GBM Report

Donal Loitam

July 29, 2022

Contents

1 Introduction

- The **principle behind boosting algorithms** is first we built a model on the training dataset, then a second model is built to rectify the errors present in the first model.
- This is done by building a new model **on the errors or residuals of the previous model**.
- When the target column is continuous, we use **Gradient Boosting Regressor** whereas when it is a classification problem, we use **Gradient Boosting Classifier**
- The only difference between the two is the **"Loss function"**

$$\text{Regression problems} \implies \text{MSE(Mean Squared Error)} \tag{1}$$
$$\text{Classification problems} \implies \text{(log-likelihood)} \tag{2}$$

2 Key Points of the Algorithm

Gradient Boosting Regressor with an example : Following is a sample from a random dataset where we have to predict the car price based on various features. The target column is price and other features are independent features.

Row No.	Cylinder Number	Car Height	Engine Location	Price
1	Four	48.8	Front	12000
2	Six	48.8	Back	16500
3	Five	52.4	Back	15500
4	Four	54.3	Front	14000

- Step -1 The first step in gradient boosting is to build a base model to predict the observations in the training dataset. For simplicity we take an average of the target column and assume that to be the predicted value as shown below:-

Row No.	Cylinder Number	Car Height	Engine Location	Price	Prediction 1
1	Four	48.8	Front	12000	14500
2	Six	48.8	Back	16500	14500
3	Five	52.4	Back	15500	14500
4	Four	54.3	Front	14000	14500

Although there is math involved behind this. Mathematically the first step can be written as:

$$F_0(x) = \arg\min_\gamma \sum L(y_i, \gamma) \tag{3}$$

Here L is our loss function

γ is our predicted value

argmin means we have to find a predicted value/γ for which the loss function is minimum. Since the target column is continuous our loss function will be:

$$L = \frac{1}{n}\sum_{i=0}^{n}(y_i - \gamma_i) \tag{4}$$

$$\frac{dL}{d\gamma} = -\sum_{i=0}^{n}(y_i - \gamma_i) = 0 \tag{5}$$

If you calculate eq(5) and find the optimal value for γ, it turns out that it is the average of the observed car price

- Step-2 The next step is to calculate the pseudo residuals which are (observed value – predicted value)

Row No.	Cylinder Number	Car Height	Engine Location	Price	Prediction 1	Residual 1
1	Four	48.8	Front	12000	14500	-2500
2	Six	48.8	Back	16500	14500	2000
3	Five	52.4	Back	15500	14500	1000
4	Four	54.3	Front	14000	14500	-500

This step can be written as :-

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \qquad \text{for i=1,2 ...., n} \tag{6}$$

Here $F(x_i)$ is the previous model and m is the number of DT made. We are just taking the derivative of loss function w.r.t the predicted value and we have already calculated this derivative:

$$\frac{dL}{d\gamma} = -(y_i - \gamma_i) = -(\text{observed} - \text{predicted}) \tag{7}$$

$$= -(\text{observed} - 14500) \tag{8}$$

The predicted value here is the prediction made by the previous model. In the next step, we will build a model on these pseudo residuals and make predictions.Because we want to minimize these residuals and minimizing the residuals will eventually improve our model accuracy
So, using the Residual as target and the original feature Cylinder number, cylinder height, and Engine location we will generate new predictions.
Let's say $h_m(x)$ is our DT made on these residuals.

- Step- 4 : In this step we find the output values for each leaf of our decision tree. TO find the output we can simply take the **average of all the numbers in a leaf**, doesn't matter if there is only 1 number or more than 1. Mathematically this step can be represented as:

$$\gamma_m = \arg\min_\gamma \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \tag{9}$$

When m=1 we are talking about the 1st DT and when it is "M" we are talking about the last DT.Let's take an example

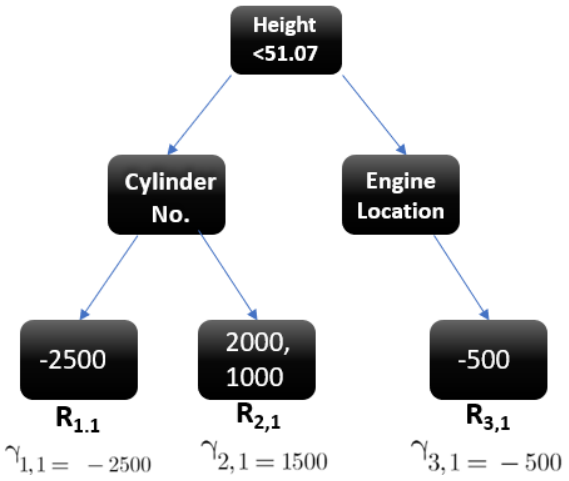


Figure 1: Calculating gamma values for each leaves

As you can observe , We end up with the average of the residuals in the leaf R2,1.

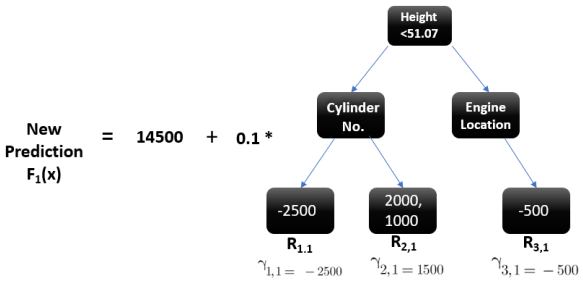
- Step-5 This is finally the last step where we have to update the predictions of the previous model. It can be updated as:

$$F_m(x) = F_{m-1}(x) + \mu_m h_m(x) \tag{10}$$

where m is the number of decision trees made.
Now to make a new DT our new predictions will be:

$$\text{New Prediction} = (\text{previous pred}) + (\text{learning rate} * \text{the tree made on residuals}) \tag{11}$$

Here, $F_{m-1}(x)$ is the prediction of the base model (previous prediction), μ is the learning rate Let $\mu = 0.1$, the new prediction now:



- **When to stop the iteration ?**
  Suppose we want to find a prediction of our first data point which has a car height of 48.8. This data point will go through this decision tree and the output it gets will be multiplied with the learning rate and then added to the previous prediction.

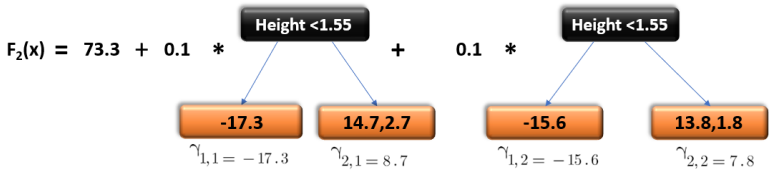
  Now let's say m=2 which means we have built 2 decision trees and now we want to have new predictions.

  This time we will add the previous prediction that is $F_1(x)$ to the new DT made on residuals. We will iterate through these steps again and again till the **loss is negligible.**

  Each time we add a new tree to the prediction, the **residuals** get smaller.

3 Some Questions

1. How do we predict the output for a new dataset?
Ans. I am taking a hypothetical example here just to make you understand how this predicts for a new dataset:



If a new data point says height = 1.40 comes, it'll go through all the trees and then will give the prediction. Here we have only 2 trees hence the datapoint will go through these 2 trees and the final output will be $F_2(x)$.

2.
Ans.

3.
Ans :
4.
Ans.
5. How can you evaluate the performance of a dimensionality reduction algorithm on your dataset?

Ans.