XGBoost

Mannem Charan AI21BTECH11019

Abstract

This report consists of my basic understanding of one of the popular Ml methods ”XG-Boost”.

1 XGBoost

XGBoost is a boosting algorithm which comes under ensemble learning where the decision trees (weak learners) are combined sequentially to predict the output of the task.It is also known as eXtreme gradient boosting and it is a modification of classical gradient boosting technique.Same as in gradient boost , XGBoost creates decision trees of residual at each step and uses it for prediction.But in doing so it uses something known as ”Similarity scores” by which we can get access to new features like pruning trees and regularization.Now we will see Xgboost in action using a dataset as an example.

2 Understanding XGBoost

Consider the dataset given below ,

Years Of Experience	Gap	Annual Salary($100k$)
1	No	4
1.5	Yes	4
2.5	Yes	5.5
3	Yes	7
5	No	7.5
6	NO	8

Here the target attribute is ”Annual Salary($100k$)” and to predict that we have two features in our disposal , ”Years of Experience” and ”Gap”.So to start the algorithm, first we need a base learner to make the first prediction.There is no correct way of doing it but in general we will take average of salary as the first prediction.

1) Then

$$F_0 = \frac{4+4+5.5+7+7.5+8}{6} \tag{2.1}$$

$$= 6\,(100K) \tag{2.2}$$

Here we are indicating the prediction of base learner as $F_0$to make it analagous to what we did in GBM.

2) After the first step we will find the pseudo residual values,which is the measure of error due to our first prediction.

$$PseudoResidual = observed_i - predicted_i \tag{2.3}$$

We are calling it as pseudo residual since if the cost function has fraction $\frac{1}{2}$ then the cost function is similar to that of linear regression where we call the error as ”residuals” but often the constant term multiplying may not be $\frac{1}{2}$ that is the reason why we are calling it as ”pseudo residuals”.
So now our new dataset will be,

Years of Experience	Gap	Annual Salary($100k$)	Pseudo Residuals
1	No	4	-2
1.5	Yes	4	-2
2.5	Yes	5.5	-0.5
3	Yes	7	1
5	No	7.5	1.5
6	No	8	2

3) In this step, we will construct a DT which predict the residuals.XGBoost follows an unique way doing it, it will first construct a leaf of residuals.And we will calculate similarity/quality score of the leaf using the below formula.

$$SimilarityScore = \frac{(\text{Sum of Residuals})^2}{No.ofResiduals + \lambda} \tag{2.4}$$

where λ is regularization parameter.
Now if we are doing classification, the similarity score will be

$$SimilarityScore = \frac{(\text{Sum of Residuals})^2}{\sum_i p_i(1-p_i) + \lambda} \tag{2.5}$$

where p_i is the predicted probability of the i^{th} residual in the leaf.
In this case the similarity score of the leaf is ,

$$Similarityscore = \frac{(-2 + -2 + -0.5 + 1 + 1.5 + 2)^2}{6 + \lambda} \tag{2.6}$$

$$= \frac{0}{6 + \lambda} \tag{2.7}$$

$$= 0 \tag{2.8}$$

The similarity score is the measure of how similar are residuals in a leaf.Now we will split the leaf in say that the chidrens have more similarity scores than the leaf we actually took. For that we will take Exp ¡2.75 as decsion maker. Now we will

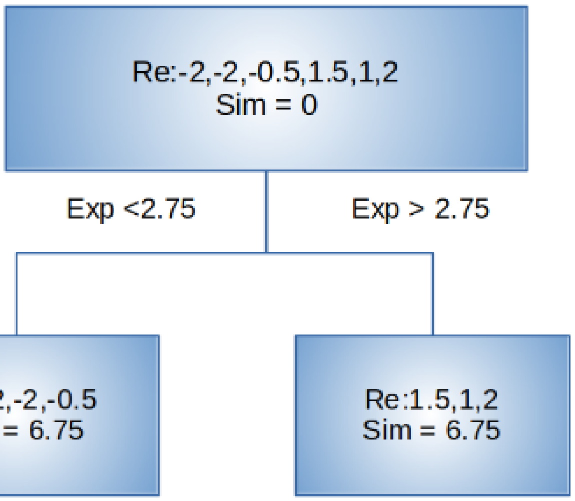

Fig. 2.1: First Split

calculate similarity scores of left and right branches with $\lambda = 0$,

$$\begin{aligned} S_L &= \frac{(-2 + -2 + -0.5)^2}{3+\lambda} && (2.9)\\ &= \frac{(-4.5)^2}{3+0} && (2.10)\\ &= 6.75 && (2.11)\\ S_R &= \frac{(1 + 1.5 + 2)^2}{3+\lambda} && (2.12)\\ &= \frac{(4.5)^2}{3+0} && (2.13)\\ &= 6.75 && (2.14) \end{aligned}$$

Then we will calculate the **Gain** for this split,

$$\begin{aligned} Gain &= S_L + S_R - S_{\text{Root}} && (2.15)\\ &= 6.75 + 6.75 - 0 && (2.16)\\ &= 13.5 && (2.17) \end{aligned}$$

And I will assure you that it is the best split as we are able to ”cluster” the similar residuals into different groups.

