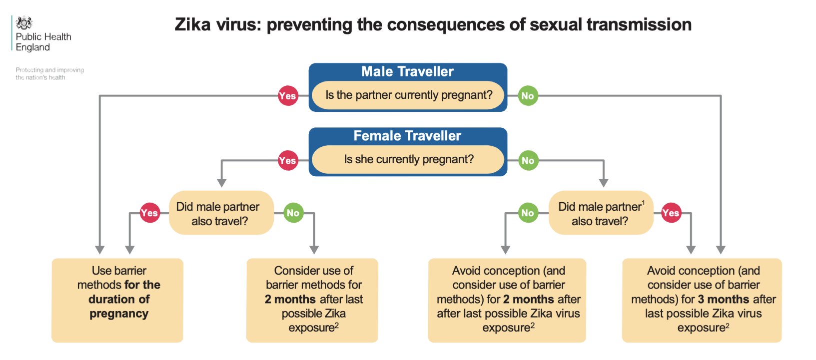
Decision Trees and Random Forest

Machine Learning
QSRI summer school – July 2022

Sarah Filippi

Department of Mathematics
Imperial College London

Many decisions are tree-structured



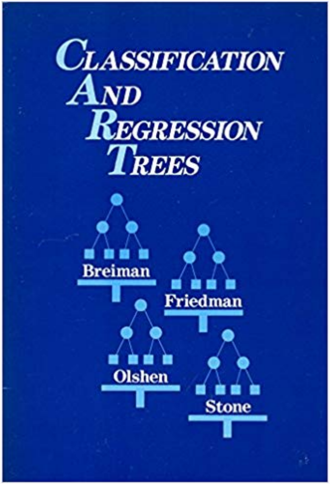
Outline

Tree-based methods

Ensemble methods

- Bagging
- Random Forests

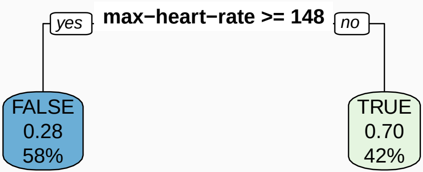
Long-standing, very effective method [Breiman et al 1984][1]



[1] $>$ 41K citations!

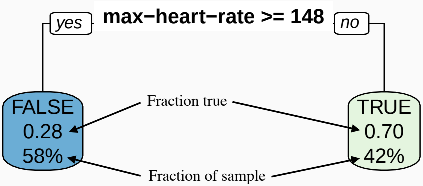
Decision trees

Example: predict heart disease based on age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), maximum heart rate achieved (rate)



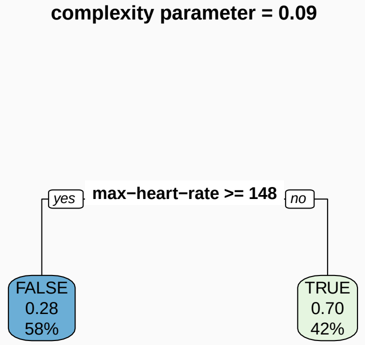
Decision trees

Example: predict heart disease based on age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), maximum heart rate achieved (rate)



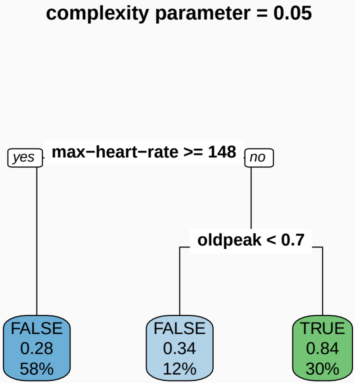
Decision trees

Example: predict heart disease based on age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), maximum heart rate achieved (rate)



