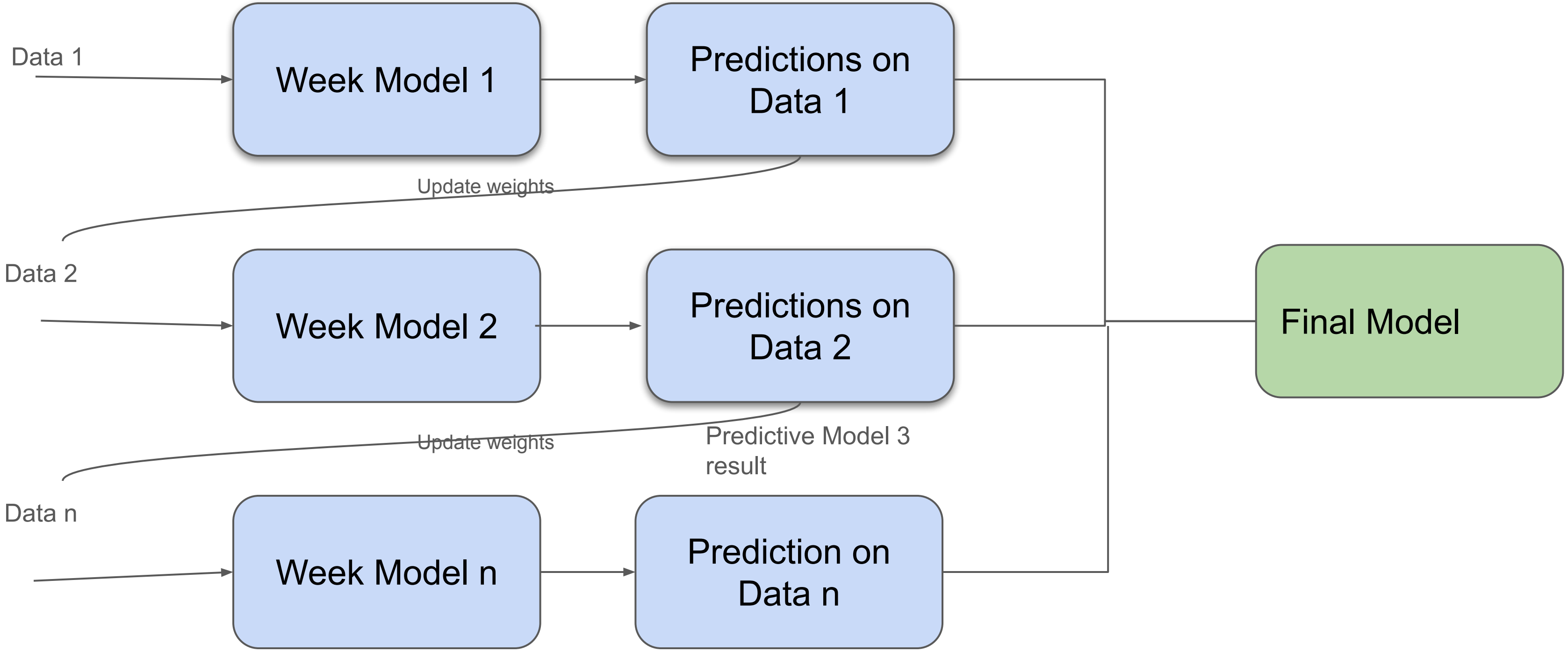
AdaBoost Algorithms

Detailed presentation

What is AdaBoost (Adaptive Boosting) Algorithms



Week Model 1,2,..n are individual models (mostly decision tree stumps which has max_depth=1)

Detailed Steps

Steps:

1. Initially assign same weights to each record in the dataset.
    a. **Sample weight = 1/N**
        i. Where N = Number of records
2. Get random samples and replace the original data with the probabilities equal to the sample weights
    a. Fit the model to the random samples and predict the classes for the original data. Here model is Decision stumps which has one node and two leaves
3. Calculate Total Error
    a. Total error is nothing but the sum of weights of misclassified record.

    b. **Total Error** = **Weights of wrongly predicted records i.e sum(wrongly predicted data * sample weights)**

    c. Total error will be always between 0 and 1.

    d. 0 represents perfect stump (correct classification)

    e. 1 represents weak stump (misclassification)

Detailed Steps

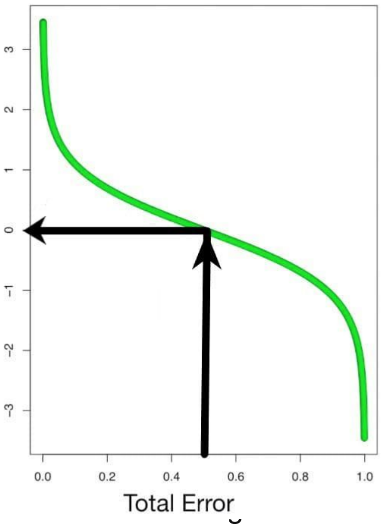
Steps:

4. **Calculate Performance of the Stump**

   a. Performance of the stump(α) = ½ log (1 – Total error/Total error)
   b. If the total error is 0.5, then the performance of the stump will be zero.
   c. If the total error is 0 or 1, then the performance will become infinity

      or -infinity respectively.

   d. If the performance high , then stump did the good job and if the performance is low, then performance stump(α), we can increase the weights of the wrongly classified records and correctly classified records.



