EECS E6690: Statistical Learning for Biological and Information Systems
Lecture 6: Tree-Based Methods

Prof. Predrag R. Jelenković

Time: Tuesday 4:10-6:40pm
303 Seeley W. Mudd Building

Dept. of Electrical Engineering
Columbia University , NY 10027, USA
Office: 812 Schapiro Research Bldg.
Phone: (212) 854-8174
Email: predrag@ee.columbia.edu
URL: http://www.ee.columbia.edu/∼predrag

Last lecture: Classification

- Regression: quantitative response
- Classification: categorical response
- Probability that a data point belongs to a class $c \in C$
- Example: Medical diagnosis
  - cancer, stroke, drug overdose, epileptic seizure
  - unordered set

Last lecture: General Bayes approach

The Optimal Bayes Classifier

- Assign x to a class k for which

  $$\mathbb{P}[Y = k \,|\, X = x]$$

  has the **maximum value**
- Bayes formula (assume X is discrete; otherwise, replace $X = x$ with $X \in (x, x + dx)$)

  $$p_k(x) = \mathbb{P}[Y = k \,|\, X = x] = \frac{\mathbb{P}[Y = k, X = x]}{\mathbb{P}[X = x]}$$

  $$= \frac{\mathbb{P}[Y = k]\mathbb{P}[X = x|Y = k]}{\sum_{l=1}^{K} \mathbb{P}[Y = l]\mathbb{P}[X = x|Y = l]} =: \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

  where

  - $\pi_k = \mathbb{P}[Y = k]$ - **prior** probability for class k
  - $p_k(x) = \mathbb{P}[Y = k \,|\, X = x]$ - **posterior** probability
  - $f_k(x) = \mathbb{P}[X = x \,|\, Y = k]$ - likelihood function: the density of X in class k
- Problem: $\mathbb{P}[Y = k \,|\, X = x], \mathbb{P}[X = x \,|\, Y = k]$ unknown
  - Make assumptions: logistic, Gaussian, independence, etc.

Proof of optimality of the Bayes classifier

- We consider the case of two classes $Y \in \{0, 1\}$ and general X, say $X \in \mathbb{R}^p$
- The proof is not needed for the grade.

Definition (Bayes classifier)

Let $\eta(x) = \mathbb{P}[Y = k \,|\, X = x]$ and

$$f^*(x) = \begin{cases} 1, & \text{if } \eta(x) \geq 1/2 \\ 0, & \text{otherwise,} \end{cases}$$

i.e., it assigns x to a class k for which $\mathbb{P}[Y = k \,|\, X = x]$ has maximum value.

Theorem (Optimality)

For any classifier $g(x) \in \{0, 1\}$,

$$\mathbb{P}[g(X) \neq Y] \geq \mathbb{P}[f^*(X) \neq Y],$$

i.e., the Bayes classifier is optimal.

Proof of optimality of the Bayes Classifier

Proof. We will actually prove a stronger statement that

$$\mathbb{P}[g(X) \neq Y | X = x] \geq \mathbb{P}[f^*(X) \neq Y | X = x],$$

which by taking the expectation with respect to X yields the theorem.

$$\begin{aligned}
&\mathbb{P}[g(X) \neq Y | X = x] = 1 - \mathbb{P}[g(X) = Y | X = x] \\
&= 1 - (\mathbb{P}[Y = 1, g(X) = 1 | X = x] + \mathbb{P}[Y = 0, g(X) = 0 | X = x]) \\
&= 1 - \left(\mathbb{E}[1_{\{Y=1\}} 1_{\{g(X)=1\}} | X = x] + \mathbb{E}[1_{\{Y=0\}} 1_{\{g(X)=0\}} | X = x]\right) \\
&= 1 - \left(1_{\{g(x)=1\}} \mathbb{E}[1_{\{Y=1\}} | X = x] + 1_{\{g(x)=0\}} \mathbb{E}[1_{\{Y=0\}} | X = x]\right) \\
&= 1 - \left(1_{\{g(x)=1\}} \mathbb{P}[Y = 1 | X = x] + 1_{\{g(x)=0\}} \mathbb{P}[Y = 0 | X = x]\right) \\
&= 1 - \left(1_{\{g(x)=1\}} \eta(x) + 1_{\{g(x)=0\}} (1 - \eta(x))\right)
\end{aligned}$$