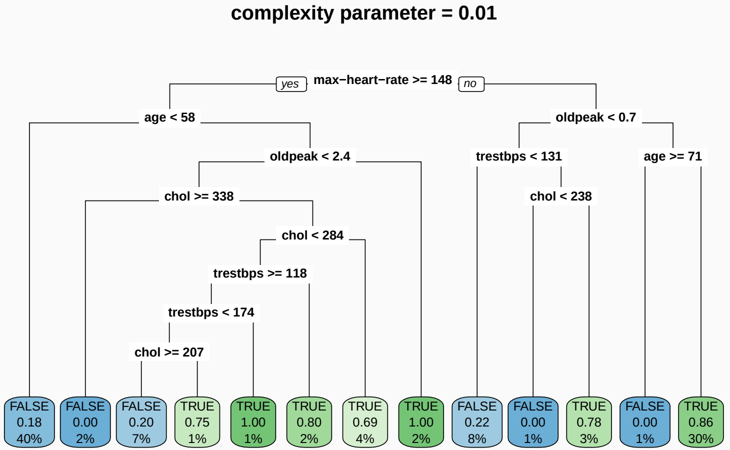
Decision trees

Example: predict heart disease based on age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), maximum heart rate achieved (rate)



Decision trees

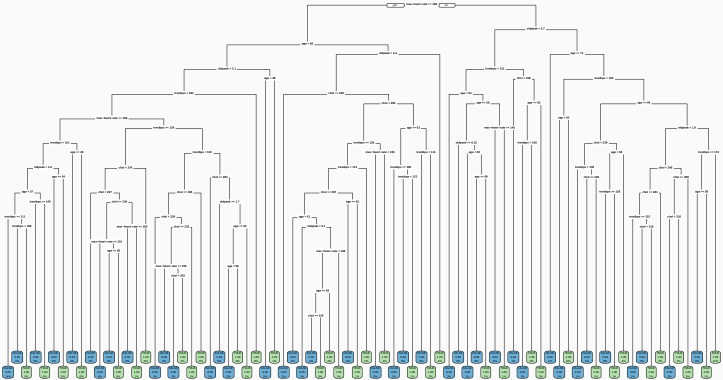
Example: predict heart disease based on age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), maximum heart rate achieved (rate)



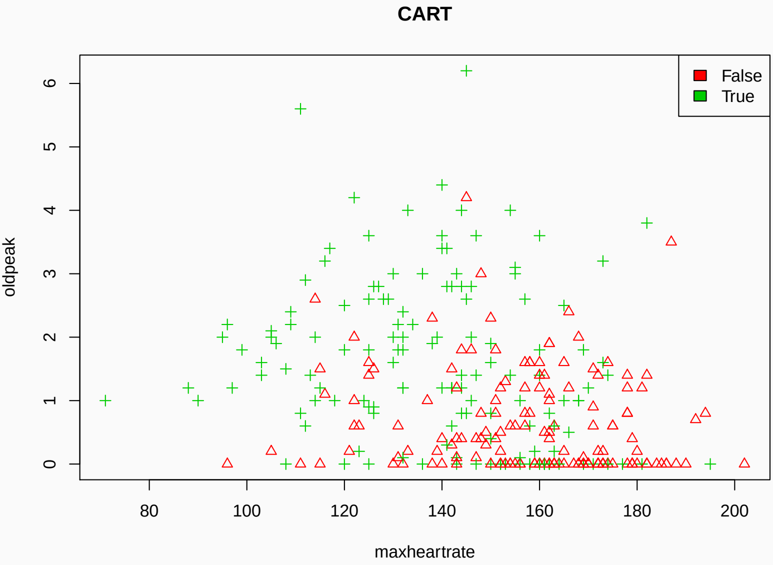
Decision trees

Example: predict heart disease based on age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), maximum heart rate achieved (rate)

