**Ensemble Technique on Gradient Boosting and Ada Boosting**

**Ensemble Technique:**

The concept of ensemble methodology is to build a predictive model by integrating multiple models. It is familiar that ensemble methods can be used for improving prediction performance. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way, typically by weighted or unweighted voting, to classify new examples with the goal of improving accuracy and reliability.

* The Ensemble model provides an efficient way to improve accuracy and robustness over single model methods.
* Ensemble technique helps to improve machine learning results by combining several models.
* This approach allows the construction of better predictive performance compared to a single model. That is why ensemble methods placed first in many important machine learning competitions, such as the Netflix Competition, KDD 2009, and Kaggle.
* Ensemble methods provide a huge practical convenience in that it has the promise of reducing and perhaps even eliminating some key shortcomings of standard learning algorithms.
* Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to **decrease** **variance** (bagging), **bias** (boosting),and **improve predictions** (stacking)
* Learning ensembles have emerged as being among the most capable learning methods. Their structure model takes the form



Where M is the size of the ensemble and ensemble members (“base members”) fm(x) is a different function of the input variables x derived from the training data.

There are two different techniques in ensemble:

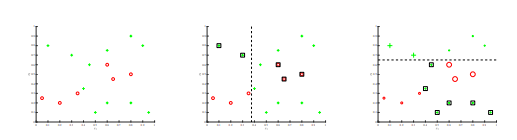
* Bagging (Bootstrap Aggregation)
* Forecasting
* Boosting
* Ada boosting
* Gradient boosting
* Xg boosting

**Bagging:**

Bagging stands for **Boostrap Aggregation** and it is a variance reduction ensembling method. Bootstrap is a method from statistics commonly used to measure uncertainty of some estimator (e.g. mean).

**Boosting:**

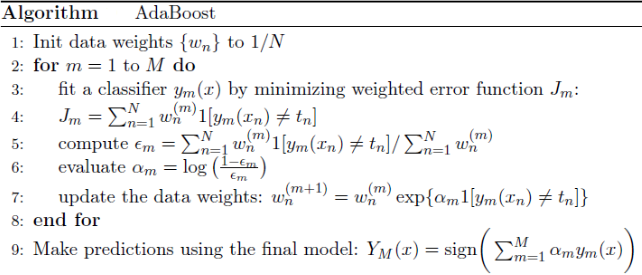
Bagging is a variance reducing technique, boosting is used for bias-reduction. We therefore want high bias, low variance models, also called as weak learners. Continuing our analysis via the use of decision trees, we can make them into weak learners by allowing each tree to only make one decision before making a prediction. These are known as decision stumps.



We analyze the intuition behind boosting via the example above. We start with a dataset on the left, and allow a single decision stump to be trained, as show in the middle panel. The key idea is that we then track which examples the classifier got wrong, and increase their relative weight compared to the correctly classified examples. We then train a new decision stump which will be more motivate to correctly classify these “hard negatives”. We continue as such, incrementally re-weighting examples at each step, and at the end we output a combination of these weak learners as an ensemble classifier.

**Adaboosting:**

The below algorithm describes the most generally used form of boosting algorithm called **AdaBoost**,which stands for adaptive boosting.



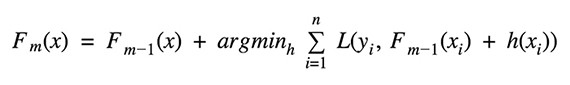
We see that the first base classifier y1(x) is trained using weighting coefficients that are all equal. In subsequent boosting rounds, the weighting coefficients are expanded for data points that are misclassified and decreased for data points that are correctly classified. The quantity epsilon represents a weighted error rate of each of the base classifiers. So that, the weighting coefficients alpha give greater weight to the more accurate classifiers.

**Gradient Boosting** :

It is a generalization of boosting to arbitrary differentiable loss function. It can be used for regression and classification problems. Gradient Boosting builds the model in a subsequent way.

https://miro.medium.com/max/325/1*NCol0wpk85JG1K5Qek-6Ig.jpeg

At each stage the decision tree hm(x) is chosen to reduce a loss function L given the current model Fm-1(x):



**Gradient boosting vs Ada boosting:**

Gradient boosting develop learners during the learning process. It build first learner to predict **labels of samples**, and calculate the loss (the difference between the outcome of the first learner and the real value). It will build a second learner to predict the loss after the first step. The step continues to learn the third, forth and fifth etc until certain threshold. Adaboost requires users specify a set of weak learners (alternatively, it will randomly generate a set of weak learner before the real learning process). It will determine the weights of how to add these learners to be a strong learner. The weight of each learner is learned by whether it predicts a sample accurately or not.

**Python code:**

Packages:

import pandas as pd

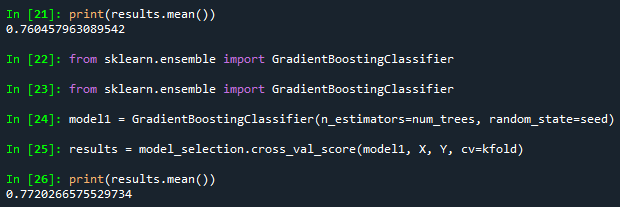
from sklearn import model\_selection is used for Split arrays or matrices into random train and test subsets

from sklearn.ensemble import AdaBoostClassifier is used to implements  the algorithm known as AdaBoost



Output:

Compare to adaboost algorithm gradient boosting is better in the above data because it has high accuracy



**Rcode:**

**Packages:**

install.packages("xgboost")

install.packages("mlbench")

install.packages("gbm")

install.packages("adabag")