Similarly, we can express

$$\mathbb{P}[f^*(X) \neq Y | X = x] = 1 - \left(1_{\{f^*(x)=1\}} \eta(x) + 1_{\{f^*(x)=0\}} (1 - \eta(x))\right)$$

Proof of optimality of the Bayes Classifier

Next, consider the difference

$$\begin{aligned}
&\mathbb{P}[g(X) \neq Y | X = x] - \mathbb{P}[f^*(X) \neq Y | X = x] \\
&= \eta(x) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right) + (1 - \eta(x)) \left(1_{\{f^*(x)=0\}} - 1_{\{g(x)=0\}}\right) \\
&= (2\eta(x) - 1) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right),
\end{aligned}$$

where the last equality uses
$1_{\{g(x)=0\}} = 1 - 1_{\{g(x)=1\}}, 1_{\{f^*(x)=0\}} = 1 - 1_{\{f^*(x)=1\}}$.
Finally, we show that the last expression is nonnegative. To this end, consider the following two cases:

1. $f^*(x) = 1 \Leftrightarrow \eta(x) \geq 1/2$, and therefore

   $$(2\eta(x)-1) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right) = (2\eta(x)-1) \left(1 - 1_{\{g(x)=1\}}\right) \geq 0$$

2. $f^*(x) = 0 \Leftrightarrow \eta(x) < 1/2$, which also imples

   $$(2\eta(x)-1) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right) = (2\eta(x)-1) \left(0 - 1_{\{g(x)=1\}}\right) \geq 0$$

□

Last lecture: Logistic regression

- Model $p_k(X), k = 0, 1,$ as logistic function
- Example:

$$\mathbb{P}[\text{default=Yes} \mid \text{balance}] = p_1(\text{balance})$$

- Logistic function (for binary variables, can be extended)

$$\mathbb{P}[\text{default=Yes} \mid X] \equiv p_1(X) \equiv p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

- Estimate $\boldsymbol{\beta}$ via Maximum Likelihood Estimation (MLE)
- Odds:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$$

- A unit increase in X multiplies odds by e^{β_1}
- $\operatorname{logit} p(X) = \beta_0 + \beta_1 X$

Example

```
> glm1a<-glm(default~balance,data = Default,family = binomial())
> summary(glm1a)

Call:
glm(formula = default ~ balance, family = binomial(), data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
        (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1596.5  on 9998  degrees of freedom
AIC: 1600.5

Number of Fisher Scoring iterations: 8
```

- Example:

$$\hat{\mathbb{P}}[\text{default} = \text{Yes} \,|\, \text{balance} = 1000] = \frac{e^{-10.65+0.0055\cdot 1000}}{1+e^{-10.65+0.0055\cdot 1000}} = 0.006$$

Last lecture: MLE fit

- Assume that $\mathbb{P}[Y_i = y_i \,|\, X_i = x_i]$ follows the logistic function $p(x)$, and the conditional independence

$$\mathbb{P}[Y_1 = y_1, \dots, Y_n = y_n \,|\, X_1 = x_1, \dots, X_n = x_n]$$
$$= \mathbb{P}[Y_1 = y_1 \,|\, X_1 = x_1] \cdots \mathbb{P}[Y_n = y_n \,|\, X_n = x_n]$$

- Hence the preceding conditional probability is maximized on observed data for $\boldsymbol{\beta}$ that maximizes the likelihood function

$$\ell(\boldsymbol{\beta}) = \prod_{i\colon y_i=1} p(x_i) \prod_{i\colon y_i=0} (1 - p(x_i))$$

- MLE: select a model that maximizes likelihood of data

$$\max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})$$

or equivalently

$$\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ y_i(\beta_0 + \beta_1 x_i) - \ln\left(1 + e^{\beta_0 + \beta_1 x_i}\right) \right\}$$

