Ensemble learning methods

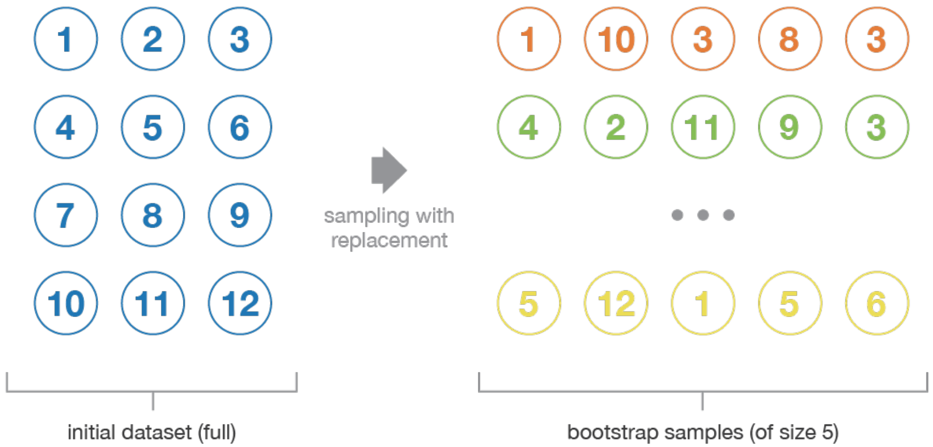
Bagging, Random Forests, Boosting

Bagging

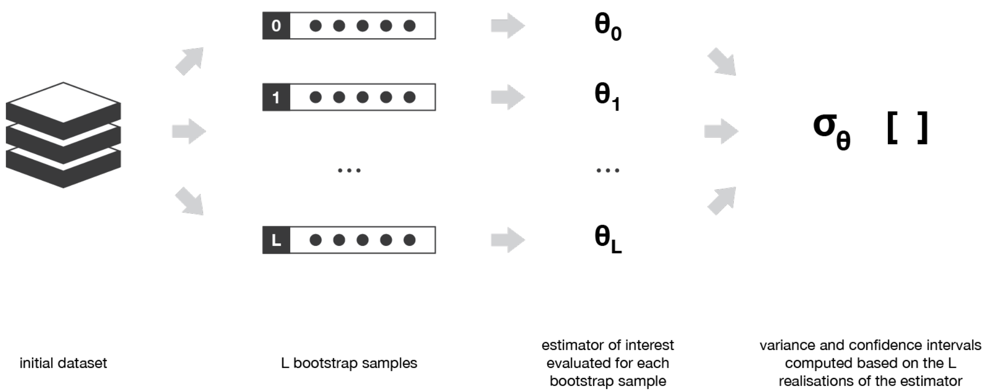
(Bootstrap aggregating)

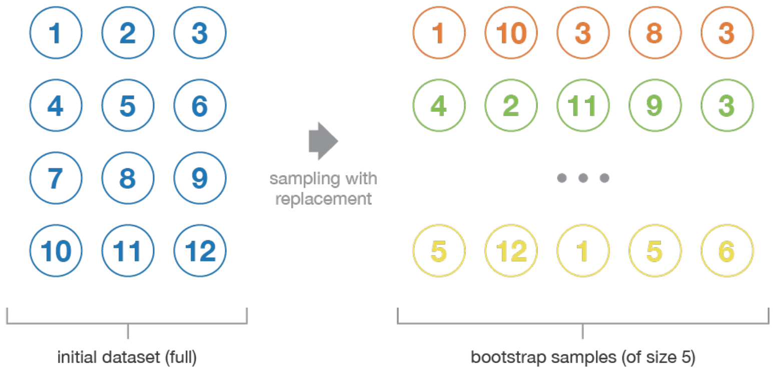
Bagging is used when the goal is to reduce the variance of a decision tree classifier. Here the objective is to create several subsets of data from training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees. As a result, we get an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree classifier.