Detailed Steps

Steps:

5. Update weights

a. Based on the performance of the stump(α) update the weights
b. New weight = Weight * $e^{(performance)}$ → wrongly predicted records

c. New weight = Weight * $e^{-(performance)}$ → correctly predicted records

d. Wrongly predicted: e^(performance)

    i. When the performance is relatively large, the last stump did good job and new sample weight is high as compared to old stumps.

    ii. When the performance is relatively low, the last stump did not do good job and new sample weight is little bit higher as compared to old stumps

Detailed Steps

Steps:

5. Update weights

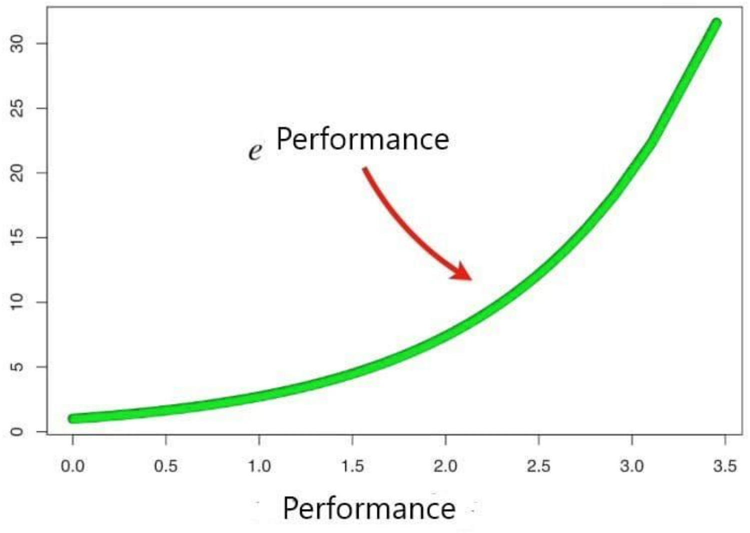
e. Correctly predicted:e^-(performance)

i. When the performance is relatively large, the last stump did good job and new sample weight is small as compared to old stumps.

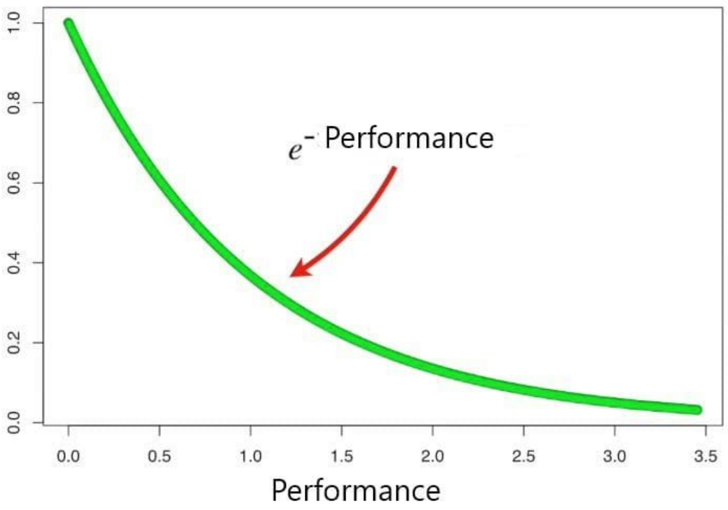
ii. When the performance is relatively low, the last stump did not do good job and new sample weight is little bit smaller as compared to old stumps

Detailed Steps

Performance of wrong prediction



Performance of right prediction



Detailed Steps

Steps:

6. Normalize weight

1) The initial sum sample weights is equal to 1 so the updated weight should not be zero so we need to divide the weight with sum of updated weights
    a) **Normalize weight = weight/sum(weight)**

7. Update weights in iteration

1. Go to the second stump and update the weight so that wrongly predicted records have high weights and will get higher probability of getting selected

8. **Repeat the steps 2 to 5 until we reach the 100% accuracy or the n_estimators are exhausted**

9. Final prediction

Final prediction = $\sum (\alpha_i$* (predicted value at each iteration)). Here α_i is performance values of stump