Week 7 - Adaboost

MACHINE INTELLIGENCE LABORATORY

Teaching Assistants
Sarasharish
Abaksy
Vighneshkamath

An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together .To make more accurate predictions than any individual model. One such algorithm is Adaboost.

In week 7 you are required to code for the class AdaBoost which will implement Adaboost algorithm.

Your task is to complete the code for the class AdaBoost and its methods

You are provided with the following files:

1. **Week7.py**
2. **SampleTest.py**

Note: These sample test cases are just for your reference.

SAMPLE TEST CASE Data

This is a randomly generated test case and can be analyzed by opening sampleTest.py

Important Points:

1. Please do not make changes to the function definitions that are provided to you. Use the skeleton as it has been given. Also do not make changes to the sample test file provided to you. Run it as it is.
2. You are free to write any helper functions that can be called in any of these predefined functions given to you. Helper functions must be only in the file named 'YOUR_SRN.py'.
3. Your code will be auto evaluated by our testing script and our dataset and test cases will not be revealed. Please ensure you take care of all edge cases!
4. **The experiment is subject to zero tolerance for plagiarism. Your code will be tested for plagiarism against every code of all the sections and if found plagiarized both the receiver and provider will get zero marks without any scope for explanation.**
5. **Kindly do not change variable names or use any other techniques to escape from plagiarism, as the plagiarism checker is able to catch such plagiarism**
6. Hidden test cases will not be revealed post evaluation.

week4 .py

- ✓ You are provided with structure of class AdaBoost.
- ✓ The class AdaBoost contains one constructor and 6 methods out of which two is already written.
- ✓ Your task is to write code for the four of these 6 methods.
- ✓ Do not make any changes to the already written methods

- ■ def__init_(self, n_stumps=20)
  - ◆ This initialises a number of decision stumps that you will be using and a an empty list to store these stumps

- def fit(self, X, y)
  - This is fit function that will be finally called to train the adaboost algorithm
  - This creates Decision Tree Classifier with given parameter appends it to the stump list
  - This also calls the method stump_error,compute_alpha,update_weights which you are required to implement further

- def stump_error(self, y, y_pred, sample_weights)
  - This takes the actual value and the predictions along with the weights assigned to each sample
  - Returns the stump error

- def compute_alpha(self, error)
  - The functions takes the error of the decision stump and computes the value alpha, that is the weight the stump has in the final prediction.
  - Use numerical stability of 1e-9 so that you don’t get zero division error.

- def update_weights(self, y, y_pred, sample_weights, alpha)
  - The function takes the true output and prediction along with the sample weight ans alpha value
  - Returns the updates sample weights which is normalized.

- def predict(self, X)
  - Predict the output of sample input X using the Adaboost model

USE THE BELOW MENTIONED FORMULA TO UPDATE WEIGHTS AND ALPHA

$$\alpha_t \leftarrow \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t} \; ;$$

$$w_i^{s+1} = \frac{w_i^s}{N^s} \cdot e^{-\alpha_s}$$ For correctly classified samples (we ***reduce*** their weight)

$$w_i^{s+1} = \frac{w_i^s}{N^s} \cdot e^{+\alpha_s}$$ For incorrectly classified samples (we ***increase*** their weight)

1. **You may write your own helper functions if needed**
2. **You can import libraries that come built-in with python 3.7**
3. **You cannot change the skeleton of the code**
4. **Note that the target value is an int**

SampleTest.py

1. This will help you check your code.
2. Make sure you have installed numpy and sklearn
3. Passing the cases in this does not ensure full marks, you will need to take care of edge cases
4. Name your code file as YOUR_SRN.py
5. Run the command

python3 SampleTest.py --SRN YOUR_SRN

if import error occurs due to any libraries that is mentioned in the skeleton code try:

python3.7 SampleTest.py --SRN YOUR_SRN