- First-order conditions: Newton's method

Last lecture: Multiple logistic regression

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}$$

```
> glm2<-glm(default~balance+income+student,data = Default,family = binomial())
> summary(glm2)

Call:
glm(formula = default ~ balance + income + student, family = binomial(),
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4691  -0.1418  -0.0557  -0.0203   3.7383

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.5  on 9996  degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```

Last lecture: Discriminant classification

Gaussian assumptions

- Start with $p = 1$
- Gaussian density (mean μ_k, variance σ_k^2):

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

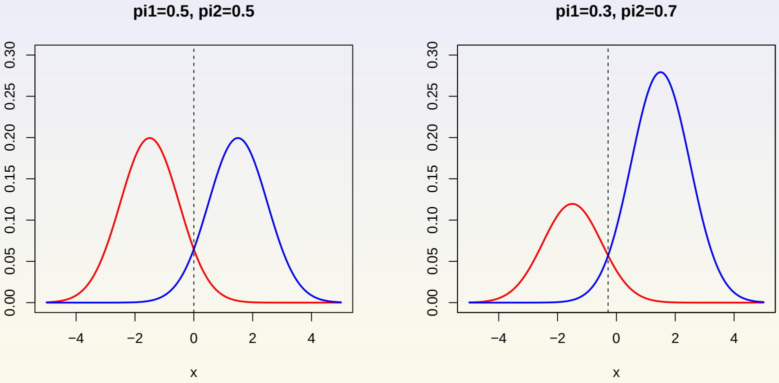
- Discriminant function δ_k is quadratic (in x):

$$p_k(x) \propto \delta_k(x) = -x^2 \frac{1}{2\sigma_k^2} + x\frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} - \log \sigma_k + \log \pi_k$$

- Probabilities:

$$\mathbb{P}[Y = k \mid X = x] = \frac{e^{\delta_k(x)}}{\sum_{l=1}^{K} e^{\delta_l(x)}}$$

Example



Last lecture: Linear discriminant analysis

- **Special case**: $\sigma_1 = \sigma_2 = \ldots = \sigma_K = \sigma$
- Discriminant function δ_k is linear (in x):

$$p_k(x) \propto \delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

- Example: $K = 2$, $\pi_1 = \pi_2$ – decision boundary is at

$$x = \frac{\mu_1 + \mu_2}{2}$$

- Parameter estimation:

$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:\, y_i = k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i:\, y_i = k} (x_i - \hat{\mu}_k)^2$$

Last lecture: Quadratic discriminant analysis

Quadratic - sigma unequal

- Density:

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}$$

- Linear discriminant function (equal $\boldsymbol{\Sigma}_k$):

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

- Quadratic discriminant function (different $\boldsymbol{\Sigma}_k$):

$$\delta_k(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log \pi_k$$

Last lecture: Logistic regression vs. LDA

- Two classes
- Logistic regression

$$\log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

- LDA

$$\log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} = \left(\log \frac{\pi_1}{\pi_2} - \frac{1}{2} \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 \right) + \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 \right)$$

- Same linear form
- Different way to estimate parameters

Last lecture: Naïve Bayes

- Assumes features are independent in each class
- Covariance matrices $\boldsymbol{\Sigma}_k$ are diagonal:

$$\pi_k f_k(\boldsymbol{x}) = \pi_k \prod_{i=1}^{p} f_{ki}(x_i) = \pi_k \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_{ki}} e^{-\frac{(x_i-\mu_{ki})^2}{2\sigma_{ki}^2}}$$

and

$$\delta_k(\boldsymbol{x}) = -\sum_{i=1}^{p} \left[\frac{(x_i-\mu_{ki})^2}{2\sigma_{ki}^2} + \log \sigma_{ki} \right] + \log \pi_k$$

- Advantages
  - much easier to estimate parameters for $p \gg 1$
  - can use both qualitative and categorical features (use PMFs instead of PDFs)
  - often produces good results

Predictability versus Interpretability

