Boosting

Vishnuram Ayyavu Vijayakumar

Boosting

- Boosting refers to any procedure that combines many weak learners to yield a much higher performance. Unlike bagging, the base learners in boosting are sequentially generated by focusing on examples misclassified by earlier weak learners in the chain.

Weak Learners

A weak learner is an algorithm that performs only slightly better than random guessing on a given problem, i.e. slightly better than flipping a coin. It typically has low complexity and tends to underfit the data.

Let $\delta, \epsilon \in (0, 1)$, and let $\gamma \in (0, 1/2)$.

δ - tolerance of uncertainty
ϵ - tolerance of inaccuracy
γ - achievable improvement in accuracy

$$\mathbb{P}\left(\text{Accuracy}(f) > \frac{1}{2} + \gamma\right) \geq 1 - \delta.$$

Strong Learners

A strong learner, on the other hand, is an algorithm that can learn complex patterns in the data and make highly accurate predictions with very low error rates. It typically has high complexity and can overfit the data if not properly regularized.

Let $\delta, \epsilon \in (0, 1)$, and let $\gamma \in (0, 1/2)$.

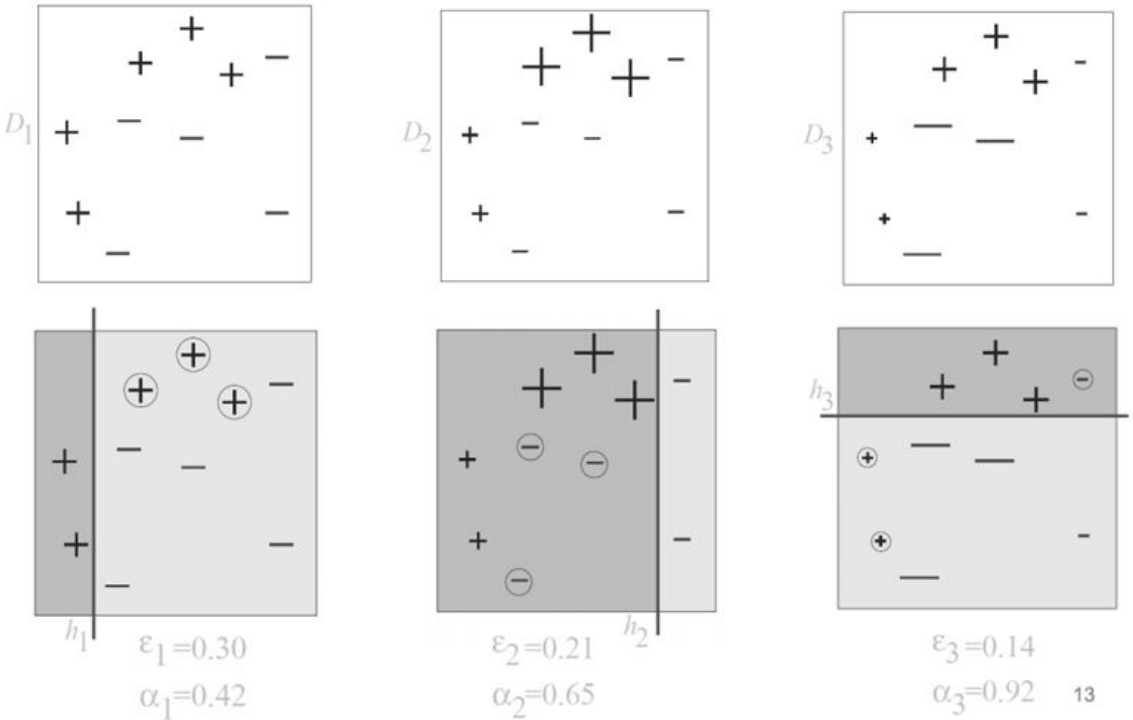
δ - tolerance of uncertainty
ϵ - tolerance of inaccuracy
γ - achievable improvement in accuracy

