Pattern recognition on spatial data

Predicting structure : classification methods

Pr Yves Brostaux, ULiège

Openspat ULisboa

June 2019

Contents

1 Introduction

2 Linear predictive discriminant analysis

3 Nearest neighbours methods

4 Decision trees

5 Random Forest

6 Spatial smoothing

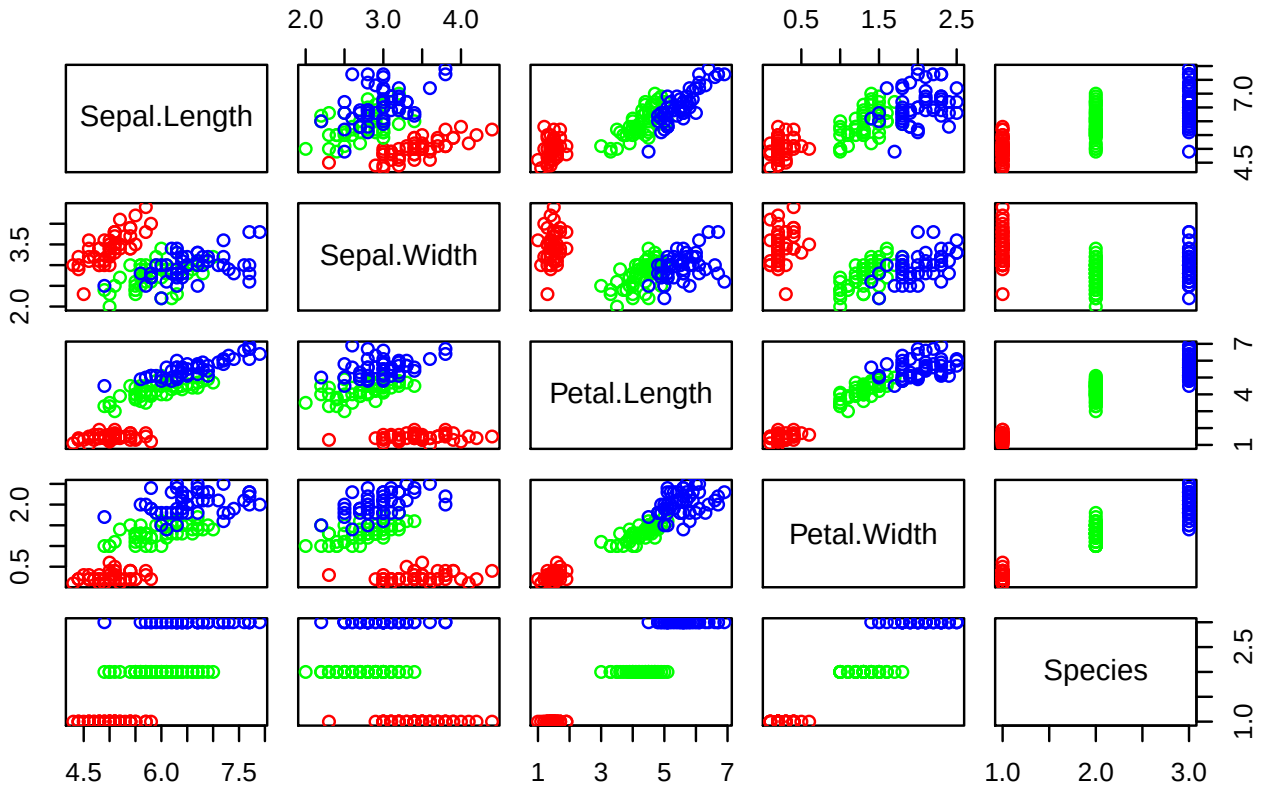
Introduction

- allocation of n observations into k pre existing groups

Example data

Iris data (FISHER, 1936)

- 3 species (50 obs/sp)
  - I. setosa
  - I. versicolor
  - I. virginica
- 4 variables
  - sepal length
  - sepal width
  - petal length
  - petal width


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
2.0
3.0
4.0
0.5
1.5
2.5
4.5
7.0
2.0
3.5
1
4
7
0.5
2.0
1.0
2.5
4.5
6.0
7.5
1
3
5
7
1.0
2.0
3.0

Contents

1 Introduction

2 Linear predictive discriminant analysis
- Two populations/one variable
- Two populations/two variables
- g populations/p variables
- Error rates

3 Nearest neighbours methods

4 Decision trees

5 Random Forest

Linear predictive discriminant analysis

- Two populations/one variable
- Two populations/two variables
- g populations/p variables
- Error rates