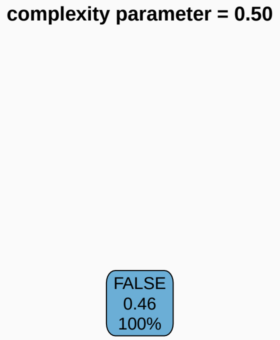
complexity parameter = 0.00

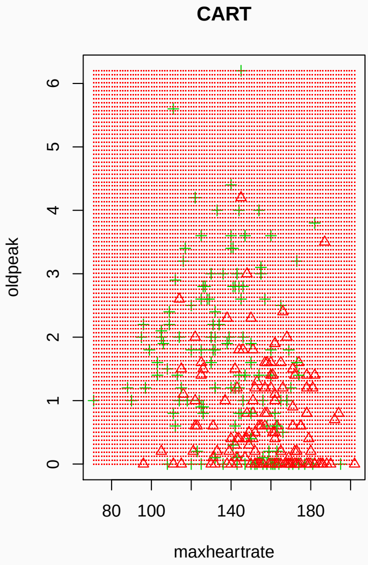


Decision boundaries of decision trees

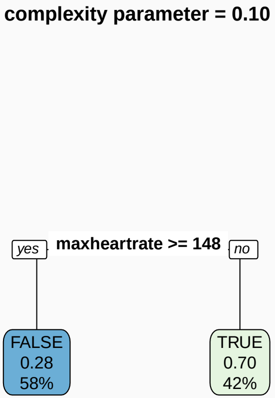


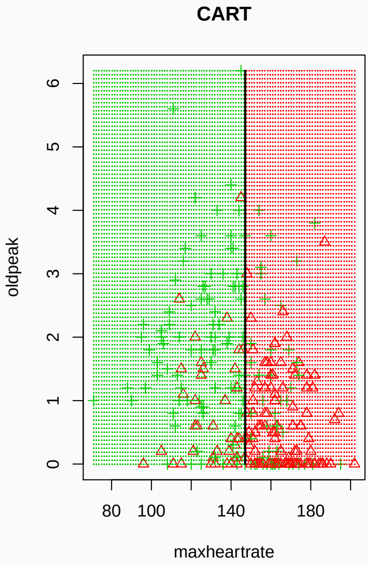
Decision boundaries of decision trees



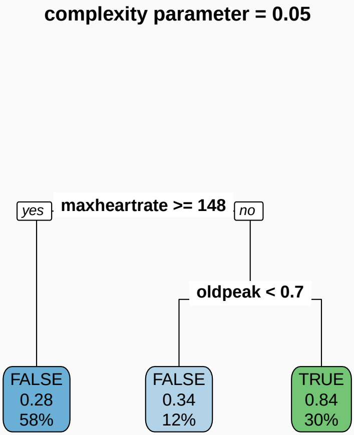


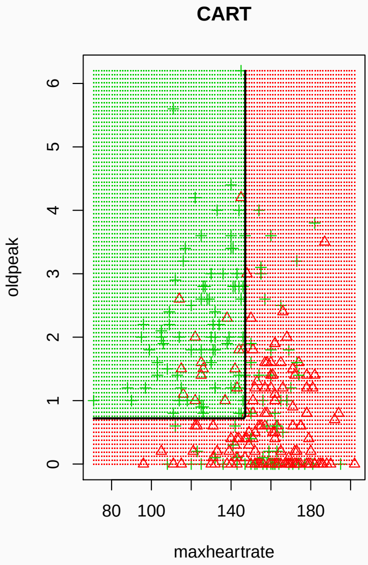
Decision boundaries of decision trees



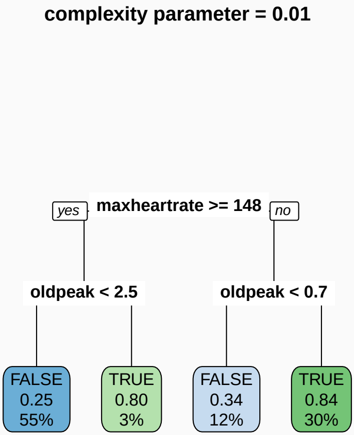


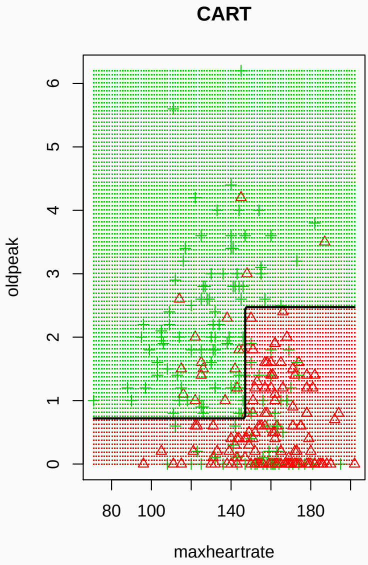
Decision boundaries of decision trees



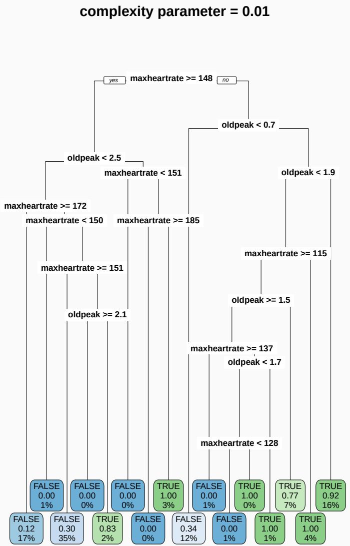


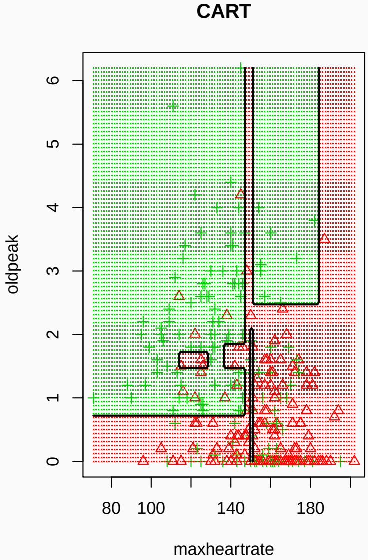
