Lecture 4: Rule-based classification and regression

Felix Held, Mathematical Sciences

MSA220/MVE441 Statistical Learning for Big Data

12th April 2021

Classification and Partitions

Classification and Partitions

A classification algorithm constructs a partition of feature space and assigns a class to each.

- kNN creates local neighbourhoods in feature space and assigns a class in each
- Logistic regression divides feature space implicitly by modelling $p(i|\mathbf{x})$ and determines decision boundaries through Bayes' rule
- Discriminant analysis creates an explicit model of the feature space conditional on the class. It models $p(\mathbf{x}, i)$ by assuming that $p(\mathbf{x}|i)$ is a normal distribution and either estimates $p(i)$ from data or through prior knowledge.

New point-of-view: Partitioning

Idea: Create an **explicit partition** by dividing feature space into **non-overlapping regions** and assign a **constant conditional mean** (regression) or **constant conditional class probability** (classification) to each region.

Given regions R_m for $m = 1, \dots, M$, a classification rule for classes $i \in \{1, \dots, K\}$ is

$$\hat{c}(\mathbf{x}) = \underset{1 \leq i \leq K}{\arg\max} \sum_{m=1}^{M} \mathbb{1}(\mathbf{x} \in R_m) \left(\frac{1}{|R_m|} \sum_{\mathbf{x}_l \in R_m} \mathbb{1}(i_l = i) \right)$$

and a regression function is given by

$$\widehat{f}(\mathbf{x}) = \sum_{m=1}^{M} \left(\frac{1}{|R_m|} \sum_{\mathbf{x}_l \in R_m} y_l \right) \mathbb{1}(\mathbf{x} \in R_m).$$

Note that $|R_m|$ denotes the number of elements in R_m or its size.

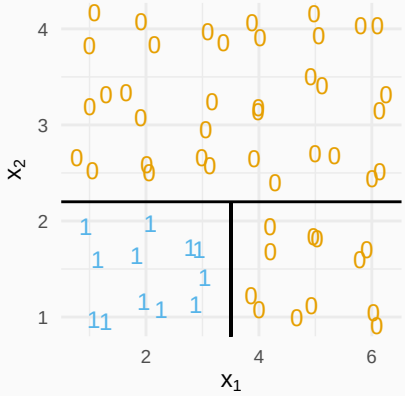
Complexity of partitioning:

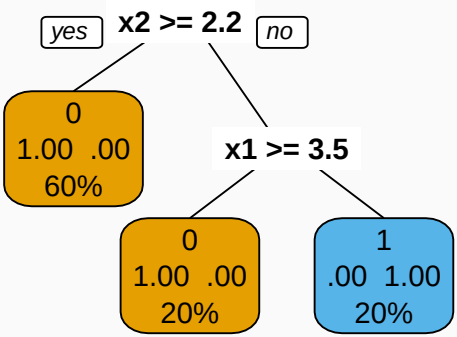
Arbitrary Partition > Rectangular Partition > Partition from a sequence of binary splits

Classification and Regression Trees (CART)

- Create a **sequence of binary axis-parallel splits**
- in order to **reduce variability** of values/classes in each region

Example of classification with CART





CART: Tree building/growing

1. Start with all data in a **root node**
2. Binary splitting
   2.1 Consider each feature $x_{\cdot j}$ for $j = 1, \ldots, p$. Choose a **threshold** t_j (for continuous features) or a **partition of the feature categories** (for categorical features) that results in the greatest improvement in **node purity**:

$$\{i_l : x_{lj} > t_j\} \quad \text{and} \quad \{i_l : x_{lj} \leq t_j\}$$

   2.2 Choose the feature j that led to the best splitting of the data and create a new **child node** for each subset
3. Repeat Step 2 on all child nodes until the tree reaches a **stopping criterion**

All nodes without descendents are called **leaf nodes**. The sequence of splits preceding them defines the regions R_m.

