Mathematics behind AdaBoost method

quantra

Contents

In the previous video, we learned about the sequential ensemble methods and implementation of AdaBoost method. In this document, we will walk you through the math behind AdaBoost method.

AdaBoost method steps:

1. Let's say we have 5 (n) items in the training dataset. We assign equal weight to all the items such that the weight assigned to the ith item (Wi) is 1/n = 1/5 = 0.20.

2. Then we generate a random subset with replacement from the training dataset. For the first iteration the weights are equal and therefore, all the items have an equal probability of appearing in the subset.

3. Fit the model on the random subset generated in the above step.

4. We re-assign weights to each item in the training dataset so that the items which were misclassified in previous step, 3, are given higher weights. This may result in improving the prediction on these items.
   The weights are re-assigned using the formula:

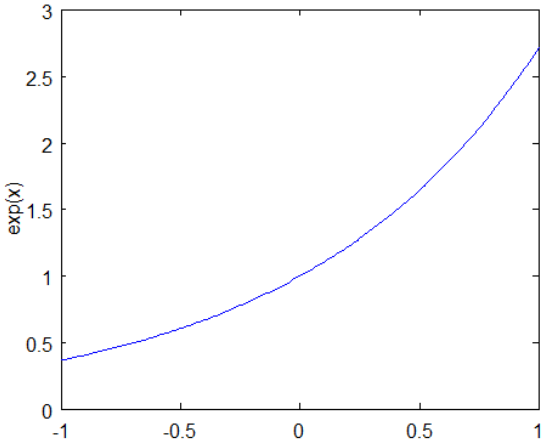
$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Where,

Wt (i) is the weight of the ith item
Wt+1 (i) is the new weight of the ith item
Zt is the sum of weights of all the items
Alpha t is the weight of the model trained
Y (i) is the actual output
H(X (i)) is the predicted output

The exponential function works as follows:

1. When the input value passed to it is positive, it returns a value greater than 1
2. When the input value passed to it is negative, it returns a value between 0 and 1



Let's say we have binary classification model with the output as +1 and -1. When the actual y(i) and predicted h(x(i)) output is same (Case I and IV in the table below) then y(i) * h(x(i)) is positive and when they are different (Case II and Case III in the table below) then y(i) * h(x(i)) is negative (-1).

	Actual Output y(i)	**Predicted Output h(x(i))**	**y(i) * h(x(i))**	**Comments**
Case I	1	1	1	
Case II	-1	1	-1	-1 indicates that the item is misclassified and therefore, will be given higher weight.
Case III	1	-1	-1	
Case IV	-1	-1	1	

This is then multiplied by the -1 * model weight (alpha) to give misclassified items higher weight and vice-versa. The alpha term is included in the equation to give misclassification by a strong model more weight compared to a weak model.

5. The steps 2 to 4 are repeated to train N models.
6. Create a final prediction model using N training models:

The weighted majority vote (or average) prediction of all the models is used to determine final prediction.

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Where,

H(x) is the output predicted by boostingmethod
T is the number of models
Alpha t is the t^{th} model weight
$h_t(x)$ is the output from the t^{th} model

In the upcoming unit, we will code the AdaBoost method in the python notebook.