Notes on Various Topics

서용덕 *Yongduek Seo yndk@sogang.ac.kr*

2018-09-30

Contents

Chapter 1

Prerequisites

Read linear algebra, probability and statistics, computer programming to do data science.

Chapter 2

Gradient Boosting

2.1 Useful links to read

1. Gradient Boosting from scratch
2. How to explain gradient boosting by Terence Parr and Jeremy Howard
3. What is the difference between Bagging and Boosting
4. Gradient boosting machines (GBM), A Tutorial
5. Use XGBoost.

2.2 Gradient Boosting from Scratch

The reason we use ensembles is that many different predictors trying to predict same target variable will perform a better job than any single predictor alone.

Ensembling techniques are further classified into Bagging and Boosting.

1. **Bagging** is a simple ensembling technique in which we build many independent predictors/models/learners and combine them using some model averaging techniques. (e.g. weighted average, majority vote or normal average). We typically take random sub-sample/bootstrap of data for each model, so that all the models are little different from each other. Each observation is chosen with replacement to be used as input for each of the model. So, each model will have different observations based on the bootstrap process. Because this technique takes many uncorrelated learners to make a final model, it reduces error by reducing variance. Example of bagging ensemble is Random Forest models.

2. **Boosting** is an ensemble technique in which the predictors are not made independently, but sequentially. This technique employs the logic in which the subsequent predictors learn from the mistakes of the previous predictors. Therefore, the observations have an unequal probability of appearing in subsequent models and ones with the highest error appear most. (So the observations are not chosen based on the bootstrap process, but based on the error). The predictors can be chosen from a range of models like decision trees, regressors, classifiers etc. Because new predictors are learning from mistakes committed by previous predictors, it takes less time/iterations to reach close to actual predictions. But we have to choose the stopping criteria carefully or it could lead to overfitting on training data. Gradient Boosting is an example of boosting algorithm.

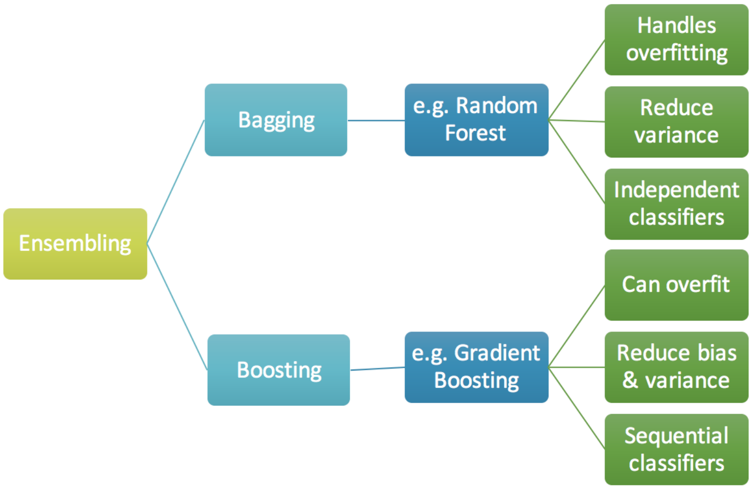

Figure 2.1: Two branches of ensemble-based learning

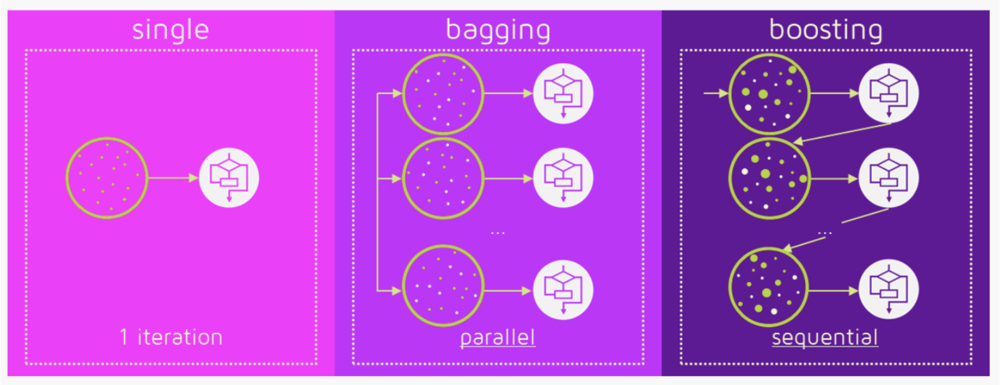

Figure 2.2: Bagging (independent models), Boosting (sequential models)

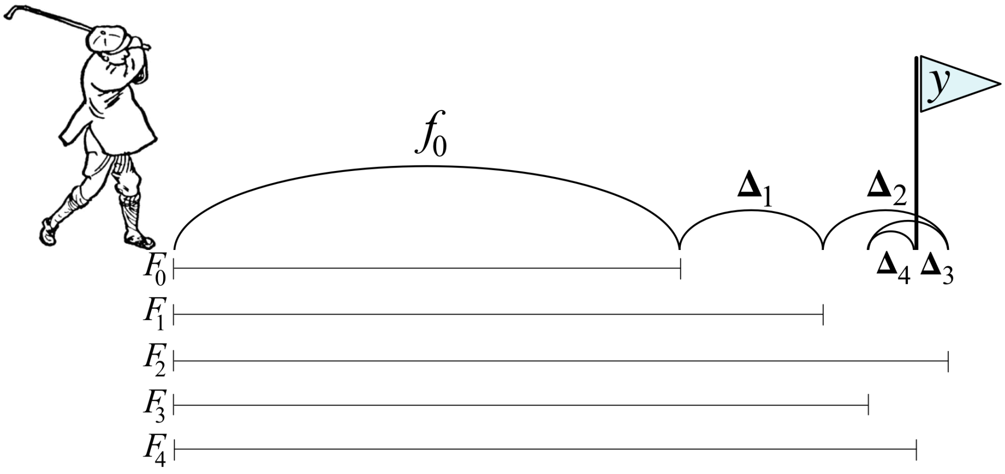

Figure 2.3: Bagging (independent models), Boosting (sequential models)

2.3 How to explain gradient boosting by Terence Parr and Jeremy Howard

Gradient boosting machines (GBMs) are currently very popular and so it's a good idea for machine learning practitioners to understand how GBMs work. The problem is that understanding all of the mathematical machinery is tricky and, unfortunately, these details are needed to tune the hyper-parameters. (Tuning the hyper-parameters is required to get a decent GBM model unlike, say, Random Forests.) Our goal in this article is to explain the intuition behind gradient boosting, provide visualizations for model construction, explain the mathematics as simply as possible, and answer thorny questions such as why GBM is performing "gradient descent in function space."

2.3.1 The intuition behind gradient boosting

To construct a boosted regression model, let's start by creating a crappy model, , that predicts an initial approximation of y given feature vector . Then, let's gradually nudge the overall model towards the known target value y by adding one or more tweaks, :

$$\begin{aligned}
\hat{y} &= f_0(\vec{x}) + \Delta_1(\vec{x}) + \Delta_2(\vec{x}) + ... + \Delta_M(\vec{x}) \\
&= f_0(\vec{x}) + \sum_{m=1}^{M} \Delta_m(\vec{x}) \\
&= F_M(\vec{x})
\end{aligned}$$

Or, using a recurrence relation, let:

$$\begin{aligned}
F_0(\vec{x}) &= f_0(\vec{x}) \\
F_m(\vec{x}) &= F_{m-1}(\vec{x}) + \Delta_m(\vec{x})
\end{aligned}$$