Measures of node purity

Define

$$\widehat{\pi}_{im} := \frac{1}{|R_m|} \sum_{\mathbf{x}_l \in R_m} \mathbb{1}(i_l = i)$$

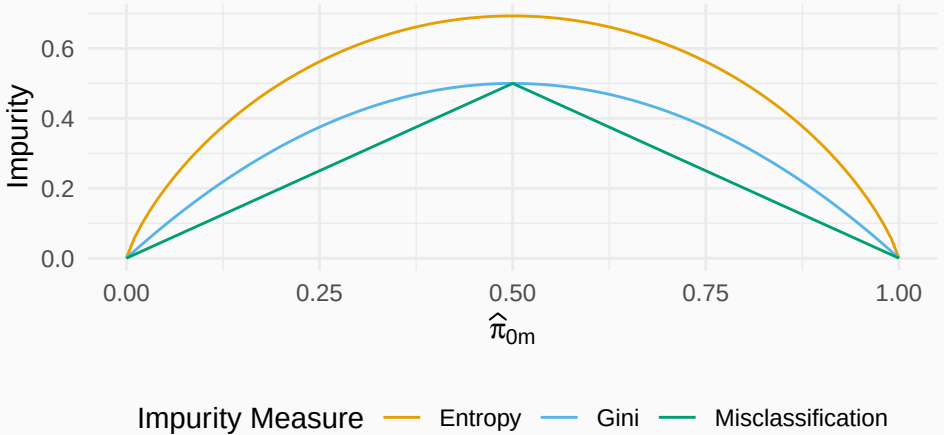
- Three common measures to determine impurity in a region R_m are (for classification trees)

Misclassification error:	$1 - \max_i \widehat{\pi}_{im}$
Gini impurity:	$\sum_{i=1}^{K} \widehat{\pi}_{im}(1 - \widehat{\pi}_{im})$
Entropy/deviance:	$-\sum_{i=1}^{K} \widehat{\pi}_{im} \log \widehat{\pi}_{im}$

- All criteria are zero when only one class is present and maximal when all classes are equally common.
- For regression trees the **decrease in mean squared error** after a split can be used as an impurity measure.

Node impurity in two class case

Example for a two-class problem ($i = 0$ or 1). $\hat{\pi}_{0m}$ is the empirical frequency of class 0 in a region R_m.



Stopping criteria

Examples

- Minimum size of leaf nodes (e.g. 5 samples per leaf node)
- Minimum decrease in impurity (e.g. cutoff at 1%)
- Maximum tree depth, i.e. number of splits (e.g. maximum of 30 splits from root node)
- Maximum number of leaf nodes

Running CART until one of these criteria is fulfilled generates a **max tree**.

Observations about CART

- **Pro:** Outcome is easily interpretable
- **Pro:** Can easily handle missing data
- **Neutral:** Only suitable for axis-parallel decision boundaries
- **Con:** Features with more potential splits have a higher chance of being picked
- **Con:** Prone to overfitting/unstable (only the best feature is used for splitting and which is best might change with small changes of the data)

CART and overfitting

How can overfitting be avoided?

Tuning of stopping criteria

Beware: This can lead to early stopping since a weak early split might lead to a strong split later

Pruning

- Build a max tree first.
- Then reduce its size by collapsing internal nodes.

Principle: 'The silly certainty of hindsight'

Ensemble methods

Examples are bagging, boosting, stacking, …

Pruning

A common strategy is **cost-complexity pruning**.

- For a given $\alpha > 0$, the cost-complexity of a tree T is defined as

$$C_\alpha(T) = \underbrace{\sum_{R_m \in T} \left(\frac{1}{|R_m|} \sum_{\mathbf{x}_l \in R_m} \mathbb{1}(i_l \neq \hat{c}(\mathbf{x})) \right)}_{\text{Cost}} + \underbrace{\alpha |T|}_{\text{Complexity}}$$

  where $(i_l, \mathbf{x}_l)$ is the training data, $\hat{c}$ the CART classification rule and $|T|$ is the number of leaf nodes/regions defined by the tree.
- It can be shown that successive subtrees T_k (i.e. $T_k \subset T_{k-1}$) of the max tree $T_0 = T_{\max}$ can be found such that each tree T_k minimizes $C_{\alpha_k}(T_k)$ where $\alpha_0 = 0 < \alpha_1 < \cdots < \alpha_J$.
- Note that for $\alpha_0 = 0$ the cost is minimized for $T_{\max}$ and the subtree minimizing cost-complexity with α_J consists of only the root node.