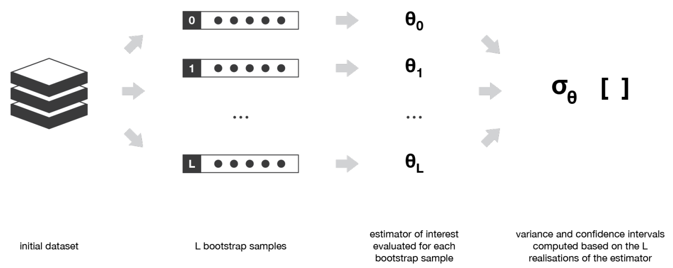
Bootstrapping



This technique consists in generating samples of size B (called bootstrap samples) from an initial dataset of size N by randomly drawing with replacement B observations.





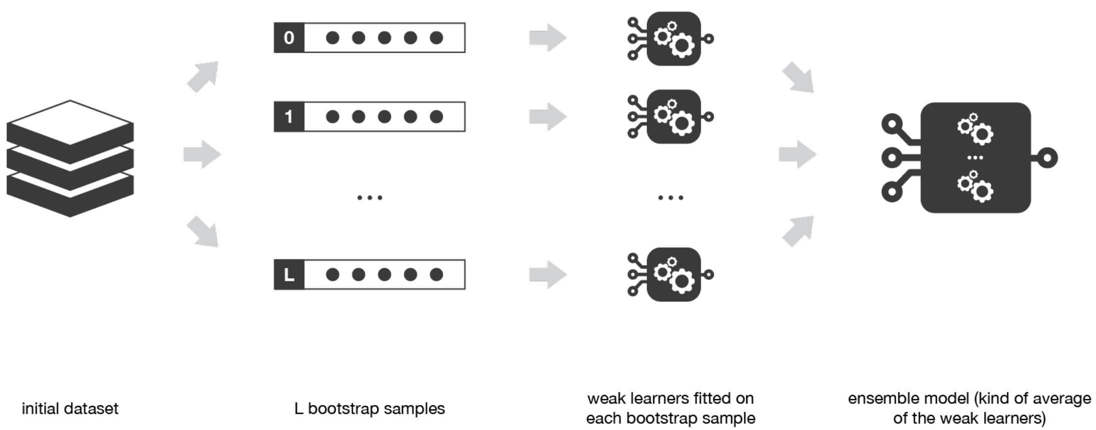


Bagging

(Bootstrap aggregating)

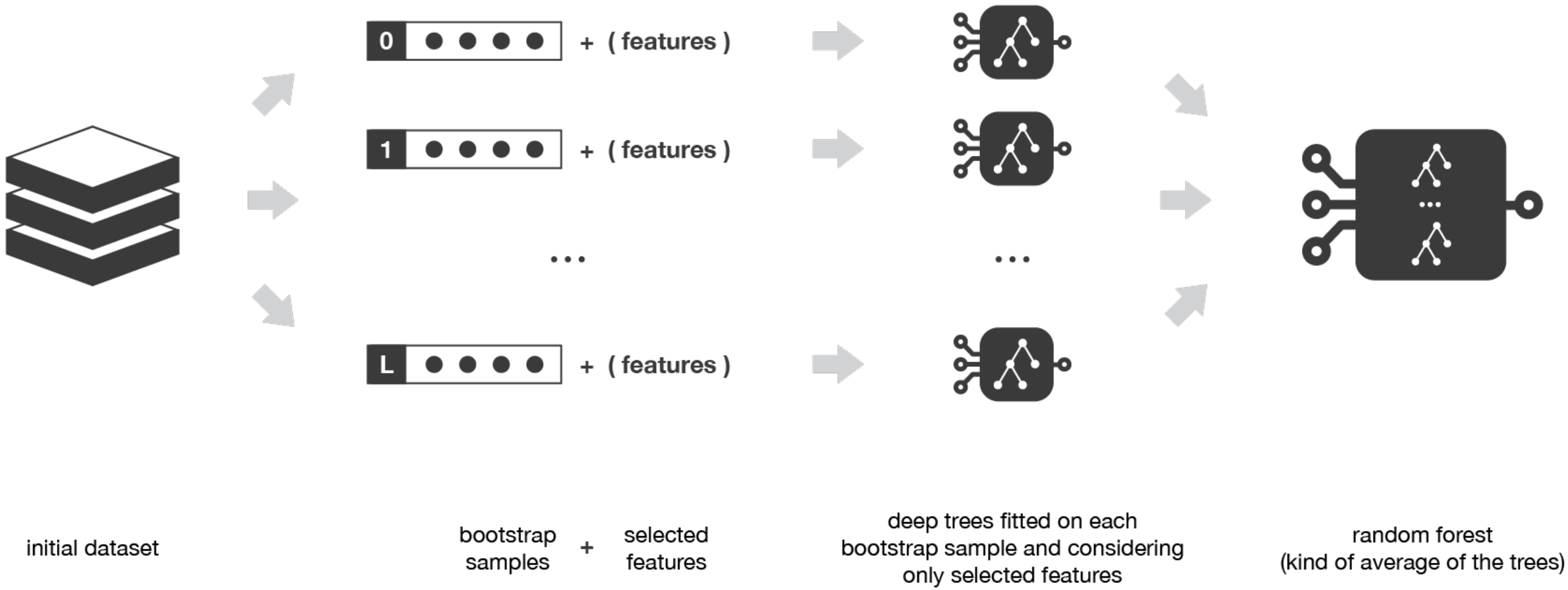
Bagging is used when the goal is to reduce the variance of a decision tree classifier. Here the objective is to create several subsets of data from training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees. As a result, we get an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree classifier.

