IE 506: Machine Learning - Principles and Techniques Jan-May 2023

Programming Challenge Report

Name: Kanad Sen Roll No: 22M1674

1. Data Preprocessing

- First take the data input. Here I have created a f_L() function which takes inputs from the user about the dataset they want to use. After that read each line from the file and store it in a list. Pass this list to a different function for further preprocessing.
- In the max_class() function find and store the maximum number of labels possible for a particular datapoint to belong to.
- In the encode() function I have encoded the y variables have been one hot encoded the labels here. The labels which belong to the data point has been encoded as 1 and the rest are encoded as 0. So if the labels of a particular datapoint are [1,6,7] and maximum number of labels is 13 then the one hot representation is [1,0,0,0,0,1,1,0,0,0,0,0,0].
- After that we use X_features() function to separate the features from the extracted data.

2. Training Process

This code trains an AdaBoost algorithm using a Decision Tree as the weak learner. The AdaBoostMH class contains the training function **train** that takes in training and validation data as inputs. The code starts by initializing the number of epochs (iterations) as **epoch_num**, which is taken as user input. The **calculate_mult** function calculates the multiplication of the true and predicted label arrays and determines if the prediction is correct or incorrect based on a threshold of 0.8.

In the **train** function, the weights of each sample are initialized as 1/N where N is the number of data points. Then, the algorithm trains the Decision Tree using the current sample weights and calculates the predicted labels. It then calculates the training and validation accuracy and F1-score for each epoch, prints them out, and selects the best model based on the highest validation accuracy. The algorithm then calculates the error rate (r) by multiplying the true and predicted labels with the weights and sums them up. The alpha value for each epoch is then calculated using the error rate. The weights are updated based on the alpha value and the calculated multiplier. The **train** function saves the best model in a file using the pickle module.

The training process of the AdaBoost algorithm using a Decision Tree as the weak learner involves the following steps:

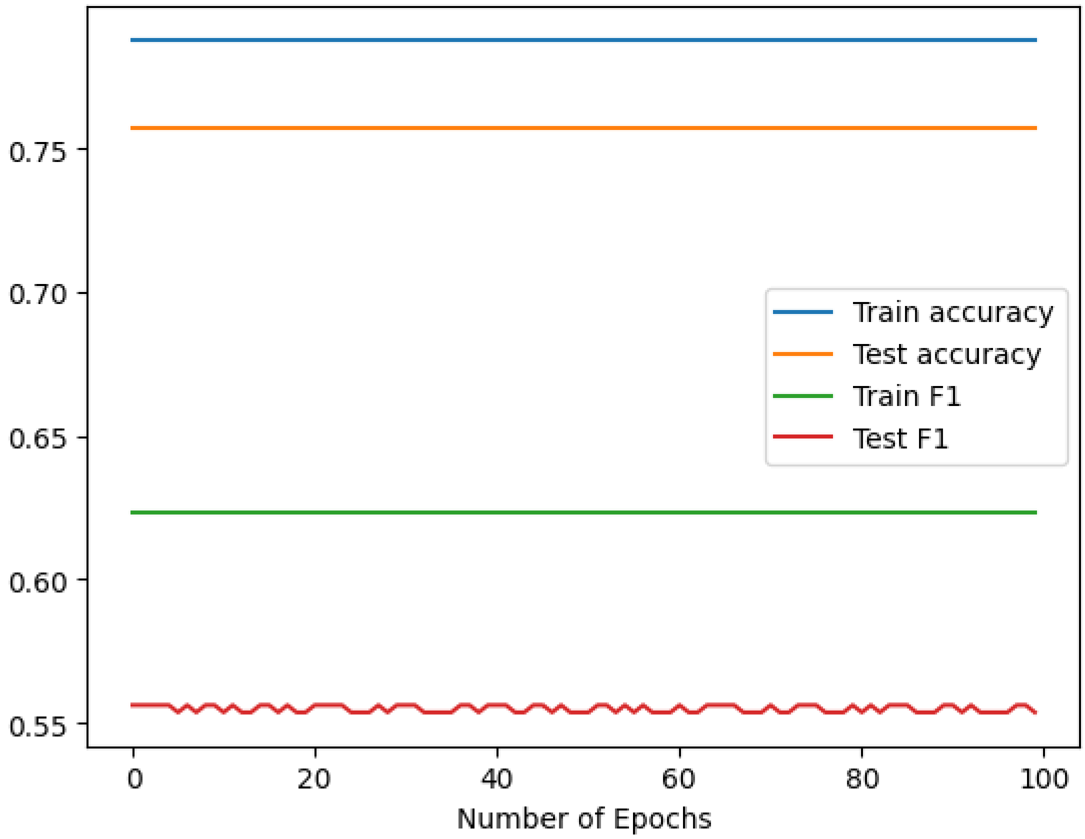
1. Initialization: The number of epochs (iterations) is initialized by taking user input. The weights of each sample are initialized as 1/N where N is the number of data points.
2. Weak learner training: The algorithm trains the Decision Tree using the current sample weights. The **fit** method of the **DecisionTreeClassifier** is used to train the Decision Tree.
3. Prediction: The predicted labels are calculated using the trained Decision Tree. The **predict** method of the **DecisionTreeClassifier** is used to obtain the predicted labels.
4. Accuracy and F1-score calculation: The algorithm calculate the training and validation accuracy and F1-score for each epoch. The accuracy is calculated as the sum of correct predictions divided by the total number of predictions. The F1-score is a weighted average of precision and recall, where precision is the ratio of true positives to the sum of true positives and false positives, and recall is the ratio of true positives to the sum of true positives and false negatives.
5. Best model selection: The algorithm selects the best model based on the highest validation accuracy. If the current validation accuracy is higher than the previous best validation accuracy, the current model is saved as the best model.
6. Error rate calculation: The error rate (r) is calculated by multiplying the true and predicted labels with the weights and summing them up.
7. Alpha calculation: The alpha value for each epoch is calculated using the error rate. The alpha value is calculated as 0.5 times the natural logarithm of (1+r) divided by (1-r).
8. Weight update: The weights are updated based on the alpha value and the calculated multiplier. The updated weights are calculated as the exponential of (-alpha times the multiplication of true and predicted labels) divided by the sum of the new weights.
9. Iteration: The algorithm repeats the above steps for the specified number of epochs.
10. Best model saving: The best model is saved in a file using the pickle module.

Overall, the training process involves iteratively updating the weights of the samples and combining the predictions of multiple weak learners to improve the accuracy of the model. The AdaBoost algorithm learns to assign higher weights to the misclassified samples and lower weights to the correctly classified samples, thus increasing the importance of the former in the subsequent iterations.

3. Hyperparameter Tuning

In the train algorithm I have used Hyper parameter tuning for my Decision tree and have found that Decision tree of max_depth=5 performs the best.

4. Results:



In the above results we don't see much changes for depth=5. Minor changes have been observed in the accuracy and F1 values. On the given test set I have got the following values for accuracy and F1 score:

```
Accuracy on the given Test Set : 0.7707692307692311

F1 on the given Test Set : 0.5940054495912799
```