Pruning: Choosing the best subtree

To choose the best subtree, one either needs

- to have access to a **test set**, or
- perform **cross-validation**
  1. Determine the max tree on the full training data and perform cost-complexity pruning to get a sequence of α_k.
  2. Split the data into folds and for each fixed test fold build a max tree on all remaining folds and perform cost-complexity pruning on these max trees.
  3. For each α_k from Step 1 compute the test error on each fold.
  4. Choose subtree built in Step 1 with minimal test error in Step 3.

Note: Full dataset is only used before CV to create candidates for α_k.

Recap of the bootstrap

The Bootstrap – A short recap (I)

Given a sample x_i, $i = 1, \dots, n$ from an underlying population estimate a statistic θ by $\hat{\theta} = \hat{\theta}(x_1, \dots, x_n)$.

What is the uncertainty of $\hat{\theta}$?

Solution: Find confidence intervals (CIs) quantifying the variability of $\hat{\theta}$.

Computation:

- Through theoretical results (e.g. linear models) if distributional assumptions fulfilled
- Linearisation for more complex models (e.g. nonlinear or generalized linear models)
- Nonparametric approaches using the data (e.g. **bootstrap**)

All of these approaches require fairly large sample sizes.

The Bootstrap – A short recap (II)

Nonparametric bootstrap

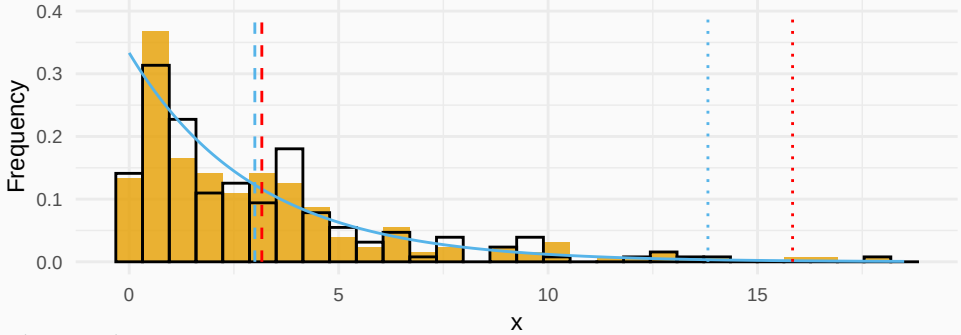
Given a sample $x_1, \dots, x_n$ **bootstrapping** performs for $b = 1, \dots, B$

1. Sample $\tilde{x}_1, \dots, \tilde{x}_n$ with replacement from original sample
2. Calculate $\hat{\theta}_b(\tilde{x}_1, \dots, \tilde{x}_n)$

- B should be large (in the 100–1000s)
- The distribution of $\hat{\theta}_b$ approximates the sampling distribution of $\hat{\theta}$
- The bootstrap makes **exactly one strong assumption**:
  The data is discrete and values not seen in the data are impossible.[1]

[1] Check out this blog post!

CI for statistics of an exponential random variable



Data (n = 200) simulated from $x \sim \mathrm{Exp}(1/3)$, i.e. $\mathbb{E}_{p(x)}[x] = 3$

- Orange histogram shows original sample
- Blue line is the true density
- Black outlined histogram shows a bootstrapped sample
- Vertical lines are the mean of x (dashed) and the 99% quantile (dotted) [red = empirical, blue = theoretical]

CI calculation: Normal approximation and percentile method

1. **Normal approximation:** Set $\overline{\theta} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b$ and estimate the standard error of $\hat{\theta}$ as

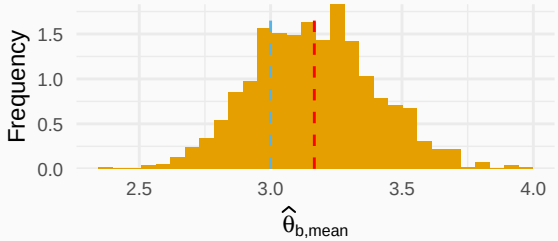
$$\widehat{\sigma}_{se} = \sqrt{\frac{\sum_{b=1}^{B}(\hat{\theta}_b - \overline{\theta})^2}{B-1}}$$

