07/02/2019

Advanced Regression SPH024

Lect. 4c: Tree based methods

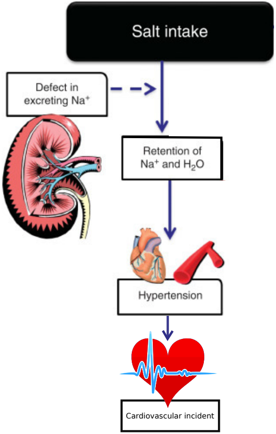
Mentimeter access: www.menti.com 31 56 08

So far

1. Linear models, OLS, MLE...
2. Variable importance selection
3. High dim. analysis and regularisation
4. Mixed effects / hierarchical models
5. Non-linear models

$\longrightarrow$ All parametric models: pre-specifed relationships between X and Y

UKB case study

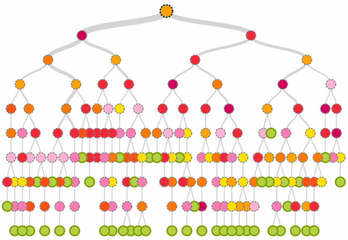


Case study: urinary sodium vs CVD in UKBiobank

- Outcomes: Cardiovascular incidents Systolic blood pressure
- Exposure: Na^+
- Confounders: Age, Sex, K, BMI

Decision trees

No



Yes

Decision trees

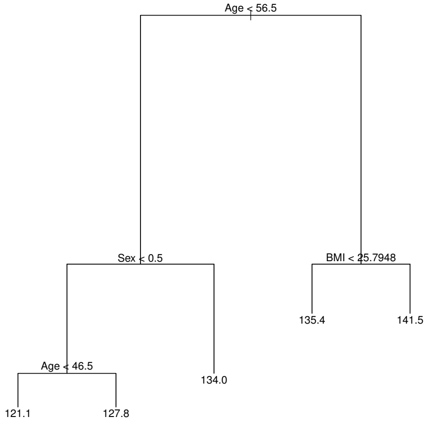
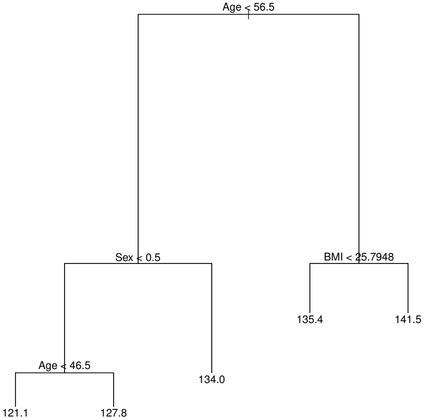


Figure 1: Regression tree on Sys. blood pressure

Decision trees



Exercise: Interpret this tree

Decision trees

Regression trees

- $Y \in \mathbb{R}$, continuous
- Aim: prediction
- Tree nodes - "leaves" - are discrete
  $\longrightarrow$ Need to discretise data space
- $\{R_1, R_2, .., R_J\}$: partition of X, $J < n$
  - For all $\{X_i\} \in R_j$, same prediction $\hat{Y}_j$
  - Group most similar data points together

$$\{R_1, R_2, ..., R_J\} = \arg\min \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_j)^2$$

Decision trees

Regression trees

$$\{R_1, R_2, ..., R_J\} = \arg\min \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_j)^2$$

- Computationally untractable
- Solution: **recursive binary splitting**
  - Recursively cut X space set in 2

Decision trees

Recursive binary splitting:

1. Split space in 2:

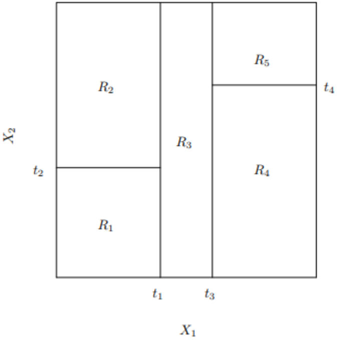
$$R_1(h,s) = \{X | X_h < s\},\ R_2(h,s) = \{X | X_h \geq s\}$$

2. Minimise RSS:

$$(h^*, s^*) = \arg\min \sum_{i \in R_1(h,s)} (y_i - \hat{y}_1)^2 + \sum_{i \in R_2(h,s)} (y_i - \hat{y}_2)^2$$

3. Repeat 1) and 2) within R_1
4. Stop when too few observations left

Decision trees



Draw this tree

Decision trees

What to do with a tree too big ?

