Gradient Boosting Algorithms

Epoch IIT Hyderabad

Arin Aggarwal
MA22BTECH11006

Contents

1 Introduction

Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error using gradient descent. It is used for both claasification as well as regression problems. Both are done in similar ways just that for classification we use $log(odds)$ everywhere instead of target feature in regression. It is later converted back to probability for classification.

2 Working

1) The first step includes building a base model to predict the dataset. For this, we take an average of the target column.

$$F_0(x) \ = \ argmin_\gamma \sum_i L(y_i, \gamma)$$

where,

$$L \ = \ Loss \ Function \ = \ \frac{1}{2}\sum_i (y_i - \gamma_i)^2$$

$$\gamma \ = \ Predeicted \ Value$$

$$argmin \ = \ gamma \ to \ be \ found \ of \ which \ the \ loss \ function \ is \ minimum$$

For argmin the γ comes out to be the average. This is our 1st predicted value.

2) Find the pseudo residuals (observed value – predicted value). The predicted value in this is the value forecasted in the prior model.
3) Now make tree to predict the **residual** value and **not the target**.
4) Next step includes finding the output values of each leaf of the decision tree . We have to find the output for all the leaves since a single leaf might get more than one result. This will once again be done by taking the average of all the values.
5) These steps will be done number of times and each time we will get nearer to optimal value. We will update the model each time as:

$$F_m(x) \ = \ F_{m-1}(x) \ + \ v_m h_m(x)$$

where,

$$\begin{aligned} v_m &= Learning\ Rate \\ h_m(x) &= Tree\ leaf\ values \end{aligned}$$

6) Each time the residuals will be predicted using new trees and model will be updated. The final value will be the addition of results from all trees.
   Here the trees are 'weak learners' which are compounded to form a strong model.

3 Fine Tuning

- Learning Rate
- Number of Trees in the Forest
- Max Depth of Trees

This is usually done using Cross-Validation.