Gradient Boosting

1 Input

We define our input data as the set $\{(x_i, y_i)\}_{i=1}^n$ where x_i denotes a row of predictors, and y_i denotes the value we are trying to predict for that row. n refers to the number of rows of data we have.

We also have a loss function $L(y_i, F(x))$ which evaluates how well we can predict y. $F(x)$ is a function that predicts the predicted values, and y is the observed value. The loss function most commonly used for gradient boost regression is

$$\frac{1}{2}(Observed - Predicted)^2$$

This is the same function we use for linear regression except with the $\frac{1}{2}$ in front. The reason we use this formula is that when we differentiate the formula with respect to "predicted" and use the chain rule, it cancels out with the squared. The actual value doesn't matter as long as it is the same for all predictions.

2 Step 1

Firstly we initialise the model with a constant value determined by the formula

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$$

$L(y_i, \gamma)$ is just the loss function, with γ referring to the predicted values. The summation means we add up one loss function for each observed value in the dataset. The $\operatorname{argmin}_\gamma$ means we need to find a predicted value that minimises the summation. In other words if we plot the y variables on a line, we want to find the point on the line that minimises the squared residuals divided by 2 (the loss).

To find the value that minimises this, we calculate the derivative of each term with respect to predicted using the chain rule.

$$\frac{d}{dPredicted}\frac{1}{2}(Observed - Predicted)^2 = -(Observed - Predicted)$$

We then add these together and solve for 0.

$$-(Observed - Predicted)_i + ... + -(Observed - Predicted)_n = 0$$
$$n \times Predicted = Observed_i + Observed_{i+1} + ... + Observervered_n$$
$$Predicted = \frac{\sum_{i=1}^{n} Observed_i}{n}$$

Which just gives us the average of all the observed values. This is the initial leaf we use to start the algorithm.

3 Step 2

Secondly we create a set amount M of trees. In practice, most people set M to be 100. m refers to an individual tree, so when $m = 1$ we are talking about the first tree.

For $m = 1$ to M:

(A) Compute

$$r_{im} = -[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}]_{F(x)=F_{m-1}(x)} \text{ for } i = 1, ..., n$$

$[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}]$ is just the derivate of the loss function with respect to the predicted value which we have already calculated to be $\frac{1}{2}(Observed - Predicted)^2$
We are multiplying the derivate by -1 which just leaves us with (Observed - Predicted) aka the residual.

$F(x) = F_{m-1}(x)$ is just the predicted value for the previous tree in the sequence. Since $m = 1$, this is just $F_0(x)$ which we have already calculated as the average of the observed values. r_{im} denotes the pseudo residual for the given sample i and given tree m. This is calculated for all values in the dataset.

(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm}, for $j = 1...J_m$

This is just telling saying to create a regression tree to predict the residuals, and use the values as leaves or terminal regions in the tree.

(C) For $j = 1...J_m$ compute

$$\gamma_{jm} = \underset{\gamma}{\text{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$

This means for every leaf in the tree, we compute the output value γ_{jm} which is just the value that minimises the summation of loss functions for the residuals in that output leaf. Similar to how we have computed the initial prediction, this value is just the average of the residuals in that leaf.

$$\gamma_{jm} = \frac{\sum_{i=1}^{J_m} r_{im}}{J_m}$$

(D) Update

$$F(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

$\sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$ Says we should add up the output values γ_{jm} for all the leaves R_{jm} that a sample x can be found in. ν is the learning rate which is a value between 1 and 0.

Basically we add the value of the old prediction $F_{m-1}(x)$ and then add the values of the leaves that we end up in following the decision tree with a value of x, multiplied by the learning rate.

4 Step 3

Output $F_m(x)$.

Say M=100, if we have some new data we want to predict we can output $F_{100}(x)$ which will be the output of the initial prediction + the output values of running x through the next 100 trees.