- Overfitting

Decision trees

What to do with a tree too big ?

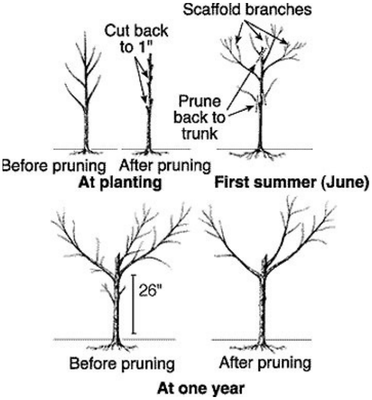
- Overfitting

Solution: Pruning

Select optimal sub-tree:

$$\min \sum_{j=1}^{|T|} \sum_{i \in R_j} (y_i - \hat{y}_j)^2 + \alpha |T|$$

$|T|$: # leaves in resulting tree



Decision trees

Classification trees

- $Y \in \{0, 1\}$ **or categorical** $\in \{1, 2, ..., K\}$
- Aim: prediction
- $\{R_1, R_2, .., R_J\}$: partition of X, $J < n$
  - For all $\{X_i\} \in R_j$, same **category** $\bar{Y}_j = k$
  - Group data points with same categories together

$$p_{mk} = \mathbb{P}\left((X_i, Y_i) \in R_m, \bar{Y}_m = k\right),$$

$$\min G = \sum_k p_{mk}(1 - p_{mk}) \text{ or } \min H = -\sum_k p_{mk} \log(p_{mk})$$

Decision trees

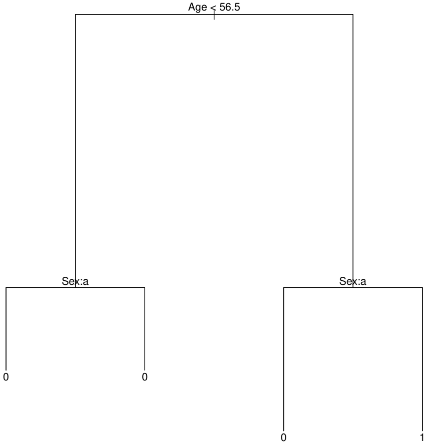


Figure 2: Classification tree on CVD risk

Decision trees - Limitations

- $+$ Interpretability
- $+$ Graphical representation
- $+$ Qualitative / categorical data

- - Poor prediction ability
- - Highly non-robust
  - slightly diff. tree parameters on the same data $=$ completely diff. results
  - high variance

Decision trees - Limitations

- + Interpretability
- + Graphical representation
- + Qualitative / categorical data

- - Poor prediction ability
- - Highly non-robust
  - slightly diff. tree parameters on the same data = completely diff. results
  - high variance

Solution: Aggregate several trees

Tree Bagging

Combine several trees together

1. Create B bootstrap samples (X^b, Y^b), $b \leq B$
2. Fit decision trees $\hat{f}^b$
   - prediction / classification: $\hat{Y}^b = \hat{f}^b(X)$
   - NO pruning
3. Average predictions over B samples:

$$\hat{f}_{bag}(X) = \frac{1}{B} \sum_b \hat{f}^b(X)$$

4. New "tree": $\hat{Y}_{bag} = \hat{f}_{bag}(X)$

Boosting

Tree boosting: recursive shrinkage for decision trees.
For $k \leq K$, do:

