I2ML :: CHEAT SHEET

The **I2ML**: Introduction to Machine Learning course offers an introductory and applied overview of "supervised" Machine Learning. It is organized as a digital lecture.

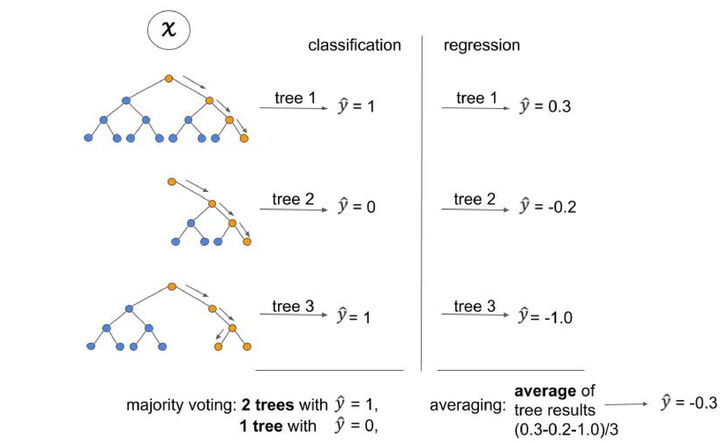
Tuning

Basic Idea:

- Bagging is short for **B**ootstrap **Agg**regation
- It's an **ensemble method**, i.e., it combines many models into one big "meta-model"
- The constituent models of an ensemble are called **base learners**
- All base learners are of the same type. The only difference between the models is the data they are trained on.

Bagging:

Base learners $b^{[m]}(x), m = 1, \ldots, M$ are trained on M **bootstrap** samples of training data $\mathcal{D}$. **Aggregate** the predictions of the M fitted base learners to get the **ensemble model** $\hat{f}^{[M]}(x)$.



Note:

Gains performance by reducing the variance of predictions, but (slightly) increases the bias: it reuses training data many times, so small mistakes can get amplified and Works best if base learners' predictions are only weakly correlated.

Random Forests

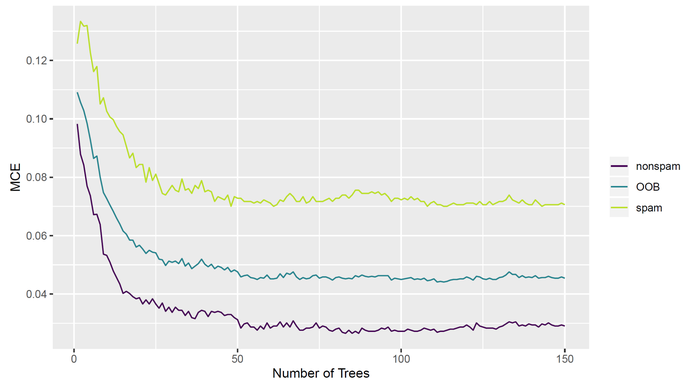
Introduction:

Modification of bagging for trees proposed by Breiman (2001):

- Tree baselearners on bootstrap samples of the data
- Uses **decorrelated** trees by randomizing splits (see below)
- Tree baselearners are usually fully expanded, without aggressive early stopping or pruning, to **increase variance of the ensemble**

Out of bag Error/ Out of bag estimate:

With the RF it is possible to obtain unbiased estimates of generalization error directly during training, based on the out-of-bag observations for each tree:



Feature Importance

- Single trees are highly interpretable
- Random Forests as ensembles of trees lose this feature
- Contributions of the different features to the model are difficult to evaluate
- Way out: variable importance measures
- Basic idea: by how much would performance of the random forest decrease if a specific feature were removed or rendered useless?

Algorithm 1 Measure based on improvement in split criterion

for features x_j, $j = 1$ to p **do**
  for tree base learners $\hat{b}^{[m]}(x)$, $m = 1$ to M **do**
    Find all nodes $\mathcal{N}$ in $\hat{b}^{[m]}(x)$ that use x_j.
    Compute improvement in splitting criterion achieved by them.
    Add up these improvements.
  end for
  Add up improvements over all trees to get feature importance of x_j.
end for

Algorithm 2 Measure based on permutations of OOB observations

While growing tree, pass down OOB observations and record predictive accuracy.
Permute OOB observations of j-th feature. This destroys the association between the target and the permuted j-th feature.
Pass down the permuted OOB observations and evaluate predictive accuracy again.
The decrease of performance induced by permutation is averaged over all trees and is used as a measure for the importance of the j-th variable.

Random Forest: Proximities

A measure of similarity ("closeness" or "nearness") of observations derived from random forests

Basic Idea:

- The proximity between two observations $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ is calculated by measuring the number of times that these two observations are placed in the same terminal node of the same tree of random forest, divided by the number of trees in the forest
- The proximity of observations $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ can be written as $\text{prox}\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)$
- The proximities form an intrinsic similarity measure between pairs of observations

Usage of random forest proximities:

Imputing missing data, locating outliers, identifying mislabeled data, visualizing the forest.

Synopsis

Hypothesis Space:
Random forest models are (sums of) step functions over rectangular partitions of (subspaces of) $\mathcal{X}$.

Risk:
Like trees, random forests can use any kind of loss function for regression or classification.

Optimization:
Exhaustive search over all (randomly selected!) candidate splits in each node of each tree to minimize the empirical risk in the child nodes.