Ensemble Methods

Beyond decision trees and ensemble classifiers

Ensembling in machine learning involves the combination of several classifiers to obtain an optimal model with better performance as opposed to just a single classifier. These classifiers can be of different algorithms and hyperparameters. Bagging, boosting, stacking and blending are methods classifiers can be combined.

Bagging

Bootstrap Aggregation or Bagging is a parallel ensembling technique that randomly bootstraps or samples the dataset with replacement to create subsets from the original. Multiple models are then trained using these subsets and the predicted results from these models aggregated to return final predictions. Bagging results in a final model that has less variance than its base classifiers.

- Bagging: Random Forests

When bagging is applied to decision trees, it results in random forests which is a supervised learning algorithm that has a large number of decision trees. For an instance in the dataset, each tree returns a prediction for the class the instance belongs to then, the class with the most votes becomes the final class for that instance. In random forests, it is assumed that a group of uncorrelated trees will do better than an individual tree. While some of the trees might be wrong in their predictions, many others will be correct.

Boosting: AdaBoost, Gradient Boosting and XGBoost

- Boosting

Boosting is a sequential process where every phase attempts to correct the errors made by the previous model. The main principle is to fit multiple weak learners which are slightly better than just random guessing. In contrast to bagging, boosting attempts to reduce both variance and bias. AdaBoost, Gradient Boosting and XGBoost are examples of boosting algorithms.

- AdaBoost

Adaptive Boosting is the first boosting algorithm. It is a very popular method for boosting that can be used on any classifier to present a more accurate model and improve its performance. It can be described with the following steps: create a subset from the entire dataset, assign equal weights to the data points, create a base model using this subset, predict using this model, calculate errors from the predicted results, assign higher weights to misclassified instances to increase their chances of being selected, create another model that tries to correct these mistakes and make new predictions then repeat until the maximum number of models specified are created. The final model is the weighted average of all the weak learners created. AdaBoost is very sensitive to noisy data and outliers so it is important to remove these when using AdaBoost.

- Gradient Boosting

This is another boosting algorithm that improves model performance where each model in the ensemble minimizes a loss function using gradient descent. The loss function which is used to obtain an estimate of how the model is performing, a weak learner - a model only slightly better than random guessing typically decision stumps (a decision tree with a single split - one level) and an additive model that combines the weak learners to make the final model are three important components in gradient boosting.

- XGBoost

Extreme Gradient Boosting is a supervised learning algorithm that implements gradient boosting by building trees parallely while applying regularization. It is well known for its scalability and fast execution. XGBoost can automatically identify missing values in data and it builds very deep trees before pruning for optimisation.