- Predictability: Determined by how good is the model in predicting the future values, i.e., minimizing the prediction error.
- Interpretability: Determined by how well the model interprets/explains data.

Often, these important questions don't go hand in hand

- LDA is easier to interpret than the logistic classification.
- Today, we'll see tree based methods that have good interpretability for both regression and classification.

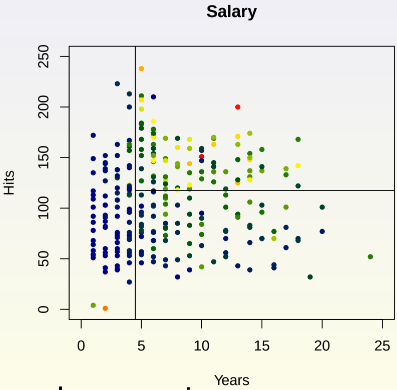
Tree-Based Methods: Decision trees

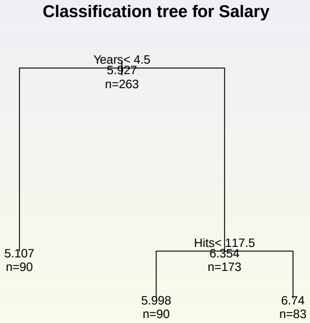
- Models for regression and classification

- Idea:
  - Segment the predictor space $(X_1, \ldots, X_p)$ into distinct and non-overlapping regions, $R_1, \ldots, R_j$
  - Prediction (classification) based on:
    - Average (majority vote) over segments

Example: Hitters

- Predict `Salary` based on `Years` and `Hits`
- Remove missing values and apply log-transform
- Salary encoding from low to high: blue - green - yellow - red





- Interpretation
- Prediction

Regression Trees: Segmentation

- In general, the regions can have any shape
- Focus on high-dimensional rectangles (boxes)

- Goal: Find $R_1, \ldots, R_J$ that minimize the RSS:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

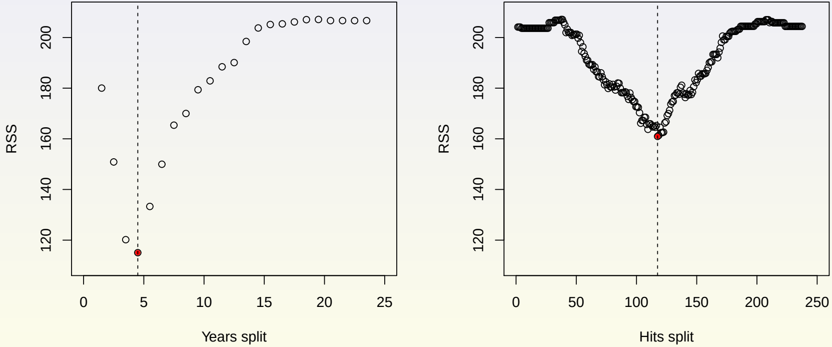
  where $\hat{y}_{R_j}$ is the mean response for the observations in R_j
- Computationally infeasible to consider all partitions

- Top-down, greedy approach: Binary splitting
- Stopping criteria (e.g., max number of observations in a box)

Segmentation: Example

- $R^-(j,s) = \{X : X_j \le s\}$ and $R^+(j,s) = \{X : X_j > s\}$
- Minimize

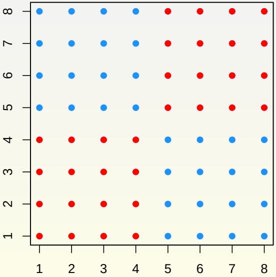
$$\sum_{i:\, x_i \in R^-(j,s)} (y_i - \hat{y}_{R^-})^2 + \sum_{i:\, x_i \in R^+(j,s)} (y_i - \hat{y}_{R^+})^2$$



Year-split gives the minimal RSS

Overfitting

- Optimal tree size?
  - training error decreases as the size increases
  - testing error decreases, but then increases
- Grow the tree only if RSS decreases – poor results
- Example: 2 vales: red and blue
  always same RSS regardless of the cut
  but for the next cut - there is (!)



- Alternative: Grow the tree to a large size and then trim it back

Tree pruning

- Start with a large tree T_0
- Cost complexity pruning (weakest link pruning)
- Sequence of trees indexed by α
- For each α:

$$\min_{T \subseteq T_0} \left\{ \sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \right\},$$

  where

  - $|T|$ is the number of leafs in T
  - R_m is the box corresponding to the mth leaf
  - $\hat{y}_{R_m}$ is the mean of training observations in R_m
- Parameter α
  - Controls the complexity/fit tradeoff
  - Select $\hat{\alpha}$ using cross-validation

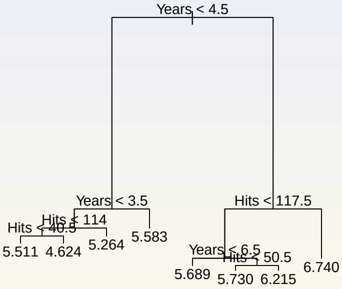
Fitting a tree

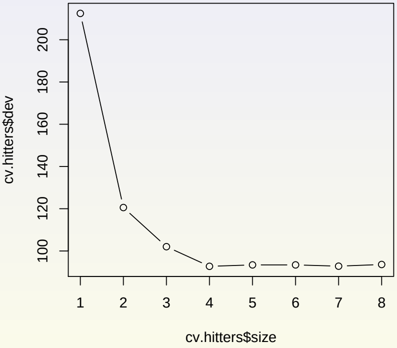
1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α.
3. Use K-fold cross-validation to choose α. For each $k = 1, \ldots, K$:
   3.1. Repeat Steps 1 and 2 on the $(K-1)$ fraction of the training data, excluding the kth fold
   3.2. Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α.
4. Return the subtree from Step 2 that corresponds to the chosen value of α.

Example: Hitters

