Gradient Boosting Algorithms
Epoch IIT Hyderabad

Chakka Surya Saketh
AI22BTECH11005

Introduction

Ensemble simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model. It uses two types of methods namely-

1) Bagging: It creates a different training subset from sample training data with replacement and the final output is based on majority voting.
2) Boosting: It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy.

The family of gradient boosting algorithms has been recently extended with several interesting proposals (i.e. XGBoost, LightGBM and CatBoost) that focus on both speed and accuracy. There are mainly two types of error, bias error and variance error. Gradient boost algorithm helps us minimize bias error of the model.

The main idea behind this algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model. The only difference between the two(for regression and for classification) is the ”Loss function”. We could have MSE(mean squared error) for regression and log-likelihood for classification.

Algorithm workflow

1) Initialize the ensemble with a simple model (e.g., a single leaf or constant value).
2) Calculate the negative gradient of the loss function with respect to the current ensemble's predictions.
3) Fit a weak learner (e.g., decision tree) to the negative gradient values.
4) Update the ensemble by adding the predictions of the new learner, scaled by a learning rate.
5) Repeat steps 2-4 for a predefined number of iterations or until a convergence criterion is met.

Variations of Gradient Boosting

- XGBoost(Extreme Gradient Boosting): Incorporates regularization, handling missing values, and parallel processing to enhance performance.
- LightGBM (Light Gradient Boosting Machine): Optimizes memory usage and training speed by utilizing histogram-based techniques.
- CatBoost (Categorical Boosting): Designed to handle categorical features effectively by using an innovative method to encode them.
- AdaBoost (Adaptive Boosting): One of the earliest boosting algorithms, it assigns weights to instances and focuses on misclassified samples.

Hyper parameter tuning

1) Number of trees: Balances underfitting (few trees) and overfitting (many trees).
2) Max Depth: Limits tree depth to manage complexity.
3) Min Samples per Leaf: Prevents overfitting by requiring minimum samples in a leaf node.
4) Min Samples per Split: Controls tree growth with minimum samples for internal node splitting.
5) Subsample: Fraction of data for tree fitting. ¡1.0 adds randomness to prevent overfitting.
6) Loss Function: Problem-specific loss.
7) Early Stopping: Prevents overfitting by monitoring validation error.
8) Regularization: Key to control complexity.

Conclusion

The Gradient Boosting algorithm stands as a powerful and versatile tool in the machine learning arsenal. Its ability to iteratively improve model predictions by learning from errors and emphasizing their corrections has made it a staple in both research and real-world applications. By understanding the fundamentals, variations, advantages, and limitations discussed in this report, practitioners can make informed decisions when applying Gradient Boosting to their own data analysis tasks.