Two populations/one variable

Example

- I. versicolor and I. virginica
- petal length

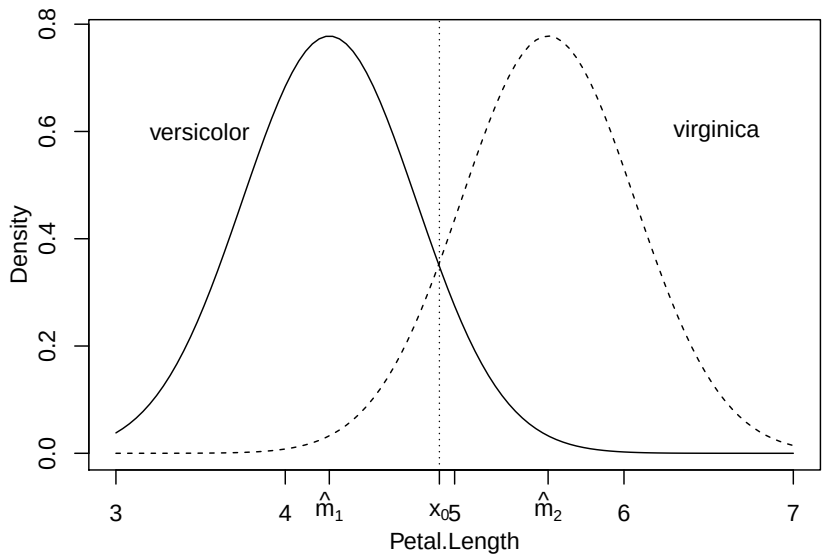
Species	I. versicolor	I. virginica
mean	4.260	5.552
standard-deviation	0.470	0.552

Assumptions

- normality
- homogeneity of within group variance

Two populations/one variable

$$\sigma^2 = \frac{(49)(0.470)^2 + (49)(0.552)^2}{49 + 49} = 0.263$$



Classification rules

- **Threshold**

$$x_0 = (\hat{m}_1 + \hat{m}_2)/2$$

- **Distance**

$$d_{1i}^2 = \left(\frac{x_i - \hat{m}_1}{\hat{\sigma}}\right)^2 \quad \text{et} \quad d_{2i}^2 = \left(\frac{x_i - \hat{m}_2}{\hat{\sigma}}\right)^2$$

- **Density**

$$f_1(x_i) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x_i - \hat{m}_1}{\hat{\sigma}}\right)^2\right]$$

$$f_2(x_i) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x_i - \hat{m}_2}{\hat{\sigma}}\right)^2\right]$$

Classification rules

- **Posterior probability**

$$P(A1 \mid x_i) = \frac{f_1(x_i)}{f_1(x_i) + f_2(x_i)} = \frac{\exp\left(-\frac{1}{2}d_{1i}^2\right)}{\sum_{k=1}^{2}\exp\left(-\frac{1}{2}d_{ki}^2\right)}$$

$$P(A2 \mid x_i) = \frac{f_2(x_i)}{f_1(x_i) + f_2(x_i)} = \frac{\exp\left(-\frac{1}{2}d_{2i}^2\right)}{\sum_{k=1}^{2}\exp\left(-\frac{1}{2}d_{ki}^2\right)}$$

Classification rules

Assign unit i to population 1 if :

- $x_i < x_0$
- $d_{1i}^2 < d_{2i}^2$
- $f_1(x_i) > f_2(x_i)$
- $\mathrm{P}(\mathrm{A1} \mid x_i) > \mathrm{P}(\mathrm{A2} \mid x_i)$

Example

$x_i = 4.7$

- $x_i = 4.7 < x_0 = 4.91$
- $d_{1i}^2 = 0.74 < d_{2i}^2 = 2.76$
- $f_1(x_i) = 0.54 > f_2(x_i) = 0.20$
- $$\mathrm{P}(\mathrm{A1} \mid x_i) = \frac{0.54}{0.54 + 0.20} = 0.73$$
  $$\mathrm{P}(\mathrm{A2} \mid x_i) = \frac{0.20}{0.54 + 0.20} = 0.27$$

$\Rightarrow$ x_i is allocated to population 1

Linear predictive discriminant analysis

- Two populations/one variable
- Two populations/two variables
- g populations/p variables
- Error rates