```
> library(tree)
> hitters.fit<-tree(Salary~Years+Hits, data=myHitters)
> summary(hitters.fit)

Regression tree:
tree(formula = Salary ~ Years + Hits, data = myHitters)
Number of terminal nodes:  8
Residual mean deviance:  0.2708 = 69.06 / 255
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.2400 -0.2980 -0.0365  0.0000  0.3233  2.1520
> cv.hitters<-cv.tree(hitters.fit)
> cv.hitters
$size
[1] 8 7 6 5 4 3 2 1

$dev
[1]  95.23044  91.91239  95.49769  95.49769  90.07986  96.01860 117.07588 211.16929

$k
[1]      -Inf  2.293634  3.470318  3.501308  3.793540  9.210099 23.728527 92.095258

$method
[1] "deviance"

attr(,"class")
[1] "prune"         "tree.sequence"
```

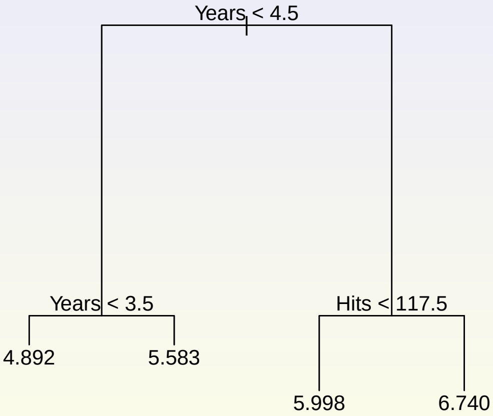
Example: Hitters





Example: Hitters

```
> prune.hitters<-prune.tree(hitters.fit,best=cv.hitters$size[which.min(cv.hitters$dev)])
```



Classification trees

- Similar to regression trees
- Predict a qualitative response
- Prediction within a box: most commonly occurring class
- Need an alternative to RSS

Objective

- $\hat{p}_{m,k}$ – proportion of training observations in the mth box that are from class k

- Minimize one of the following measures
  - Classification error rate

$$E = 1 - \max_k \hat{p}_{m,k}$$

  - Gini index

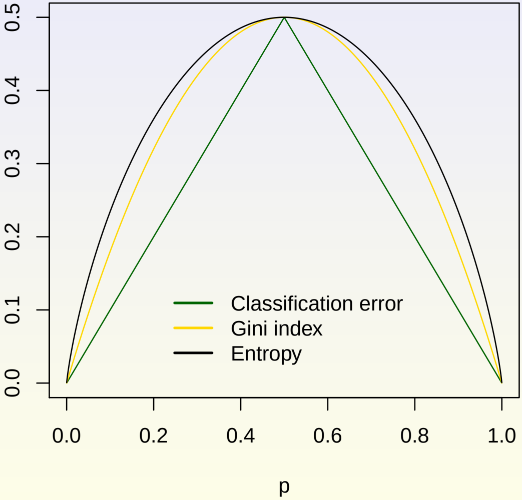
$$G = \sum_{k=1}^{K} \hat{p}_{m,k}(1 - \hat{p}_{m,k})$$

  - Entropy

$$D = -\sum_{k=1}^{K} \hat{p}_{m,k} \log \hat{p}_{m,k}$$

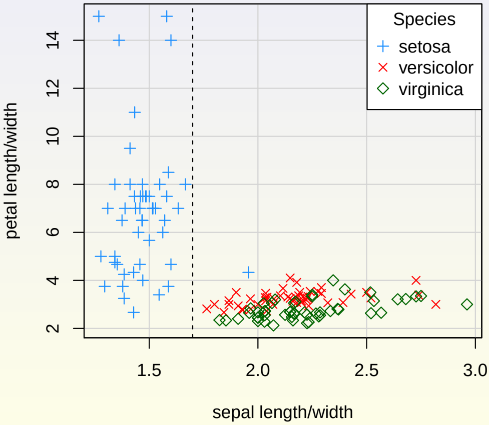
Measures

- $K = 2$


0.5
0.4
0.3
0.2
0.1
0.0
Classification error
Gini index
Entropy
0.0
0.2
0.4
0.6
0.8
1.0
p

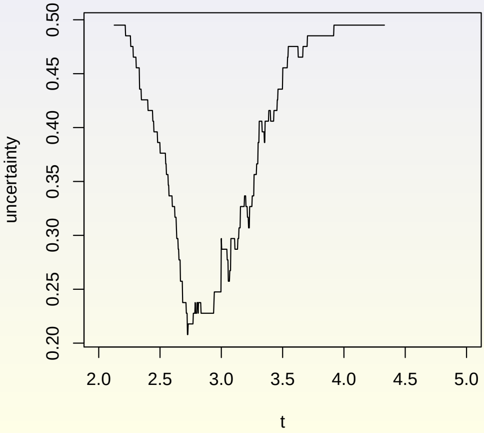
Example: Irises

- Classifying irises using sepal and petal measurements:
  - $x \in \mathbb{R}^2$, $y \in \{1, 2, 3\}$
  - x_1 = ratio of sepal length to width
  - x_2 = ratio of petal length to width



Example: Irises

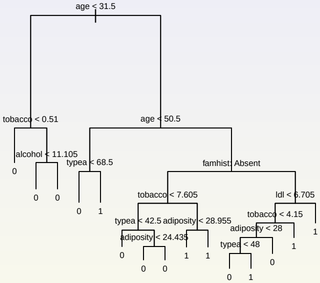
- Split R_2 using $1_{\{x_2 \leq t\}}$
  - $u(R_2^-)$
  - $u(R_2^+)$
  - $p_{R_2^-} u(R_2^-) + p_{R_2^+} u(R_2^+)$

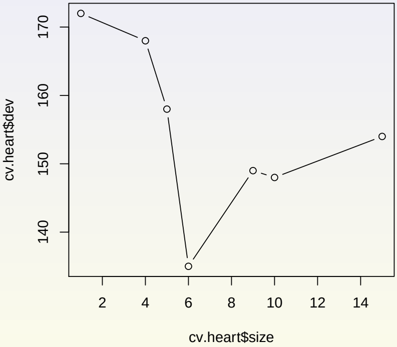


Example: South African heart disease set