1. Fit decision trees $\hat{f}^k$ with d leaves only to $(X_i, \epsilon_{i,k})$:

$$\hat{f}^{k+1}(X_i) = \hat{f}^{k-1}(X_i) + \lambda \hat{f}^k(X_i)$$

2. Update errors

$$\epsilon_{i,k} = Y_i - \hat{f}^k(X_i)$$

Shrinkage parameter λ: learning rate
d: max. depth of trees, fixed

Random Forests

Bagging of de-correlated trees

1. Create B bootstrap samples (X^b, Y^b), $b \leq B$
2. Fit decision trees $\hat{f}^b$
   - at every split $j \leq J$, optimise over $m << p$ random predictors only:

$$\Omega(m) = \{q | q \in [1, p]\}, \ |\Omega(m)| = m < p$$

$$\min_{h \in \Omega(m)} \sum_{i \in R_j(h)} (y_i - \hat{y}_j)^2$$

   - Often $m \simeq \sqrt{p}$
3. Average predictions:

$$\hat{f}_{bag}(X) = \frac{1}{B} \sum_b \hat{f}^b(X)$$

Random Forests vs Bagging

Bagging

Random forest

Variable importance

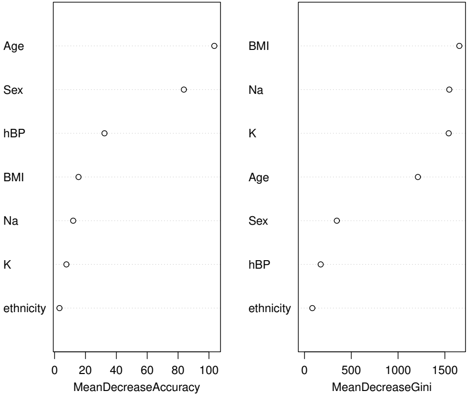
- Measure how much each variable improves trees' prediction across forest
- Over all nodes $j \leq J_b$ of all trees $\hat{f}_1, \hat{f}_2, ...\hat{f}_B$, record decrease in RSS / G or H index over every predictor $h \leq p$:

$$i(j) = RSS(j) - \frac{n_{j+1,1}}{n} RSS(j+1,1) - \frac{n_{j+1,2}}{n} RSS(j+1,2)$$

$$Imp(X_h) = \frac{1}{B} \sum_b \sum_{j \leq J_b, X_h \text{ used}} \frac{n_j}{n} i(j)$$

- "Important" predictors X_h for Y: high $Imp(X_h)$ values

Variable Importance



OOB error estimation

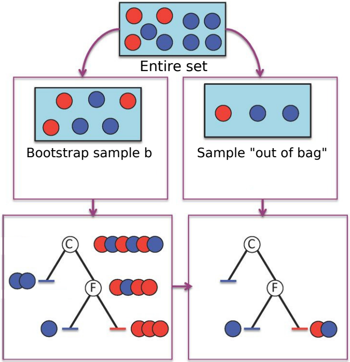
Similar to cross-validation, in bagging / random forests

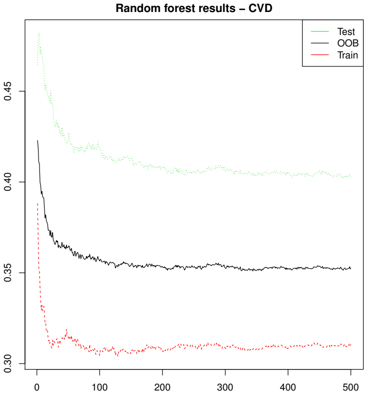
- B bootstrap samples, with decision tree $\hat{f}_b$
  - No tree $\hat{f}_b$ uses every observation (X_i, Y_i), $i \leq n$
  - $OOB_b = \{i \leq n | i \notin b\}$
- Use OOB_b samples as validation sets:

$$\epsilon(OOB) = \frac{1}{B} \sum_b \sum_{i, i \in OOB_b} RSS(Y_i, \hat{f}_b(X_i))$$

- $\epsilon(OOB)$: out of bag error

OOB error estimation





Important takeaways

- Decision trees
  - Regression & classification
  - Binary splitting
  - Pruning
- Bagging & Random forests
  - Bootstrapping
  - Boosting
  - Variable importance
  - OOB Error estimate