$$\mathbb{P}\big(\text{Accuracy}(f) > 1 - \epsilon\big) \geq 1 - \delta$$

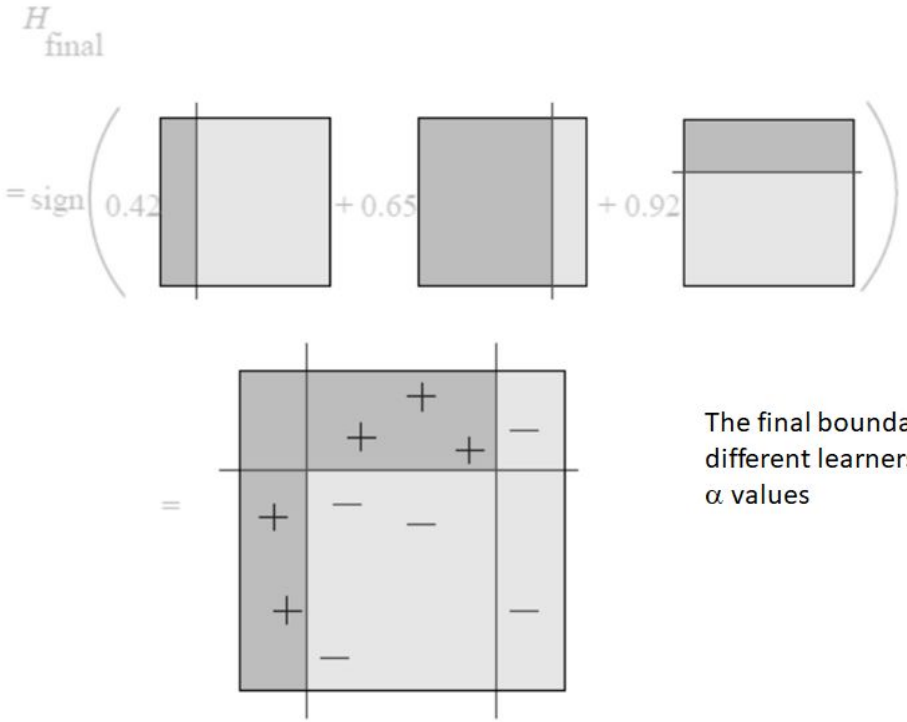
Adaboost

- Each training example is assigned a weight. The weight determines the training example's probability of being selected for training by the next base learner.
- Initially, all examples have identical weight. The misclassified examples see their weight go up to increase their selection chance for training
- The outputs of the learners are combined using weights to yield the final response

Visual Illustration of AdaBoost



Visual Illustration of AdaBoost



The final boundary is a weighted linear sum of different learners weighted by the respective α values

Gradient Boosting Tree

- The gradient boosting is another way of building a sequence of trees whose outputs are added to obtain the final prediction.
- The sequence of trees are really stumps or trees with depth of 2.
- Unlike the AdaBoost where the training examples vary in their importance, the successive trees in the gradient boosted algorithm are generated by minimizing a loss function.

Steps for Constructing an Ensemble of GBTs

- **Step1**: Construct a base tree with single root node. It is the initial guess for all the samples.
- **Step2:** Build a tree from errors of the previous tree.
- **Step3**: Scale the tree by learning rate (value between 0 and 1). This learning rate determines the contribution of the tree in the prediction
- **Step4**: Combine the new tree with all the previous trees to predict the result and repeat step 2 until maximum number of trees is achieved or until the new trees don't improve the fit.
- The final prediction model is the combination of all the trees.

Gradient boosting tree Illustration

We will first look at the regression problem consisting of 3 predictors and one output variable.

Height	Age	Gender	Weight
5.4	28	Male	88
5.2	26	Female	76
5	28	Female	56
5.6	25	Male	73
6	25	Male	77
4	22	Female	57

We will construct our first tree with only one output node. We will use the average of the Weight column as the output of this node. The average is 71.2.

