adaboost

Submitters:

Yishay Seroussi 305027948, Samuel Bismuth 342533064.

Python version:

3.9

Configuration

This repository includes an implementation of the adaboost alogithm in python.

We use the docker environement. Make sure docker is installed in you machine. That is the only dependency of the project.

According to your distribution, run:

```
sudo <yum/apt-get> install -y docker
```

Then, to run the script run:

```
bash start.sh
```

To enter in the container terminal (only for developement purpose):

```
bash bash.sh
```

If you don't want to use docker, you are able to run the code in any machine by folowing the next steps:

Install python3.9.

Install numpy by running:

```
pip3 install numpy
```

Run the main python file:

```
python3 main.py
```

Code structure:

The code is composed of three folders.

- The data folder containing the two txt files of data received for the assignment.
- The packages folder containing the requirement txt file with the pip lib we used (numpy).
- The src folder containing the code source.
    - main.py -> The main file of the code. This is our entrypoint. This is also the file were the prints are done.
    - data.py -> The file handle the txt data to convert it into objects.
    - geometry.py -> Here the classes Line and Point are defined.
    - model.py -> Here the Feature and Label classes are defined.
    - rule.py -> Here the Rule class is defined.
    - adaboost.py -> Here you can find the main algorithm with the computation of the final accuracy and error.

Example of outputs:

Hc Body Temperature

rule: -1.1111111111111216x + 187.7777777777788, rule weight: 0.3698335980974191
rule: -14.999999999999787x + 1545.499999999979, rule weight: 0.22246613151532224
rule: 14.166666666666632x + -1307.9166666666633, rule weight: 0.18620726376999933
rule: -189.9999999999838x + 18811.999999998403, rule weight: 0.18383602002155972
rule: -1.2500000000000044x + 189.50000000000045, rule weight: 0.18335340097559974
rule: 10.000000000000213x + -910.000000000021, rule weight: 0.17967714682327982
rule: 1.4285714285714373x + -68.71428571428655, rule weight: 0.17014897895062175
rule: 2.0x + -119.19999999999999, rule weight: 0.16607354831318452

##

Hc Body Temperature ACCURACY

hc body temperature train accuracy: 0.6923076923076923
hc body temperature test accuracy: 0.5538461538461539

##

########## Iris ####################

rule: -0.33333333333333237x + 5.999999999999997, rule weight: 1.4615807903595777
rule: -2.333333333333332x + 11.399999999999997, rule weight: 0.1315670898877912
rule: 12.000000000000036x + -27.500000000000092, rule weight: 0.022055062605134967
rule: 0.7142857142857145x + 3.1714285714285704, rule weight: 0.021889187628204295
rule: 0.3333333333333348x + 3.8666666666666623, rule weight: 0.021579059587843164
rule: x = 2.8, rule weight: 0.0
rule: 5.999999999999996x + -12.799999999999986, rule weight: 0.0
rule: 3.9999999999999956x + -7.199999999999987, rule weight: 0.0

##

########## Iris ACCURACY ####################

iris train accuracy: 0.9
iris test accuracy: 0.88

##

Here we printout the accuracy. To get the error, we only have to calculate 1 - accuracy.

Do you see overfitting?

Let first focus on the Hc Body Temperature data set. The error of the train is usually about 30% which is much lower than the error we get in the test. The error of the test is usually about 45%. The test error / train error ~= 1.5 Adding to the fact that we have 8 rules, and by the theorem of Union of hypotheses, the vc dimension is high and equal to 2*3*8*log(24).

That is, we can say that there is overfitting.

Regarding the Iris data set. The error of the train is usually about 10% which is almost equal to the error we get in the test. The test error / train error ~= 1

The result with this data set is also much better that the one we have with the Hc Body Temperature data set. That's why, there is no reason to conclude that there is overfitting with the Iris data set.

Work division

We worked on this code together using one computer as a pair programming. That is, we've handle and understand together the complexity of the adaboost implementation and the code design in python. There is nothing in this work that have been done only by one submitter. Notice that we worked only on one github account since we used only one computer.