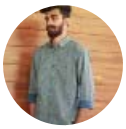


Rohith Gandhi Follow
What I cannot create, I do not understand - Richard Feynman. LinkedIn: www.linkedin.com/in/grohith327
May 6 · 4 min read

Gradient Boosting and XGBoost

- Starting from where we ended, let's continue on discussing different boosting algorithm. If you have not read the previous article which explains boosting and AdaBoost, please have a look.

  Link: https://medium.com/@grohith327/boosting-algorithms-adaboost-gradient-boosting-and-xgboost-f74991cad38c

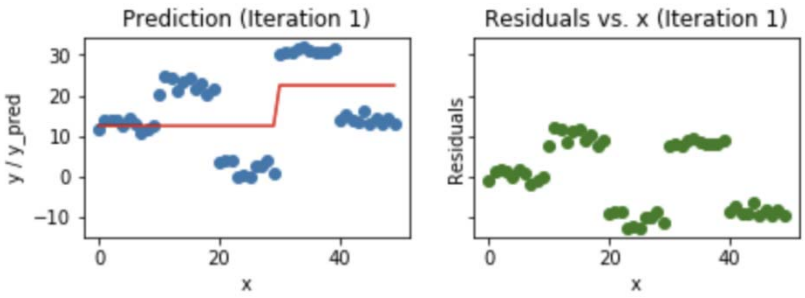
Gradient Boosting:

Moving on, let's have a look another boosting algorithm, gradient boosting. Gradient Boosting is also a boosting algorithm(Duh!), hence it also tries to create a strong learner from an ensemble of weak learners. This is algorithm is similar to Adaptive Boosting(AdaBoost) but differs from it on certain aspects. In this method we try to visualise the boosting problem as an optimisation problem, i.e we take up a loss function and try to optimise it. This idea was first developed by Leo Breiman.

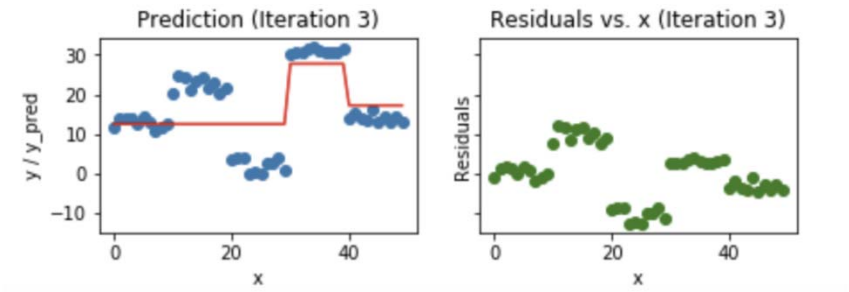
How is Gradient Boosting interpreted as an optimisation problem?

We take up a weak learner(in previous case it was decision stump) and at each step, we add another weak learner to increase the performance and build a strong learner. This reduces the loss of the loss function. We iteratively add each model and compute the loss. The loss represents the error residuals(the difference between actual value and predicted value) and using this loss value the predictions are updated to minimise the residuals.

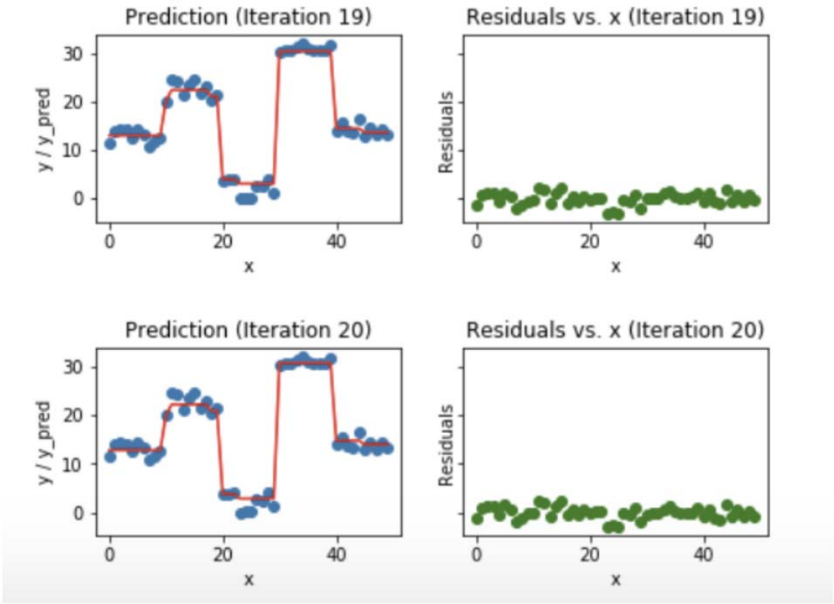
Let us break it down step by step.



In the first iteration, we take a simple model and try to fit the complete data. You can from the above image that the prediction values of the model of the ground truth are different. The error residuals are plotted on the right side of the image. The loss function is trying to reduce these error residuals by adding more weak learners. The new weak learners are added to concentrate on the areas where the existing learners are performing poorly.



After three iterations, you can observe that model is able to fit the data better. This process is iteratively carried out until the residuals are zero.



After 20 iterations, the model almost fits the data exactly and the residuals drop to zero.

```
# Gradient Boosting
from sklearn.ensemble import GradientBoostingClassifier


clf = GradientBoostingClassifier()
# n_estimators = 100 (default)
# loss function = deviance(default) used in Logistic
Regression
clf.fit(x_train,y_train)
```

```
clf.predict(x_test)
```

XGBoost(Extreme Gradient Boosting):

XGBoost has taken data science competition by storm. XGBoost seems to be a part of an ensemble of classifiers/predictors which are used to win data science competitions. Why is this so? why is XGBoost so powerful ?

XGBoost is similar to gradient boosting algorithm but it has a few tricks up its sleeve which makes it stand out from the rest.

Features of XGBoost are:

Clever Penalisation of Trees

A Proportional shrinking of leaf nodes

Newton Boosting

Extra Randomisation Parameter

In XGBoost the trees can have a varying number of terminal nodes and left weights of the trees that are calculated with less evidence is shrunk more heavily. Newton Boosting uses Newton-Raphson method of approximations which provides a direct route to the minima than gradient descent. The extra randomisation parameter can be used to reduce the correlation between the trees, as seen in the previous article, the lesser the correlation among classifiers, the better our ensemble of classifiers will turn out. Generally, XGBoost is faster than gradient boosting but gradient boosting has a wide range of application

```
# XGBoost
from xgboost import XGBClassifier
clf = XGBClassifier()
# n_estimators = 100 (default)
# max_depth = 3 (default)
clf.fit(x_train,y_train)
clf.predict(x_test)
```

Conclusion

These tree boosting algorithms have gained huge popularity and are present in the repertoire of almost all kagglers. I hope these two-part articles would've given you some basic understanding of the three algorithms

Reference:

Gradient Boosting from scratch

Simplifying a complex algorithm

medium.com

https://brage.bibsys.no/xmlui/bitstream/handle/11250/2433761/16128_FULLTEXT.pdf