GBT Illustration

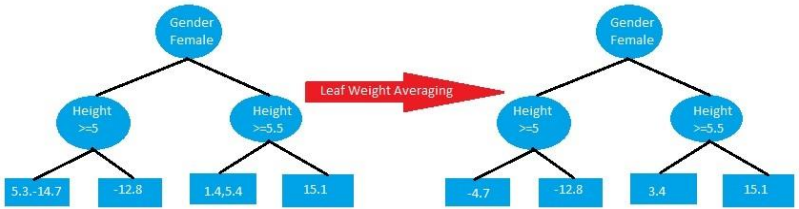
Step2 is to build a tree based on errors from previous tree. The errors that the previous tree made is the difference between the actual weight and the predicted weight. This difference is called residual or pseudo residual.

Height	Age	Gender	Weight	Predicted Weight 1	Pseudo Residuals 1
5.4	28	Male	88	71.2	88-71.2 = 16.8
5.2	26	Female	76	71.2	76-71.2 = 4.8
5	28	Female	56	71.2	56-71.2 = -15.2
5.6	25	Male	73	71.2	73-71.2 = 1.8
6	25	Male	77	71.2	77-71.2 = 5.8
4	22	Female	57	71.2	57-71.2 = -14.2

GBT Illustration

Step 3 is scaling tree with learning rate. Assuming the learning rate as 0.1.
Step 4 is combining the trees to make the new prediction. So, we start with initial prediction 71.2 and run the sample data down the new tree and sum them. Next, get the new set of values for the residuals.

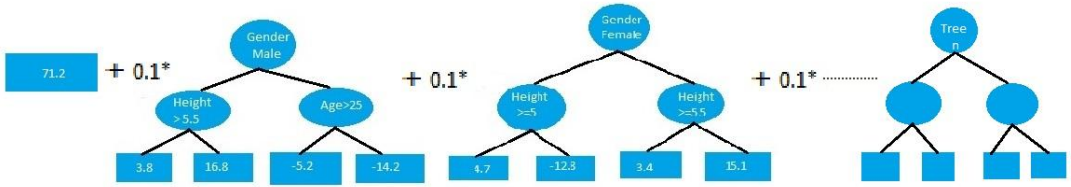
Height	Age	Gender	Weight	Predicted weight 2	Pseudo Residuals2
5.4	28	Male	88	71.2+0.1*16.8=72.9	88-72.9 = 15.1
5.2	26	Female	76	71.2+0.1*(-5.2)=70.7	76-70.7 = 5.3
5	28	Female	56	71.2+0.1*(-5.2)=70.7	56-70.7 = -14.7
5.6	25	Male	73	71.2+0.1*3.8=71.6	73-71.6 = 1.4
6	25	Male	77	71.2+0.1*3.8=71.6	77-71.6 = 5.4
4	22	Female	57	71.2+0.1*(-14.2)=69.8	57-69.8 = -12.8