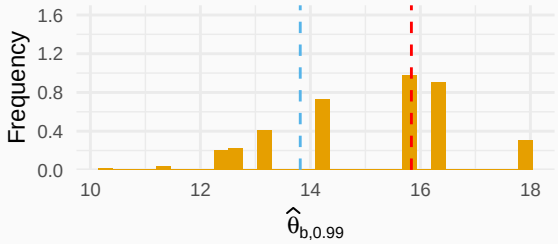
   Assume the distribution of $\hat{\theta}$ is approximately $N(\hat{\theta}, \widehat{\sigma}_{se})$ giving CI

$$\hat{\theta} \pm z_{1-\alpha/2}\widehat{\sigma}_{se}$$

2. **Percentile/quantile method:** Take the α and $\alpha/2$ quantiles of the bootstrap estimates $\hat{\theta}_b$ as boundaries of CI

CI calculation: Applied to example





Based on $B = 1000$ bootstrap samples

For the mean value, normal approximation assumption seems reasonable

95% CIs
Normal Approx. (2.71, 3.62)
Perc. Method (2.73, 3.64)

For the quantile, bootstrapping requires much larger n and shows high uncertainty

Modifications to nonparametric bootstrap

- Different sampling strategies. Some examples:
  - m-out-of-n bootstrap: Draw $m < n$ samples without replacement
  - Draw from a smooth density estimate of the data
  - Draw from a parametric distribution fitted to the original data
- Normal approximation doesn't always apply and percentile method is unstable for complicated statistics. Example of alternative
  - Bootstrap-t: Instead of normal quantiles, estimate quantiles from

$$\frac{\hat{\theta}_b - \hat{\theta}}{\widehat{\sigma}_b}$$

    where $\widehat{\sigma}_b$ is an estimate of the standard error
  - Many other alternatives exist ...

Limitations of the bootstrap

- Number of samples needs to be quite large
- Extreme values (minimum, maximum very small or large quantiles) can be hard to estimate since they might not even appear in data
- Many basic CI estimation algorithms assume that the bootstrap distribution is approximately normal (often not the case in reality)

Bootstrap aggregation

Bootstrap aggregation (bagging)

1. Given a training sample $(y_l, \mathbf{x}_l)$ or $(i_l, \mathbf{x}_l)$, we want to fit a predictive model $\widehat{f}(\mathbf{x})$
2. For $b = 1, \ldots, B$, form bootstrap samples of the training data and fit the model, resulting in $\widehat{f}_b(\mathbf{x})$
3. Define

$$\widehat{f}_{\text{bag}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}_b(\mathbf{x})$$

   where $\widehat{f}_b(\mathbf{x})$ is a **continuous value for a regression problem** or a **vector of class probabilities** for a classification problem

Majority vote can be used for classification problems instead of averaging

Bagging and variance reduction

- Bagging using averages approximates

$$f_{\text{ag}}(\mathbf{x}) = \mathbb{E}_{p(\mathcal{T})}\left[\widehat{f}(\mathbf{x})\right]$$

- If a squared error loss is used, the following relation holds

$$\mathbb{E}_{p(\mathcal{T},y|\mathbf{x})}[(y - \widehat{f}(\mathbf{x}))^2] \geq \mathbb{E}_{p(\mathcal{T},y|\mathbf{x})}[(y - f_{\text{ag}}(\mathbf{x}))^2]$$

  which implies that total prediction error for the averaged estimator is lower.
- Some notes:
  - Remember the graphs of kNN from last lecture: Noisy individually, more stable (less variable) on average
  - Bagging shows no effect on linear models

Correlation and bagged variance

For identically distributed (i.d.) random variables x_i, $i = 1, \ldots, n$

$$\mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) = \frac{1-\rho}{n}\sigma^2 + \rho\sigma^2$$

where $\rho \in [0, 1)$ is the (positive) pairwise correlation coefficient and σ^2 is the variance of each x_i.