Decision boundaries of decision trees



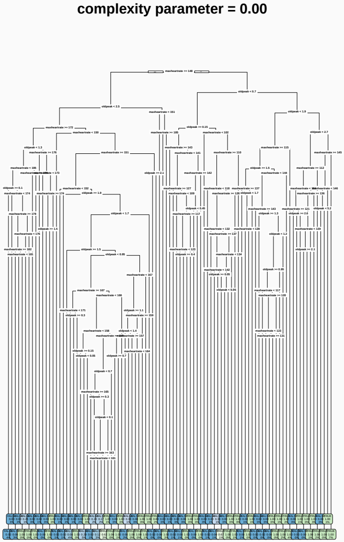


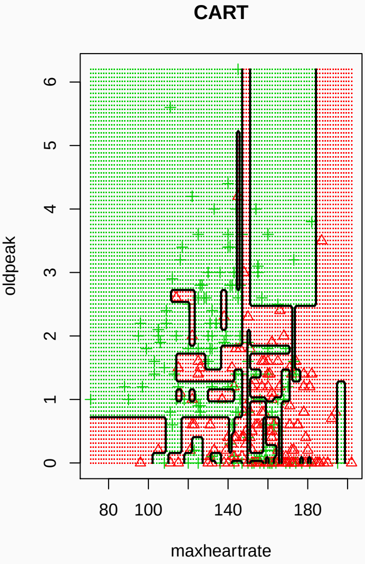
Decision boundaries of decision trees





Decision boundaries of decision trees





How to grow a decision tree

- Need a splitting rule!

How to grow a decision tree

- Need a splitting rule!
- To make a split: choose the best splitting variable x_p and cutpoint c based on the splitting rule.

How to grow a decision tree

- Need a splitting rule!
- To make a split: choose the best splitting variable x_p and cutpoint c based on the splitting rule.
- All choices are greedy.

