MSDS630 HW3: Boosting

Yannet Interian
MS in Data Science

February 2, 2024

1 AdaBoost on a toy dataset (5 points)

We will apply AdaBoost to classify a toy dataset. The dataset consists of 4 points: $(x^{(1)} = (0, -1), y^{(1)} = -1), (x^{(2)} = (1, 0), y^{(2)} = 1), (x^{(3)} = (-1, 0), y^{(3)} = 1)$ and $(x^{(4)} = (0, 1), y^{(4)} = -1)$. You may want to use Python as a calculator rather than doing the computations by hand, but you don't have to submit your code.

1. (3 points) For M = 4 (use 4 trees), show how Adaboost works for this dataset, using simple decision stumps (depth-1 decision trees that simply split on a single variable once) as weak classifiers. For each timestep fill the following table:

m	w_1	w_2	w_3	w_4	err	α	$T_m(x^{(1)})$	$T_m(x^{(2)})$	$T_m(x^{(3)})$	$T_m(x^{(4)})$
1	1	1	1	1	$1/4$	$\ln(3)$	-1	$+1$	-1	-1
2	1	1	3	1	$1/6$	$\ln(5)$	-1	$+1$	$+1$	$+1$
3	1	1	3	5	$1/10$	$\ln(9)$	-1	-1	$+1$	-1
4	1	9	3	5	$1/18$	$\ln(17)$	$+1$	$+1$	$+1$	-1

2. (1 points) What is the training error of AdaBoost for this toy dataset? (show me the computation)

3. (1 points) Is the above dataset linearly separable? Explain why AdaBoost does better than a decision stump on the above dataset.

2. $f(x_1) = \text{Sign}(-\ln(3) - \ln(5) - \ln(9) \pm \ln(17)) = \text{Sign}(-2.07) = -1$ ✓

$f(x_2) = \text{Sign}(\ln(3) + \ln(5) - \ln(9) + \ln(17)) = \text{Sign}(3.34) = 1$ ✓

$f(X_3) = \text{Sign}(-\ln(3) + \ln(5) + \ln(9) + \ln(17)) = \text{Sign}(1.14) = 1$ ✓

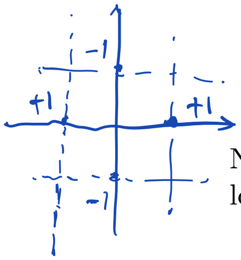
$f(x_4) = \text{Sign}(-\ln(3) + \ln(5) - \ln(9) - \ln(17)) = \text{Sign}(-4.52) = -1$ ✓

Error $= 0$ because predictions are correct.

3.

We cannot draw a line to separate +1 from -1, thus it is not linearly separable.

Since Adaboost is an ensemble model, which combines multiple weak learners, it is more flexible than a decision stump.



Note: In the Adaboost algorithm all log functions are **natural** log and not log 10.

2 Implement AdaBoost (10 points)

For this exercise, you will implement AdaBoost from scratch and apply it to a spam dataset. You will be classifying data into spam and not spam. You can use the `DecisionTreeClassifier` from `sklearn` (with default `max_depth=1`) to learn your base classifiers. Write a program that implements AdaBoost with trees using the provided template and tests in `adaboost.py`. Here is how you train a decision tree classifier with weights."

```
h = DecisionTreeClassifier(max_depth=1, random_state=0)
h.fit(X, Y, sample_weight=w)
```

2.1 Deliverables

Use your code to populate `adaboost.py`. After you are done make sure you can run:
`pytest  test_adaboost.py`

Use a notebook to submit your experiments.

Please see notebook

3 Implement Gradient Boosting for MSE (15 points)

Implement gradient boosting for "rent-ideal.csv" dataset.

1. (10 points)Write a program that implements gradient boosting for MSE loss. Use the template given on `gradient_boosting_mse.py`

2. (2 points) Fix the shrinkage to 0.1. Apply gradient boosting to your dataset using different values for the number of trees $numTrees$. How do you find the best value for $numTrees$? Report train and validation R^2 for the best value of $numTrees$. Make a plot that shows your experiment (training and validation metric as a function of the number of trees). Try as least 2000 trees.

3. (3 points) Compare your results with the results of running the gradient boosting package (XGBoost). Explore the hyper parameters given in the package. Make plots or tables that show the result of your experiments.

3.1 Deliverables

Use your code to populate `gradient_boosting_mse.py`. After you are done make sure you can run:

```
pytest  test_gradient_boosting_mse.py
```

Use a notebook to submit your experiments.