Then we will again split the DT with Gap as the parameter,

$$S_L = \frac{(-2-0.5)^2}{2+\lambda} \tag{2.18}$$

$$= \frac{6.25}{2+0} \tag{2.19}$$

$$= 3.125 \tag{2.20}$$

$$S_R = \frac{4}{1+0} \tag{2.21}$$

$$= 4 \tag{2.22}$$

Then Gain in this split will be,

$$Gain = S_L + S_R - S_{Root} \tag{2.23}$$

$$= 3.125 + 4 - 6.75 \tag{2.24}$$

$$= 0.375 \tag{2.25}$$

So the final DT will be as in 2.2, Now before going to predict the output of each leaf node we will first understand what is the role of λ.In XGBoost to prune the branches we will check whether the value Gain $-\gamma$ is greater or less than 0.This γ is user-defined so it is a tuneable parameter in the model.Now we can clearly say that when $\lambda > 0$ the similarity scores will be lesser which make the overall **Gain** to be smaller than the older value.So depending on λ we will prune the trees using the following heuristic,

$$Gain - \gamma = \begin{cases} > 0 & ,\text{then do not prune} \\ < 0 & ,\text{then prune} \end{cases} \tag{2.26}$$

In Classification, using the equation mention in (2.5) we will find the gain in each split and using that we will prune the branches**and also** there is something known as "**Cover**" which is nothing but the replacement for number of residuals in denominator of similarity score in classification i.e.,

$$Cover = \sum_i p_i(1-p_i) \tag{2.27}$$

Now the deal here is the cover should have a minimum value of 1, if it not the case we will simply remove that leaf node.This term analagous to the no. of residuals term in regression.
And the last thing is if you remember when we started creating the DT we said we take leaf where it later splits into branches then we are again refering it as root. The thing is, depending on choice of parameter γ and λ it may happen that we will

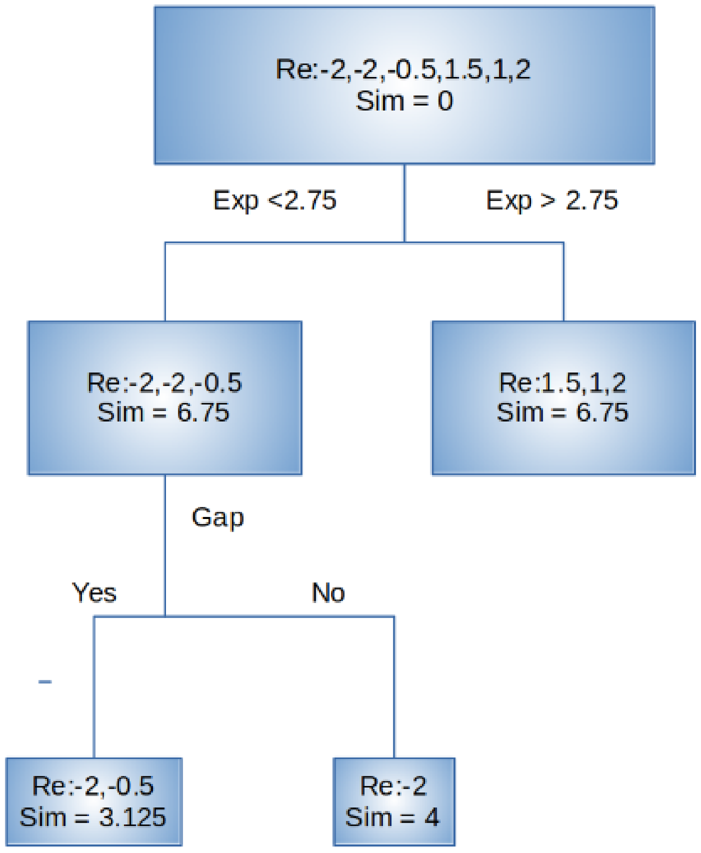

Fig. 2.2: Residual tree

prune all the branches and left with the leaf itself.So we kinda start creating tree with our output/leaf node as the root itself.