How to grow a decision tree

- Need a splitting rule!
- To make a split: choose the best splitting variable x_p and cutpoint c based on the splitting rule.
- All choices are greedy.
- Example of a splitting rule: maximize the "information gain", calculated as:

$$\begin{aligned}\mathrm{IG}(x_p, c) = &\text{ information before splitting} \\ &- \text{information after splitting on } x_p < c\end{aligned}$$

- Information gain tells us how useful a given variable of the feature vectors is for discriminating between the classes to be learnt.

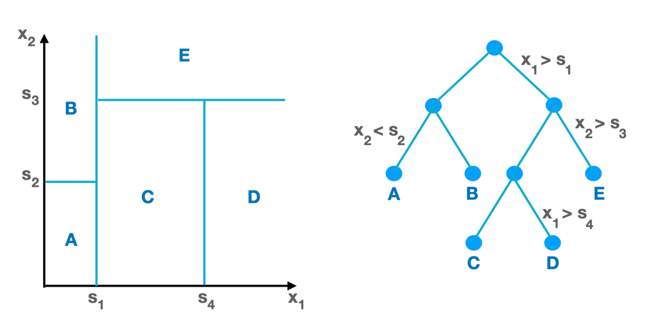
How to grow a decision tree

- Need a splitting rule!
- To make a split: choose the best splitting variable x_p and cutpoint c based on the splitting rule.
- All choices are greedy.
- Example of a splitting rule: maximize the "information gain", calculated as:

$$\begin{aligned} \mathrm{IG}(x_p, c) = &\text{ information before splitting} \\ &- \text{information after splitting on } x_p < c \end{aligned}$$

- Information gain tells us how useful a given variable of the feature vectors is for discriminating between the classes to be learnt.
- Calculate information with Gini Index or Entropy (classification), Squared Error (regression)

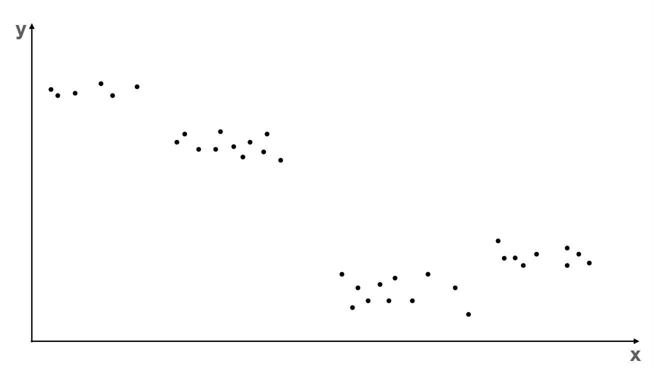
Regression trees



The main difference between regression and classification trees is the criteria used for splitting the tree. For regression the impurity measure of a region R_m of a tree T is

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i^{(i)} \in R_m} (y^{(i)} - \hat{c}_m)^2 \, .$$

Example of regression tree



How large should we grow the tree?

Tree size is a tuning parameter governing model's complexity:

- very large tree might overfit the data
- small tree might not capture the important structure.

The maximum depth of a tree is a hyper-parameter that should be tuned. Common strategy:

1. grow a large tree T_0, stopping the process splitting when some minimum node size is reached
2. prune the tree using *cost-complexity pruning*.

Summary

Advantanges:

- interpretable by non-experts
- simple to apply to many types of data (real/categorical inputs)
- building block for various ensemble methods (see later)

Disadvantanges:

- prone to overfitting
- finding partition of feature space that minimizes empirical error is computationally intractable – we have to use greedy approaches with limited theoretical underpinning
- unstable: small changes in input data lead to different trees

Outline

Bagging and Random Forests

- Bagging and Random forests are examples of an ensemble method.

Bagging and Random Forests

- Bagging and Random forests are examples of an ensemble method.
- Like wisdom of crowds, you average together many predictors.
  Diversity helps to reduce overfitting!

Bagging and Random Forests

- Bagging and Random forests are examples of an ensemble method.
- Like wisdom of crowds, you average together many predictors. Diversity helps to reduce overfitting!
- Build a collection of trees using **random** datasets.