We want to fit several independent models and "average" their predictions in order to obtain a model with a lower variance



Examples and coding…

Random forests

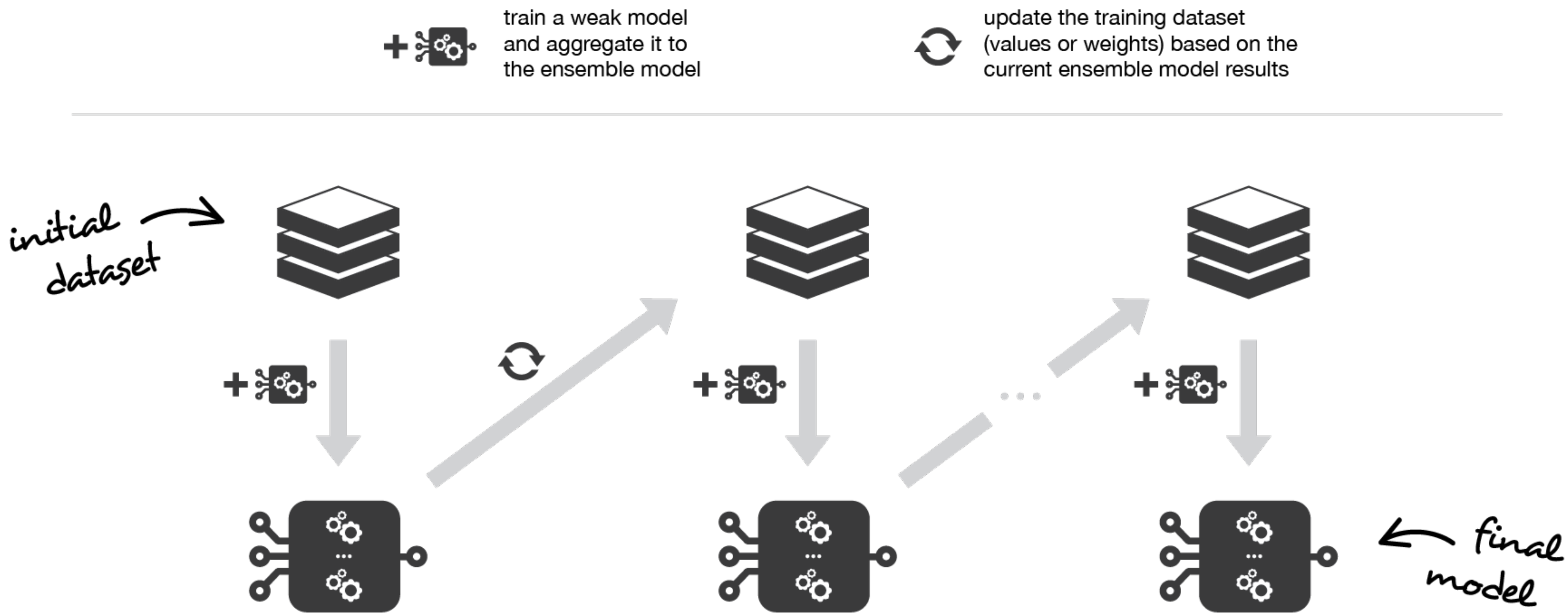


The **random forest** approach is a bagging method where **deep trees**, fitted on bootstrap samples, are combined to produce an output with lower variance.