```
> heart.tree<-tree(chd~.,data=SAheart)
> summary(heart.tree)

Classification tree:
tree(formula = chd ~ ., data = SAheart)
Variables actually used in tree construction:
[1] "age"       "tobacco"   "alcohol"   "typea"     "famhist"   "adiposity" "ldl"
Number of terminal nodes:  15
Residual mean deviance:  0.8733 = 390.3 / 447
Misclassification error rate: 0.2078 = 96 / 462

> set.seed(1)
> cv.heart<-cv.tree(heart.tree,FUN=prune.misclass)
> cv.heart
$size
[1] 15 10  9  6  5  4  1

$dev
[1] 154 148 149 135 158 168 172

$k
[1] -Inf    0    1    3    8   10   12

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

Example: South African heart disease set

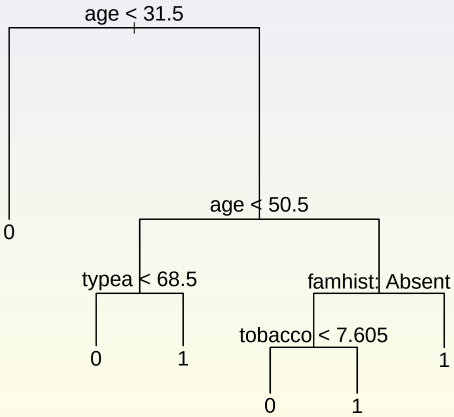




Example: South African heart disease set

```
> heart.prune<-prune.misclass(heart.tree,best=cv.heart$size[which.min(cv.heart$dev)])
> heart.predict<-predict(heart.prune,data=SAheart,type="class")
> table(heart.predict,SAheart$chd)
```

```
heart.predict   0   1
            0 266  70
            1  36  90
```



Recall Bootstrap: Basic algorithm

- Input
  - A sample of data $\boldsymbol{Z} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n)$
  - An estimation rule $\hat{T}$ for Statistic T

- Algorithm
  1. Generate bootstrap samples $\boldsymbol{Z}^{*1}, \boldsymbol{Z}^{*2}, \ldots, \boldsymbol{Z}^{*B}$
     - Create $\boldsymbol{Z}^{*b}$ by selecting points from $\boldsymbol{Z}$
     - A particular $\boldsymbol{Z}_i$ can appear in $\boldsymbol{Z}^{*b}$ multiple times
  2. Evaluate the estimator on each $\boldsymbol{Z}^{*b}$:

$$\hat{T}_b = \hat{T}(\boldsymbol{Z}^{*b})$$

- The empirical distribution of $\{\hat{T}_1, \ldots, \hat{T}_B\}$ is an estimate of the distribution of $T(\boldsymbol{Z})$
- Bootstrap distribution
- Overlap between $\boldsymbol{Z}$ and $\boldsymbol{Z}^{*b}$?