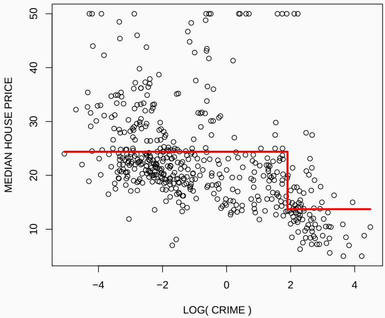
Bagging and Random Forests

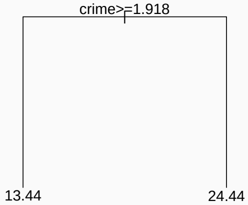
- Bagging and Random forests are examples of an ensemble method.
- Like wisdom of crowds, you average together many predictors. Diversity helps to reduce overfitting!
- Build a collection of trees using **random** datasets.
- Random datasets built with the bootstrap.

Outline

Example: Bootstrap for Regression Trees

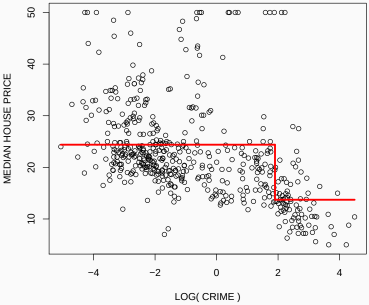
- Regression for Boston housing data.
- Aim: predict median house prices based only on crime rate.
- Consider a tree with a single split at the root.

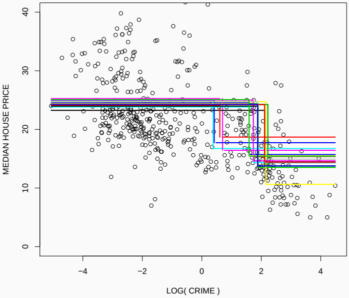




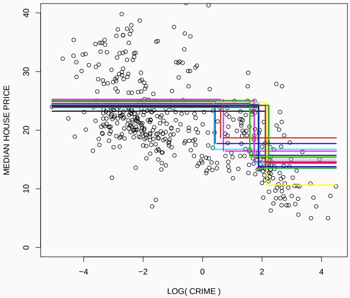
Bootstrap for Regression Trees

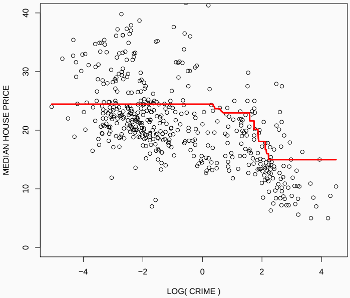
- Consider a tree with a single split at the root.
- Is the prediction "stable" if training data were slightly different?
- Fit trees on 20 bootstrap samples, i.e. samples with replacement of size n of original data





Bagging



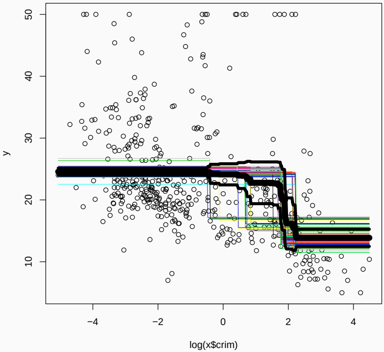


- Bagging takes the average over the predictions of the 20 trees.
- Bagging smooths out the drop in the estimate of median house prices.
- Bagging reduces the variance of predictions.

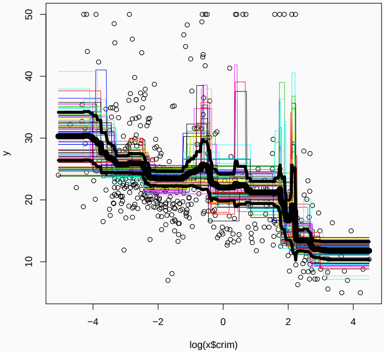
Variance Reduction in Bagging

Deeper trees have higher complexity and variance.