Main difference:
When growing each tree, instead of only sampling over the observations in the dataset to generate a bootstrap sample, we also **sample over features** and keep only a random subset of them to build the tree.

This makes the decision making process more robust to missing data

Examples and coding…

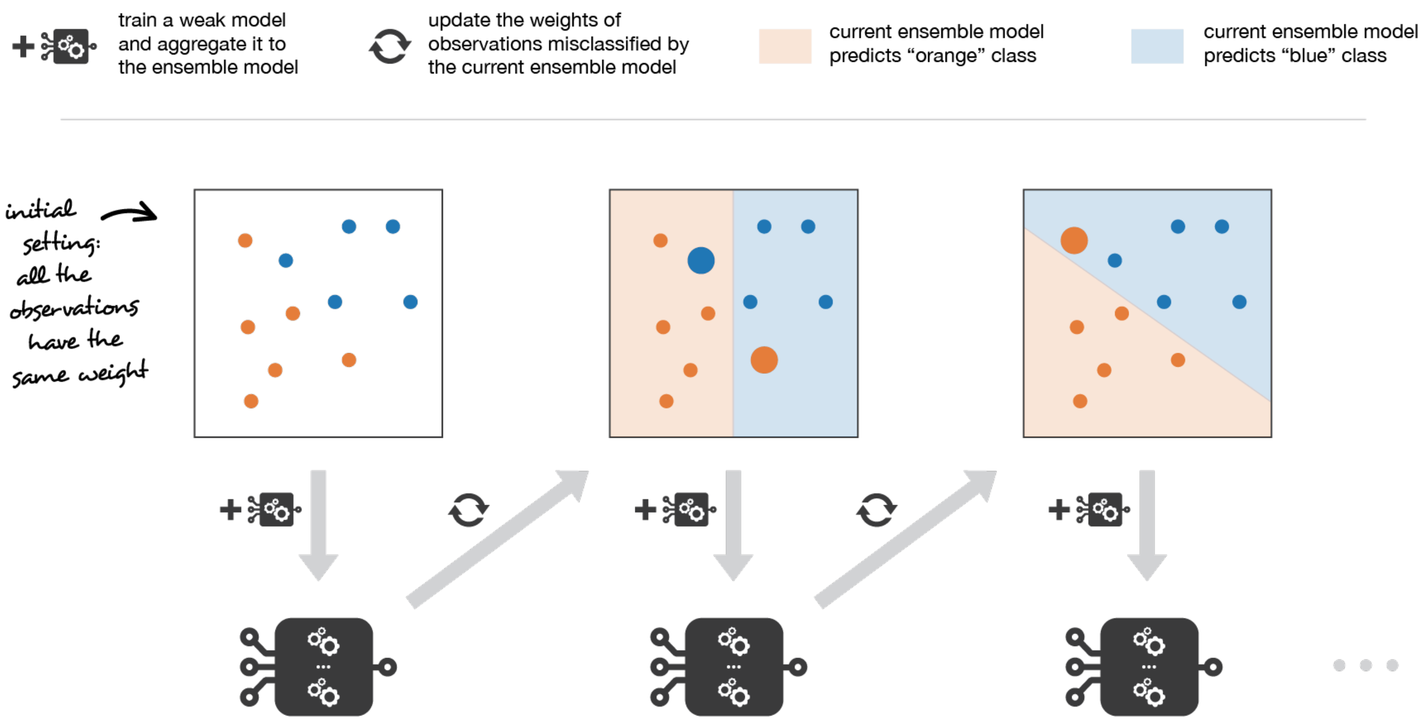


Boosting

In short, each model in the sequence is fitted giving more importance to observations in the dataset that were badly handled by the previous models in the sequence.