Two populations/two variables

Example

- I. versicolor and I. virginica
- petal length and petal width

Assumptions

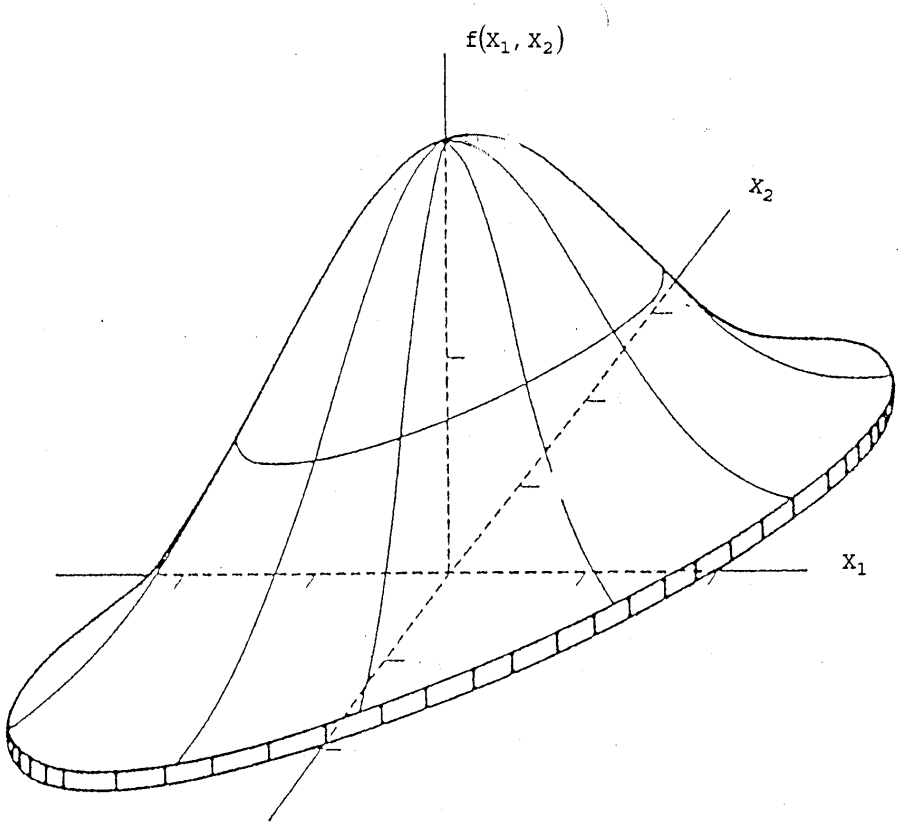
- normality
- homogeneity of within group covariance matrix

Two populations/two variables

	I. versicolor	I. virginica
mean length	4.260	5.552
mean width	1.326	2.026
covariance matrix	$\begin{bmatrix} 0.2208 & 0.0731 \\ 0.0731 & 0.0391 \end{bmatrix}$	$\begin{bmatrix} 0.3046 & 0.0488 \\ 0.0488 & 0.0754 \end{bmatrix}$

$$\hat{\Sigma} = \left(49\widehat{\Sigma}_1 + 49\widehat{\Sigma}_2\right)/98 = \begin{bmatrix} 0.2627 & 0.0610 \\ 0.0610 & 0.0573 \end{bmatrix}$$

Distribution normale à 2 dimensions



Classification rules

- **Density**

$$f_1(x_{1i}, x_{2i}) = \frac{1}{2\pi\hat{\sigma}_{x_1}\hat{\sigma}_{x_2}\sqrt{(1-\hat{\rho}^2)}} \exp\left[-\frac{1}{2}d_{1i}^2\right]$$

$$f_2(x_{1i}, x_{2i}) = \frac{1}{2\pi\hat{\sigma}_{x_1}\hat{\sigma}_{x_2}\sqrt{(1-\hat{\rho}^2)}} \exp\left[-\frac{1}{2}d_{2i}^2\right]$$