Trees of depth 1



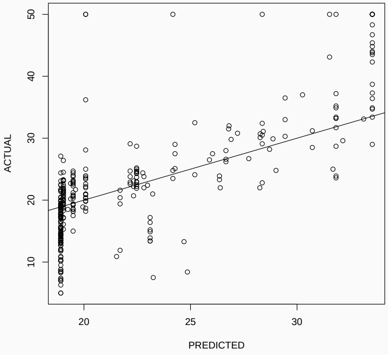
Trees of depth 3



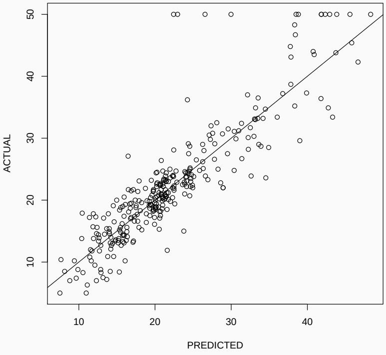
Example: Boston Housing Dataset

- Apply out of bag test error estimation to select optimal tree depth and assess performance of bagged trees for Boston Housing data.
- Use the entire dataset with $p = 13$ predictor variables.

For depth $d = 1$.



For depth $d = 10$.



Example: Boston Housing Dataset

- Test error as a function of tree depth d:

tree depth d	1	2	3	4	5	10	30
single tree $\hat{f}$	60.7	44.8	32.8	31.2	27.7	26.5	27.3
bagged trees $\hat{f}_{Bag}$	43.4	27.0	22.8	21.5	20.7	20.1	20.1

- Without bagging, the optimal tree depth seems to be $d = 10$.
- With bagging, we could also take the depth up to $d = 30$.

Summary:

- Bagging reduces variance and prevents overfitting
- Often improves accuracy in practice.
- Bagged trees cannot be displayed as nicely as single trees and some of the interpretability of trees is lost.

Outline

Random Forests

- *Random forests* are similar to bagged decision trees with a few key differences.
- Build a collection of trees using **random** datasets and **random** predictors.
- Random datasets built with the bootstrap.
- Random predictors: at each split point, search over *mtry* randomly chosen predictors.
- Random forests tend to produce better predictions than bagging.
- Implemented in `randomForest` library in `R`.

Random Forests

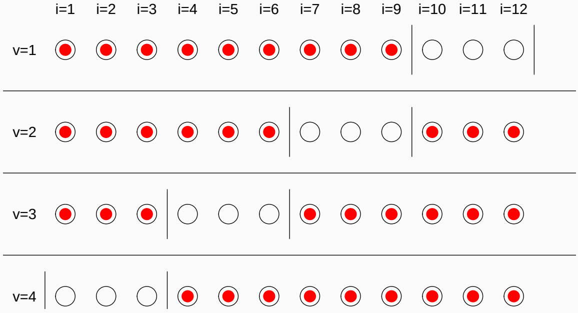
The main hyperparameters of randomForest are :

- mtry: the number of features considered at each split point
- ntree: the number of trees in the forest.
    - Increasing the number of trees reduce the variance of the full model, i.e. gives more stable results.
    - Computational cost if linear in the number of trees.
- nodesize: the maximum number of examples that are allowed at each leaf node.
    - Lowest value of nodesize constructs trees with only a single example at each leaf – likely to overfit the training data.
    - The other extreme leads to decision stumps (trees with only one level), which are likely underfit the training data.
- maxnodes: the maximum number of allowed leaf nodes in each tree.

How do we learn these hyperparmeters?