- Bootstrap samples, and therefore the resulting estimators $\widehat{f}_b(\mathbf{x})$, are correlated
- By letting $B \to \infty$ we can bring the first term towards zero, but the second term remains
- Decreasing correlation between bootstrap samples would decrease the variance of a bagging estimate

Random Forests

Random Forests

1. Given a training sample with p features, do for $b = 1, \dots, B$
   - 1.1 Draw a bootstrap sample of size n from training data (with replacement)
   - 1.2 Grow a tree T_b until each node reaches minimal node size $n_{\min}$
     - 1.2.1 Randomly select m variables from the p available
     - 1.2.2 Find best splitting variable among these m
     - 1.2.3 Split the node
2. For a new $\mathbf{x}$ predict

$$\begin{aligned} \text{Regression:} \quad & \widehat{f}_{rb}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} T_b(\mathbf{x}) \\ \text{Classification:} \quad & \text{Majority vote at } \mathbf{x} \text{ across trees} \end{aligned}$$

Note: Step 1.2.1 leads to less correlation between trees built on bootstrapped data.

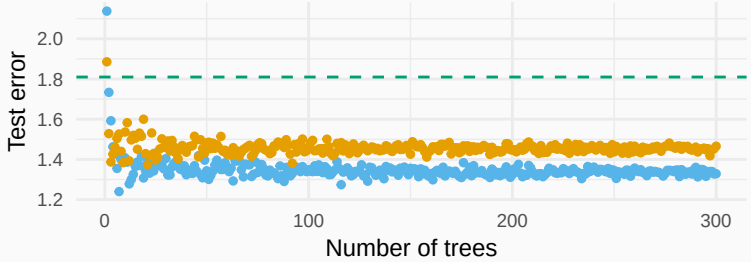
Comparison of RF, Bagging and CART

Toy example

$$y = x_1^2 + \varepsilon \quad \text{where} \quad \varepsilon \sim N(0, 1)$$

$$\mathbf{x} \sim N(\mathbf{0}, \mathbf{\Sigma}),\ \mathbf{x} \in \mathbb{R}^5, \quad \mathbf{\Sigma}_{ll} = 1, \mathbf{\Sigma}_{lk} = 0.98, l \neq k$$

Training and test data were sampled from the true model. Results for RF, bagged CART and a single CART, using $x_1, \ldots, x_5$ as predictor variables. ($n_{tr} = 50$, $n_{te} = 100$)



Variable importance

1. **Impurity index:** Splitting on a feature leads to a reduction of node impurity. Summing all improvements over all trees per feature gives a measure for variable importance
2. **Out-of-bag error**
   - During bootstrapping for large enough n, each sample has a chance of about 63% to be selected
   - For bagging the remaining samples are **out-of-bag**.
   - These out-of-bag samples for tree T_b can be used as a test set for that particular tree, since they were not used during training. Call the resulting test error E_0
   - Permute variable j in the out-of-bag samples and calculate test error again $E_1^{(j)}$
   - The increase in error
   $$E_1^{(j)} - E_0 \geq 0$$
   serves as an importance measure for variable j

