# Lab 8 – Ensemble Learning

MACHINE LEARNING

SAUMAY AGRAWAL

16BCE1151

## EXPERIMENT

- Find and report why does Bagging or Boosting work, and which is better.
- Implement Random Forest , AdaBoost and XGBoost.
- Compare the performance with your previous implementations of SVM and MLP.
- Why does XGBoost help to win competitions?

## ALGORITHMS/CONCEPTS

### ENSEMBLE LEARNING

Ensemble learning helps us in improving the results obtained from various machine learning algorithms. It does so by combining multiple learners/models, having a single objective of classification. The reason that it works is because different machine learning algorithms work differently on the same data, as they focus on different aspects of the dataset, giving rise to a stronger learner. Thus, the combination reduces the errors and gives us an overall better predictive performance.

### WHY IT WORKS?

The error in learning is mainly due to three factors:

- Noise: It is the distortion in data, that does not confirm to the general patterns and characteristics that the data exhibits. In reality, noise occurs mainly due to some fault in the methods of capturing data. Now, since it doesn't fit the general pattern of our data, it can lead to unwanted alterations to our learning function. If the model is able identify and treat noisy data separately, we say the learning is balanced. But most of the time, the noise also gets included, which results in overfitting of the model.

- Bias: It is the inability of model to correlate the input features and the output state. Bias is equal to the magnitude of difference between the predicted value and the ground truth value. Usually a high bias value means, the model is biased towards a particular way of prediction, and isn't well suited for new kind of data.

- Variance: It is the sensitivity of a model towards noise. It is equal to the magnitude of difference between the squared predicted and ground truth values.

Each machine learning model is suited for different purposes as it deals with these parameters differently. The performance of a model depends on how well it handles the noise, and thus exhibits low variance and bias values. Upon combination of multiple models, we can get significant increase in performance, as opposed to a single model.

## BAGGING

Bagging stands for bootstrap aggregation, and is a parallel ensemble technique. Similar base learners are applied on various subsets of the dataset. These subsets are randomly generated. The final result is then obtained by averaging or aggregating the outputs of all the learners. For aggregating the output, bagging uses voting for classification and averaging for regression. Thus, reduction in variance is achieved.

### Random Forest

In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e. a bootstrap sample) from the training set. In addition, instead of using all the features, a random subset of features is selected, further randomizing the tree. As a result, the bias of the forest increases slightly, but due to the averaging of less correlated trees, its variance decreases, resulting in an overall better model.

## BOOSTING

Boosting is a sequence ensemble technique. It refers to a family of algorithms that are able to convert weak learners to strong learners. The main principle of boosting is to fit a sequence of weak learners, models that are only slightly better than random guessing, such as small decision trees, to weighted versions of the data. More weight is given to examples that were misclassified by earlier rounds. The predictions are then combined through a weighted majority vote (classification) or a weighted sum (regression) to produce the final prediction.

### Adaboost

Adaptive boosting or AdaBoost is one of the simplest boosting algorithms. Usually, decision trees are used for modelling. Multiple sequential models are created, each correcting the errors from the last model. AdaBoost assigns weights to the observations which are incorrectly predicted and the subsequent model works to predict these values correctly.

**Gradient Boosting**
Gradient Tree Boosting is a generalization of boosting to arbitrary differentiable loss functions. It can be used for both regression and classification problems. At each stage the decision tree is chosen to minimize a loss function.

**XGBoost**
XGBoost, short for “Extreme Gradient Boosting”, was introduced by Chen in 2014. Since its introduction, XGBoost has become one of the most popular machine learning algorithm.

## WHICH IS BETTER BETWEEN BAGGING AND BOOSTING?

It depends on the data, the simulation and the circumstances. Bagging and Boosting decrease the variance of a single estimate as they combine several estimates from different models. So the result may be a model with higher stability.

If the problem is that the single model gets a very low performance, Bagging will rarely get a better bias. However, Boosting could generate a combined model with lower errors as it optimises the advantages and reduces pitfalls of the single model.

By contrast, if the difficulty of the single model is over-fitting, then Bagging is the best option. Boosting for its part doesn’t help to avoid over-fitting; in fact, this technique is faced with this problem itself. For this reason, Bagging is effective more often than Boosting.

## WHY DOES XGBOOST HELP TO WIN THE COMPETITIONS?

XGBoost is similar to gradient boosting algorithm but it has some additional features which make it faster than other algorithms, while giving same accuracy (in most of the cases).

Some key 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. XGBoost provides two levels of column sampling, column sampling by tree and column sampling by level, thus introducing more randomness into the learning process.

# OBSERVATIONS

## ACCURACY COMPARISON TABLE

| **Algorithm** | **Iris data** | **Olympic (20 years)** | **Olympic (40 years)** | **Olympic (120 years)** |
|---|---|---|---|---|
| Random Forest | 0.947 | 0.855 | 0.862 | 0.855 |
| AdaBoost | 0.913 | 0.856 | 0.859 | 0.853 |
| Gradient Boost | 0.94 | 0.856 | 0.86 | 0.853 |
| XGBoost | 0.956 | 0.855 | 0.862 | 0.854 |
| MLP | 0.956 | 0.858 | 0.857 | 0.854 |
| SVM | 1.00 | 0.856 | 0.862 | 0.852 |

## COMPUTATION TIME COMPARISON TABLE

| **Algorithm** | **Iris data** | **Olympic (20 years)** | **Olympic (40 years)** | **Olympic (120 years)** |
|---|---|---|---|---|
| Random Forest | 1.17s | 1m 19s | 3m 3s | 9m 32s |
| AdaBoost | 1.32s | 23s | 49.5s | 2m 8s |
| Gradient Boost | 1.41s | 1m 32s | 4m 12s | 11m 57s |
| XGBoost | 20.1ms | 5s | 11.8s | 36.9s |
| MLP | 20ms | 1.58s | 4.82s | 7.59s |
| SVM | 26.1ms | 9.78s | 40.7s | 5m 13s |

# INFERENCE

- The ensemble techniques add more robustness to our models, and helps us in creating better models.
- XGBoost is the fastest among all the ensemble techniques, while giving the same performance as compared to other alogithms.
- These algorithms, each have multiple parameters, which can be tuned to suit the model for a particular kind of dataset. This will further increase the performance of the model.
- One downside of XGBoost is that the categorical data must be converted to numerical data for its use. It doesn’t handle it by itselt. Catboost outperforms XGBoost in this area, as it can handle a variety of datatypes.