Decision Trees and Ensemble Learning Methods

Over the past few decades, ensemble learning methods have grown significantly and have been centered around aggregating decision trees. Classification and Regression Trees (CART) are a class of decision tree algorithms that are used for classification or regression problems. The CART model is represented by a binary decision tree in which each node represents an input variable; a binary decision is made at each node based on some if-else condition that the input variable either passes or fails. Each decision stems to another branch of the tree that will lead to a leaf node, with this process continuing until a final prediction is made. To identify the criteria for splits at each node, the CART model uses the Gini Impurity Test, which evaluates the optimal splits to have the highest classification accuracy for the tree.

Ensemble learning is a type of machine learning where multiple models, which are called weak learners, are combined to create a stronger model that can solve the same problem with more accuracy. Weak learners often have high variance or bias, and the goal of ensemble learning is to reduce bias or variance by combining weak learners into a strong learner. Bagging is one way that we can combine weak learners to build a strong learner.

Bagging stands for "bootstrap aggregating", and it is a parallel ensemble method that trains each weak learner independently or concurrently. Bootstrapping is a statistical method that randomly draws samples of size n from an initial dataset without replacement; these samples are bootstrap samples that are meant to be representations of the true data distribution. A weak learner is trained for each bootstrap sample, and the weak learners can be aggregated by simply averaging the outputs. For regression problems, outputs are averaged for each model to produce outputs for the strong learner; for classification problems, a majority vote is used by choosing at the class that receives the majority vote as the output for the strong learner.

CART models are often used as the base model for weak learners in a bagging problem. In a random forest model, multiple trees are combined to produce a forest of CART models to generate outputs with lower variance. CART models are trained on bootstrap samples, however, in a random forest model, bootstrap samples are slightly different: samples are bootstrapped over

both features and observations. This helps reduce correlation between trees because by sampling over features of the dataset, not all CART models look at the same information. The outputs from each tree are then averaged to produce a single output from the random forest model, our strong learner. Bagging is an ensemble learning method that looks to reduce variance, and this means that weak learners often have high variance individually and are combined to have a lower variance. With the random forest model, CART models that are deeper with more branches would be used as the base model because deeper trees typically have higher variances.

Boosting works in a similar fashion to bagging in that it looks to combine multiple weak learners into a strong learner, however, it differs in that boosting is not a parallel learning method, and so weak learners are not fitted independently. Instead, weak learners are trained sequentially, meaning that as weak learners are trained, new models are fitted by focusing on the observations that were poorly handled by the previous model. Thus, boosting is an adaptive process: as the next model is fitted, it adapts to the previous model and assigns more importance to the observations that the previous model struggled with. Boosting focuses on reducing bias, and so weak learners are often chosen to have high bias and low variance. Regarding CART models, shallow trees are used as base models because they have high bias with lower variance.

One way in which boosting is implemented is through adaptative boosting, or AdaBoost. AdaBoost trains the first tree by weighting each observation in the bootstrap sample equally; the second tree is then trained by assigning higher weights to the observations that were more difficult to classify. This process is then repeated for a given number of iterations. Gradient boosting trains decision trees through boosting, but differs from AdaBoost in that Gradient boosting is based on a loss function, which computes the residuals of the model's outputs. As each new tree is constructed, the parameters of the tree that will minimize the loss function of the ensemble are selected through a gradient descent algorithm. Thus, gradient boosting continuously attempts to reduce the error of the ensemble model by minimizing the loss function. This process is called functional gradient descent.

Extreme Gradient Boosting, or XGBoost, is an algorithm that is used to implement gradient boosting. The main advantage of XGBoost is that it utilizes regularized boosting, which

prevents overfitting of any tree in the boosting process and allows for the model to be more generalized. XGBoost also uses cross validation at each iteration, which allows for the model to evaluate its performance at each iteration to find the optimal stopping point, and parallel processing, which allows for XGBoost to be used with large datasets.

References

Brownlee, Jason. “A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning.” *Machine Learning Mastery*, 14 Aug. 2020, https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

Dobilas, Saul. “CART: Classification and Regression Trees for Clean but Powerful Models.” *Medium*, Towards Data Science, 10 Mar. 2022, https://towardsdatascience.com/cart-classification-and-regression-trees-for-clean-but-powerful-models-cc89e60b7a85.

Rocca, Joseph. “Ensemble Methods: Bagging, Boosting and Stacking.” *Medium*, Towards Data Science, 21 Mar. 2021, https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

Singh, Harshdeep. “Understanding Gradient Boosting Machines.” *Medium*, Towards Data Science, 4 Nov. 2018, https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

Victor Zhou. “A Simple Explanation of Gini Impurity.” *Victor Zhou*, Victor Zhou, 29 Mar. 2019, https://victorzhou.com/blog/gini-impurity/.

“XGBoost.” *GeeksforGeeks*, 24 Oct. 2021, https://www.geeksforgeeks.org/xgboost/.

Yiu, Tony. “Understanding Random Forest.” *Medium*, Towards Data Science, 29 Sept. 2021, https://towardsdatascience.com/understanding-random-forest-58381e0602d2.