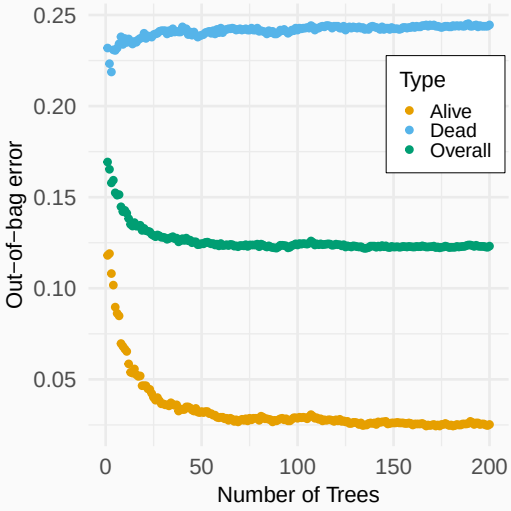
Monica dataset

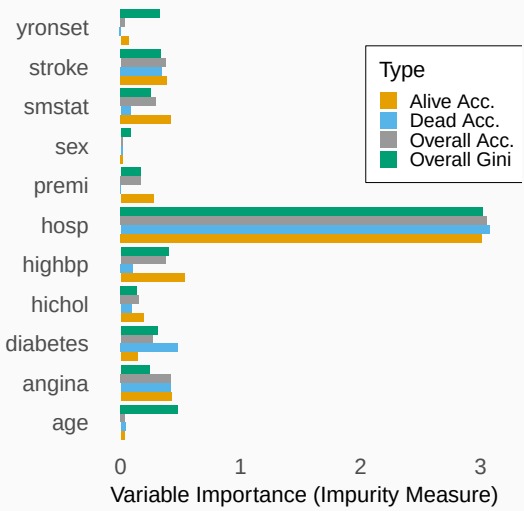
Monica dataset[2] Data from the WHO project 'Multinational MONItoring of trends and determinants in CArdiovascular disease'

- Observations on whether or not patients survive a 10 year period given a number of cardiovascular risk factors
- Collected from the 1970s to the 1990s
- $n = 6367$ samples (3525 alive, 2842 dead)
- $p = 11$ features
  - e.g. sex, age at onset, year of onset, hospitalisation status, cholesterol, blood pressure, ...

[2] http://thl.fi/monica

RF applied to cardiovascular dataset





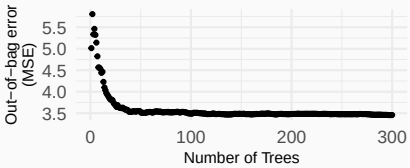
South African coronary heart disease (SAheart) dataset

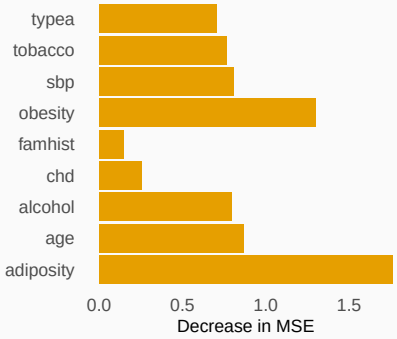
South African coronary heart disease (SAheart) dataset A retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa.

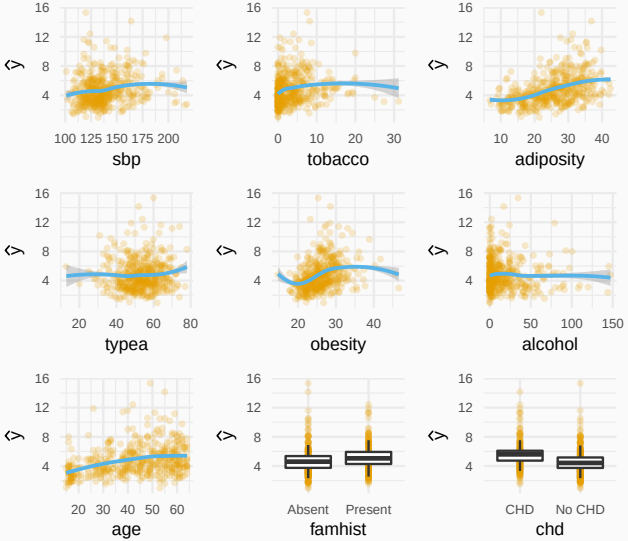
- Meant as a classification data set with response whether or not coronary heart disease was diagnosed
- $n = 462$ samples (160 diagnosed, 302 not diagnosed)
- $p = 9$ features
  - e.g. cumulative tobacco consumption (in kg), low density lipoprotein cholesterol (`ldl`), adiposity, family history, ...

To demonstrate random forests for regression, we will try to predict `ldl` from the other features and response.

RF applied to heart disease dataset







Take-home message

- Direct partitioning of feature space is a complex task
- Binary splits resulting in simple tree models
- CART is highly interpretability, but very instable/variable
- Random Forests introduce variance reduction to bagging and allow to measure variable importance