Rightarrow \varepsilon_m = \frac{1}{e^{\alpha_m} + 1}$

$$\Rightarrow e^{\alpha_m} + 1 = \frac{1}{\varepsilon_m} \Rightarrow e^{\alpha_m} = \frac{1 - \varepsilon_m}{\varepsilon_m} \Rightarrow \alpha_m = \ln\left\{ \frac{1-\varepsilon_m}{\varepsilon_m} \right\} \text{※}$$

2.



(a) mismatch rate for 2 trees :

$$M_A = \frac{100 + 100}{400 + 400} = \frac{1}{4} \qquad M_B = \frac{0 + 200}{400 + 400} = \frac{1}{4}$$

$M_A = M_B$ ※

(b) calulate pruning criterion.

* Cross-entrophy: -1.3862, -0.2876

$$C_{entrophy}(T_A) = -2\left(\frac{100}{400}\ln\frac{100}{400} + \frac{300}{400}\ln\frac{300}{400}\right) + 2\lambda \approx 1.1245 + 2\lambda$$

$$C_{entrophy}(T_B) = -\frac{200}{400}\ln\frac{200}{400} - \cancel{\frac{0}{400}\ln\frac{0}{400}} - \frac{200}{400}\ln\frac{200}{400} - \cancel{\frac{400}{400}\ln\frac{400}{400}} + 2\lambda$$

($\ln\frac{200}{400} \downarrow -0.6931$, $\ln\frac{400}{400} \downarrow 0$)

$$\simeq 0.6931 + 2\lambda$$

$C_{entrophy}(T_B) < C_{entrophy}(T_A)$ ※

3. Sum-of-squares error $E(t) = \frac{1}{2}\sum_{\{t_n\}}(t_n - t)^2$

Derivative to t and set it = zero :

$$\frac{dE}{dt} = -\sum_{\{t_n\}}(t_n - t) = 0 \Rightarrow t = \frac{1}{N}\sum_{\{t_n\}} t_n$$ ※

($N = |\{t_n\}|$, number of the values in $\{t_n\}$.)