Boosting is used to create a collection of predictors. **In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analysing data for errors. Consecutive trees (random sample) are fit and at every step, the goal is to improve the accuracy from the prior tree.**

- Adaboost (Adaptative boosting)
- Gradient boosting

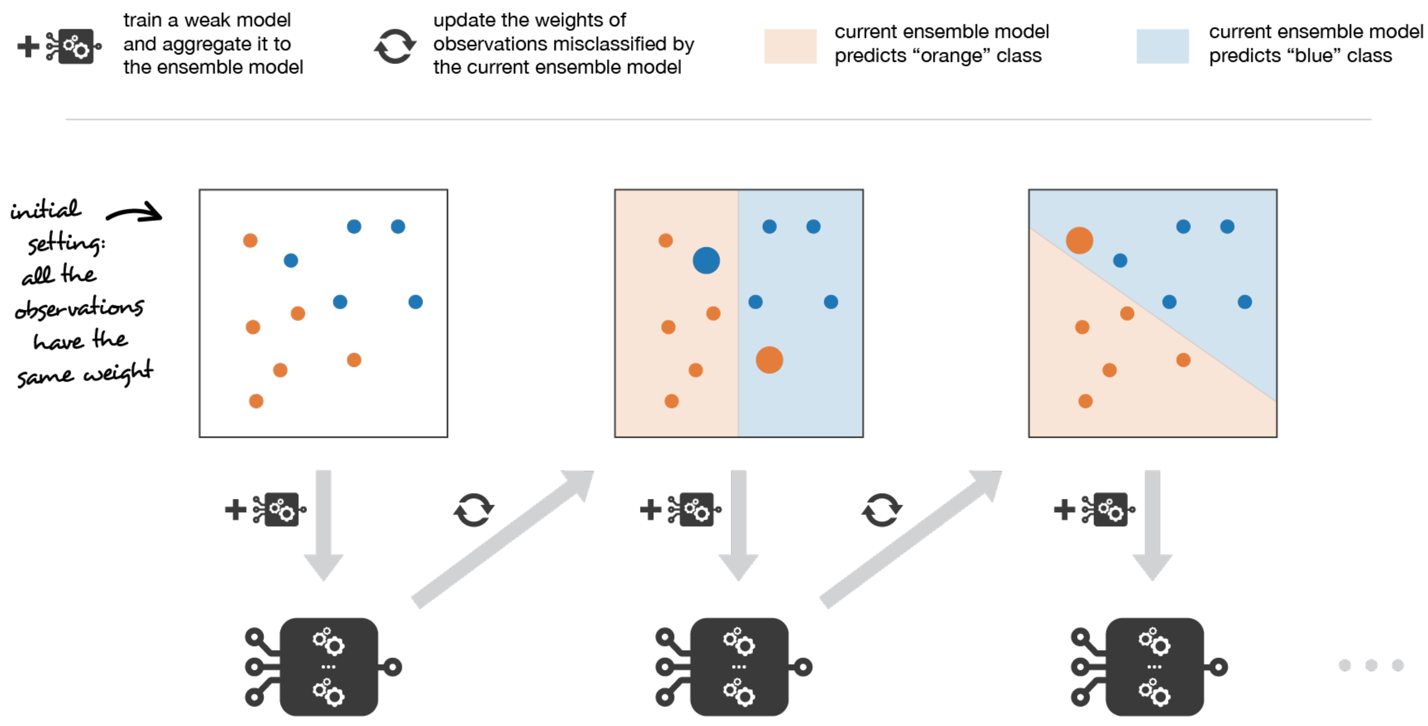


Adaboost

Adaboost updates weights of the observations at each iteration. Weights of well classified observations decrease relatively to weights of misclassified observations. Models that perform better have higher weights in the final ensemble model.

In short, each model in the sequence is fitted giving more importance to observations in the dataset that were badly handled by the previous models in the sequence.

- Adaboost (Adaptative boosting)
- Gradient boosting



Adaboost

Adaboost updates weights of the observations at each iteration. Weights of well classified observations decrease relatively to weights of misclassified observations. Models that perform better have higher weights in the final ensemble model.

In short, each model in the sequence is fitted giving more importance to observations in the dataset that were badly handled by the previous models in the sequence.

- Adaboost (Adaptative boosting)
- Gradient boosting

Code examples