Performance and cross-validation

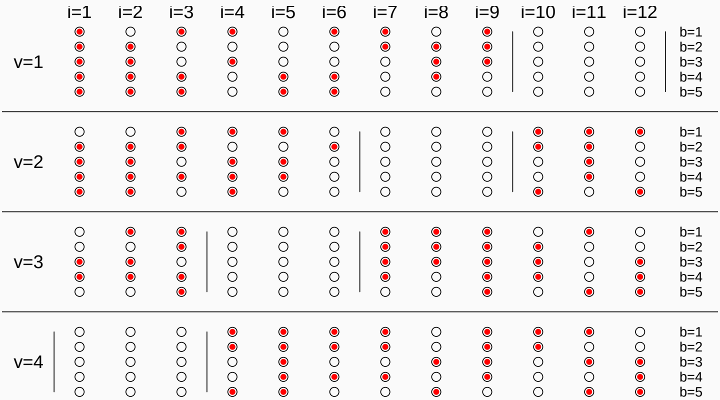
In Machine Learning, to estimate performance or tune hyper-parameters, we typically use cross-validation.



- For each fold, fit $\hat{f}_{Bag}$ on the training samples and predict on the validation set;
- Compute the cross-validation error by averaging the loss across all test observations

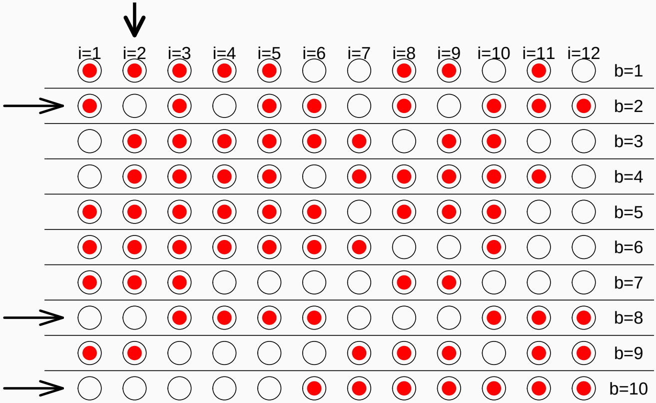
Performance and cross-validation

But to fit $\hat{f}_{Bag}$ on the training samples, we have to fit a tree on B bootstrap samples!



Out-of-bag Test Error Estimation

Idea: test on the "unused" data points in each bootstrap iteration to estimate the test error.



$$\text{OOB} = \sum_{i=1}^{12} \hat{f}^{\text{oob}}(x_i) \quad \text{where e.g.} \quad \hat{f}^{\text{oob}}(x_2) = \frac{1}{3} \sum_{b \in \{2,8,10\}} \hat{f}^{b}(x_2)$$

Variable importance

Tree ensembles have better performance, but decision trees are more interpretable.

How to interpret a forest of trees?

Approach 1:
Calculate the total amount that the MSE or Gini index is decreased due to splits over a given predictor, averaged over all B trees.

See for example section 10.13 of Hastie, Tibshirani and Friedman's book.

Variable importance

Approach 2:

Denote by $\hat{e}^{(b)}$ the out-of bag estimate of the loss for tree b. For each variable $k \in \{1, ..., p\}$,

1. permute randomly the k-th predictor variable to generate a new set of samples $(\tilde{x}^{(1)}, y^{(1)}), \ldots (\tilde{x}^{(n)}, y^{(n)})$, where $\tilde{x}_k^{(i)} = x_k^{(\tau(i))}$ for a permutation $\tau(\cdot)$.
2. compute the out-of-bag estimate $\hat{e}_k^{(b)}$ of the prediction error with these new samples.

A measure of importance of variable k is then the increase in error rate due to a random permutation of the k-th variable

$$\frac{1}{B} \sum_{b=1}^{B} \hat{e}_k^{(b)} - \hat{e}^{(b)} \ .$$

Decision Trees and Ensemble Methods: Conclusion

See demo `DecisionTreeRF.Rmd`

- Decision trees are very interpretable, but prone to overfitting
- Bagging and random forests are examples of **ensemble methods**, where predictions are based on an ensemble of many individual predictors.
- Bagging and random forests are typically less interpretable than decision trees.
- Many other ensemble learning methods: boosting, stacking, mixture of experts, BART, Bayesian model combination, Bayesian model averaging etc.
- Often gives significant boost to predictive performance, at the expense of interpretability.