Random forest

If there is a strong predictor in the dataset, the decision trees produced by each of the bootstrap samples in the bagging algorithm becomes very similar: Most of the trees will use the same strong predictor in the top split.

Random forests is a solution to this problem and a method for decorrelating the trees. The hope is to improve the variance reduction.

compound symmetry

$$\mathrm{Corr}\left(\hat{f}^{*b}(x), \hat{f}^{*l}(x)\right) = \underline{\rho}$$ for all trees l, b

$$\Rightarrow \mathrm{Var}\left(\frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)\right) = \underset{\uparrow}{\cdots} = \rho\sigma^2 + \frac{1-\rho}{B}\cdot\sigma^2$$

Exercise

$\mathrm{Var}(\hat{f}^{*b}(x)) = \sigma^2$

$B \to \infty$: $\rho\sigma^2$

$\rho = 0$: $\frac{\sigma^2}{B}$

$\rho = 1$: σ^2

Core modifications to bagging

The idea behind random forest is to *improve the variance reduction of bagging* by reducing the correlation between the trees - while hoping the possible increase in variance in each tree doesn´t cancel the improvement.

The procedure is thus as in bagging, but with the important difference, that

- at each split we are only allowed to consider $m < p$ of the predictors.

A new sample of m predictors is taken at each split and

- typically $m =$ floor$(\sqrt{p})$ (classificaton) and $m =$ floor$(p/3)$ (regression)

The general idea is at for very correlated predictors m is chosen to be small.

Got as far as here - will continue on 22.02.