4) In this step, we will compute the output values of each leaf node.This is similar to what we did in Classical GBM but here we will have extra λ,

$$Output = \frac{\text{Sum of Residuals}}{\text{Number of Residuals} + \lambda} \tag{2.28}$$

Note that λ value is also responsible in shrinking the output value.
And for the Classification the output will be,

$$Output = \frac{\text{Sum of Residuals}}{\textbf{Cover} + \lambda} \tag{2.29}$$

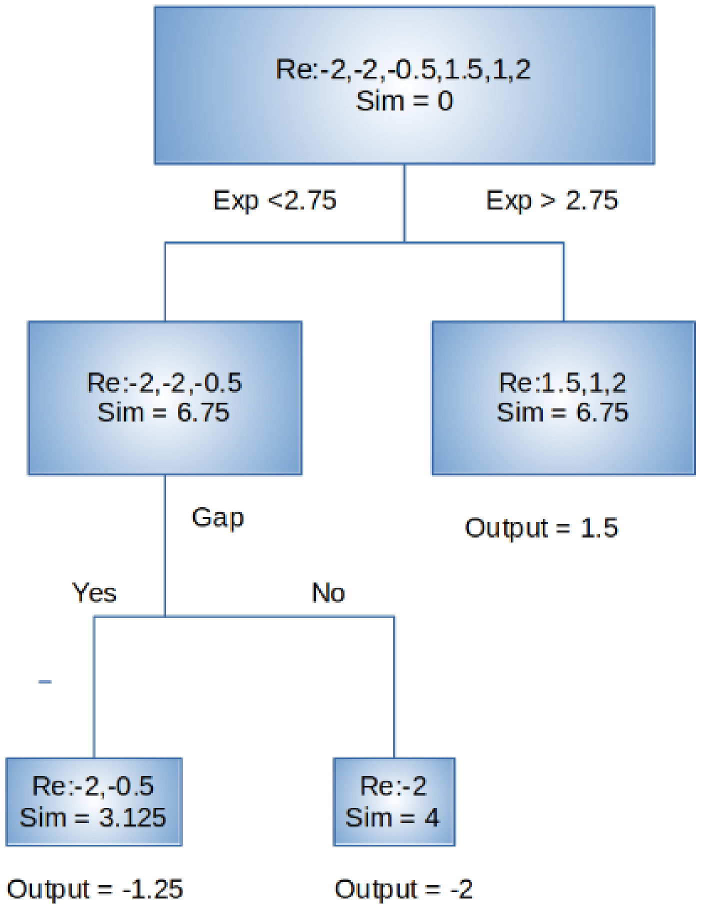

Fig. 2.3: Residual Tree with predicted output

In this case, the output value of each leaf node will be as shown in fig 2.3,

5) In final step, we the help of learning rate we will make a new prediction using old prediction and the residual tree.

$$Newprediction = Oldprediction + \eta(\text{output from residual trees}) \tag{2.30}$$

where, η is the learning rate.
In this case, we will take $\eta = 0.3$ and using that the new predictions of each input

instance x_i will be ,

$$\begin{aligned} F_1(x_1) &= F_0(x_1) + 0.3(-2) \\ &= 6 - 0.6 \\ &= 5.4 \end{aligned} \tag{2.31}$$

$$\begin{aligned} F_1(x_2) &= F_0(x_2) + 0.3(-1.25) \\ &= 6 - 0.375 \\ &= 5.625 \end{aligned} \tag{2.32}$$

$$\begin{aligned} F_1(x_3) &= F_0(x_3) + 0.3(-1.25) \\ &= 6 - 0.375 \\ &= 5.625 \end{aligned} \tag{2.33}$$

$$\begin{aligned} F_1(x_4) &= F_0(x_4) + 0.3(1.5) \\ &= 6 + 0.45 \\ &= 6.45 \end{aligned} \tag{2.34}$$

$$F_1(x_5) = 6.45 \tag{2.35}$$

$$F_1(x_6) = 6.45 \tag{2.36}$$

So the new dataset looks like,

Years of Experience	Gap	Annual Salary($100k$)	NewPrediction	New Residuals
1	No	4	5.4	-1.4
1.5	Yes	4	5.625	-1.625
2.5	Yes	5.5	5.625	-0.125
3	Yes	7	6.45	0.55
5	No	7.5	6.45	1.05
6	No	8	6.45	1.55

As you can see we moved in right direction for each input instance and from these assumptions we will construct a new Residual tree and so on.
In Classification we will do things slightly different, we will find the $\log(\text{odds})$ of the intial predicted probability and add it with thescaled output from the residual tree. Then we will again convert the resulting value into probability.This value will be the new predicted probability.The following is the expression of finding $\log(\text{odds})$ from the probability p and vice- versa,

$$\log(odds) = \log\left(\frac{p}{1-p}\right) \tag{2.37}$$

$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} \tag{2.38}$$

With this we covered most of the concepts behind XGBoost.

3 Advantages of XGboost

Some of the advantages of XGBoost are as follows,

- Effective with large data sets.
- The model is highly flexible
- It supports regularization
- It doesn't requires normalization