GBT Illustration

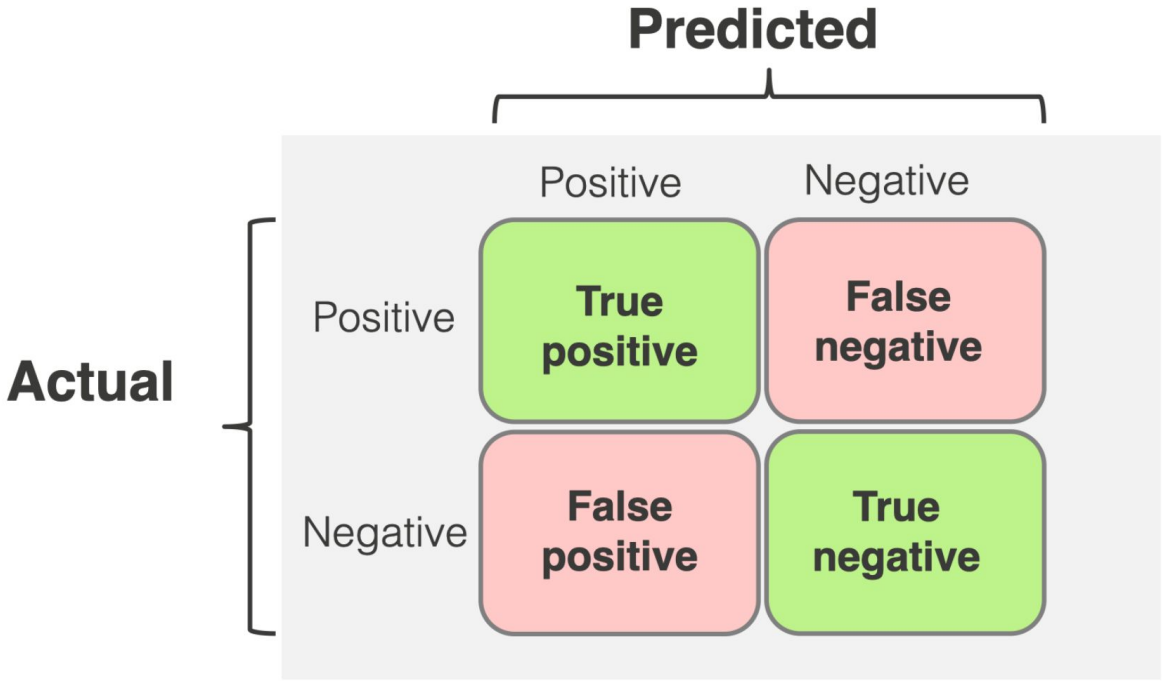
Now we combine the new tree with all the previous trees to predict the new weights and get the residuals and a new tree.

Height	Age	Gender	Weight	Predicted Weight 3
5.4	28	Male	88	71.2+0.1*16.8+0.1*15.1 = 74.4
5.2	26	Female	76	71.2+0.1*(-5.2)+0.1*(-4.7) = 70.2
5	28	Female	56	71.2+0.1*(-5.2)+0.1*(-4.7) = 70.2
5.6	25	Male	73	71.2+0.1*3.8+0.1*3.4 = 71.9
6	25	Male	77	71.2+0.1*3.8+0.1*3.4 = 71.9
4	22	Female	57	71.2+0.1*(-14.2)+0.1*(-12.8) = 68.5



Continuing this process to meet the desired goal, the process will terminate at some point.

Confusion matrix



Evaluation methods

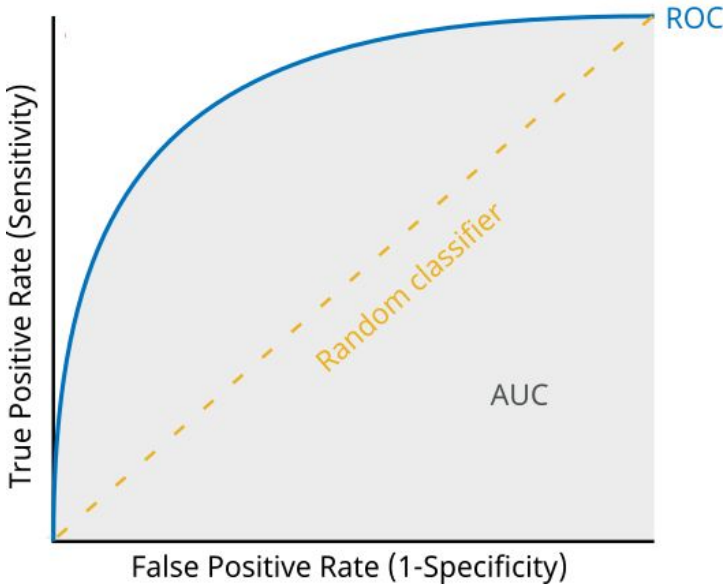
$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{n}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$



The more the AUC, the better the classification model.