Bumping

Works for both: classifiers or regressions

- Stochastic search

  avoids getting stuck in a poor solution/local minimum

- Train a classifier or regression model $\hat{f}_0$ on $\boldsymbol{Z}$
- For $b = 1, \ldots, B$:
  1. Draw a bootstrap sample $\boldsymbol{Z}^{*b}$ of size n from training data
  2. Train a classifier or regression model $\hat{f}_b$ on $\boldsymbol{Z}^{*b}$
- Select the best model, e.g.,

$$\hat{b} = \arg \min_{0 \leq b \leq B} \sum_{i=1}^{n} \left(y_i - \hat{f}_b(\boldsymbol{z}_i) \right)^2$$

Bagging

Works for both: classifiers or regressions

- Bootstrap aggregation/averaging

  reduces the variance/overfitting

- For $b = 1, \ldots, B$:
  1. Draw a bootstrap sample $\boldsymbol{Z}^{*b}$ of size n from training data
  2. Train a classifier or regression model $\hat{f}_b$ on $\boldsymbol{Z}^{*b}$

- For a “new” point $\boldsymbol{x}_0$, compute:

$$\hat{f}_{\mathsf{avg}}(\boldsymbol{x}_0) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\boldsymbol{x}_0)$$

  - Regression: $\hat{f}_{\mathsf{avg}}(\boldsymbol{x}_0)$ is the prediction
  - Classification: Pick majority

- Example: Bagging trees

Random Forests

Works for both: classifiers or regressions

- Improvement over bagged trees
- Idea: Decorrelated trees
  - Still learn a tree on each bootstrap set
  - To split a region, consider only a subset of predictors/covariates
- Input parameter: $m \leq p$, often $m \approx \sqrt{p}$
- For $b = 1, \ldots, B$
  - Draw a bootstrap sample $\boldsymbol{Z}^{*b}$ of size n from the training data
  - Train a tree classifier on $\boldsymbol{Z}^{*b}$, each split is computed as:
    - Randomly select m predictors/covariates, newly chosen for each b
    - Make the best split restricted to that subsets of covariates
- Similarly as in bagging: for regression prediction

$$\hat{f}_{\text{avg}}(\boldsymbol{x}_0) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\boldsymbol{x}_0);$$

for classification: choose the majority vote among B classifiers.

Example: South African heart disease set

```
> library(randomForest)
> set.seed(10)
> train<-sample(1:nrow(SAheart),nrow(SAheart)/2)
> bag.heart<-randomForest(chd˜., data = SAheart, subset=train, mtry=9, importance=TRUE)
> bag.heart

Call:
 randomForest(formula = chd ˜ ., data = SAheart, mtry = 9, importance = TRUE,      subset = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 9

        OOB estimate of  error rate: 36.8%
Confusion matrix:
    0  1 class.error
0 123 31   0.2012987
1  54 23   0.7012987
> table(SAheart$chd[-train],predict(bag.heart, newdata = SAheart[-train,]))

      0   1
  0 118  30
  1  54  29
>
> bag.heart<-randomForest(chd˜., data = SAheart, subset=train, mtry=3, importance=TRUE)
> bag.heart

Call:
 randomForest(formula = chd ˜ ., data = SAheart, mtry = 3, importance = TRUE,      subset = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 33.33%
Confusion matrix:
    0  1 class.error
0 134 20   0.1298701
1  57 20   0.7402597
> table(SAheart$chd[-train],predict(bag.heart, newdata = SAheart[-train,]))

      0   1
  0 128  20
  1  58  25
```

Reading:

ISL: Read Chapter 8

ESL: Section 9.2

Homework: Homework 3 due Wed, Oct 19.
No late submission allowed in order to give you enough time to study the solutions before the midterm, which is planned for Oc 25.