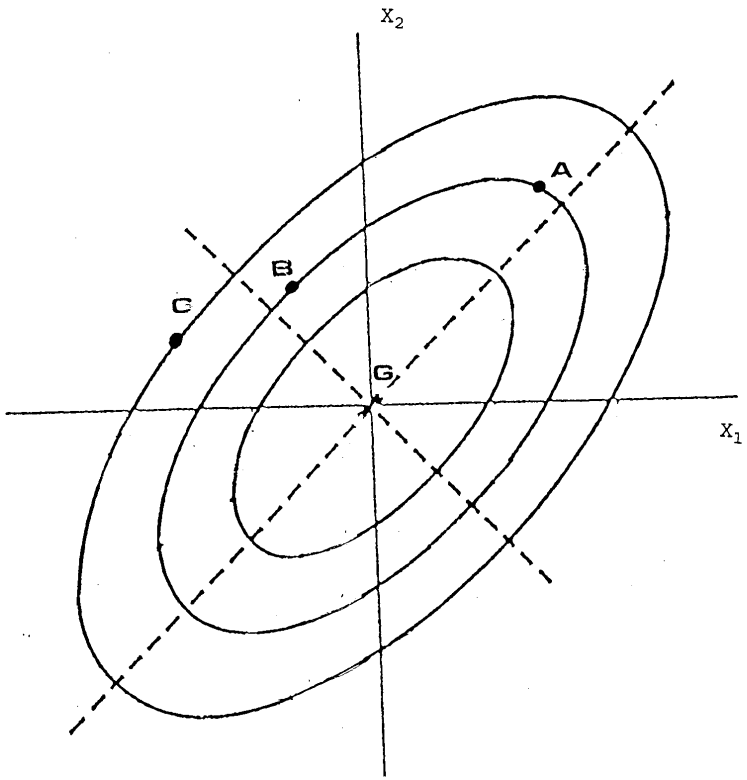
Classification rules

- **Distance (Mahalanobis)**

$$d_{1i}^2 = \frac{1}{1-\hat{\rho}^2}\left[\left(\frac{x_{1i}-\hat{m}_{11}}{\hat{\sigma}_{x_1}}\right)^2 - 2\hat{\rho}\left(\frac{x_{1i}-\hat{m}_{11}}{\hat{\sigma}_{x_1}}\right)\left(\frac{x_{2i}-\hat{m}_{12}}{\hat{\sigma}_{x_2}}\right) + \left(\frac{x_{2i}-\hat{m}_{12}}{\hat{\sigma}_{x_2}}\right)^2\right]$$

$$d_{2i}^2 = \frac{1}{1-\hat{\rho}^2}\left[\left(\frac{x_{1i}-\hat{m}_{21}}{\hat{\sigma}_{x_1}}\right)^2 - 2\hat{\rho}\left(\frac{x_{1i}-\hat{m}_{21}}{\hat{\sigma}_{x_1}}\right)\left(\frac{x_{2i}-\hat{m}_{22}}{\hat{\sigma}_{x_2}}\right) + \left(\frac{x_{2i}-\hat{m}_{22}}{\hat{\sigma}_{x_2}}\right)^2\right]$$

Mahalanobis' distance



Classification rules

- **Distance (Mahalanobis)**

$$d_{1i}^2 = [\mathbf{x}_i - \hat{\mathbf{m}}_1]' \hat{\mathbf{\Sigma}}^{-1} [\mathbf{x}_i - \hat{\mathbf{m}}_1]$$
$$d_{2i}^2 = [\mathbf{x}_i - \hat{\mathbf{m}}_2]' \hat{\mathbf{\Sigma}}^{-1} [\mathbf{x}_i - \hat{\mathbf{m}}_2]$$

$$\mathbf{x}_i = \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} \hat{\mathbf{m}}_1 = \begin{bmatrix} \hat{m}_{11} \\ \hat{m}_{12} \end{bmatrix} \hat{\mathbf{m}}_2 = \begin{bmatrix} \hat{m}_{21} \\ \hat{m}_{22} \end{bmatrix} \hat{\mathbf{\Sigma}} = \begin{bmatrix} \hat{\sigma}_{x_1}^2 & \hat{\mu}_{11} \\ \hat{\mu}_{11} & \hat{\sigma}_{x_2}^2 \end{bmatrix}$$

Classification rules

- **Posterior probability**

$$P(A1 \mid x_i) = \frac{f_1(x_i)}{f_1(x_i) + f_2(x_i)} = \frac{\exp\left(-\frac{1}{2} d_{1i}^2\right)}{\sum_{k=1}^{2} \exp\left(-\frac{1}{2} d_{ki}^2\right)}$$

$$P(A2 \mid x_i) = \frac{f_2(x_i)}{f_1(x_i) + f_2(x_i)} = \frac{\exp\left(-\frac{1}{2} d_{2i}^2\right)}{\sum_{k=1}^{2} \exp\left(-\frac{1}{2} d_{ki}^2\right)}$$

Classification rules

Assign unit i to population 1 if :

- $d_{1i}^2 < d_{2i}^2$
- $f_1(x_i) > f_2(x_i)$
- $\mathrm{P}(\mathrm{A1} \mid x_i) > \mathrm{P}(\mathrm{A2} \mid x_i)$

Example

$$x_i = \begin{bmatrix} 4.7 \\ 1.6 \end{bmatrix}$$

- $d^2_{1i} = 1.422 < d^2_{2i} = 3.972$
- $f_1(x_i) = 0.734 > f_2(x_i) = 0.205$
- $$P(A1 \mid x_i) = \frac{0.734}{0.734 + 0.205} = 0.782$$
  $$P(A2 \mid x_i) = \frac{0.205}{0.734 + 0.205} = 0.218$$

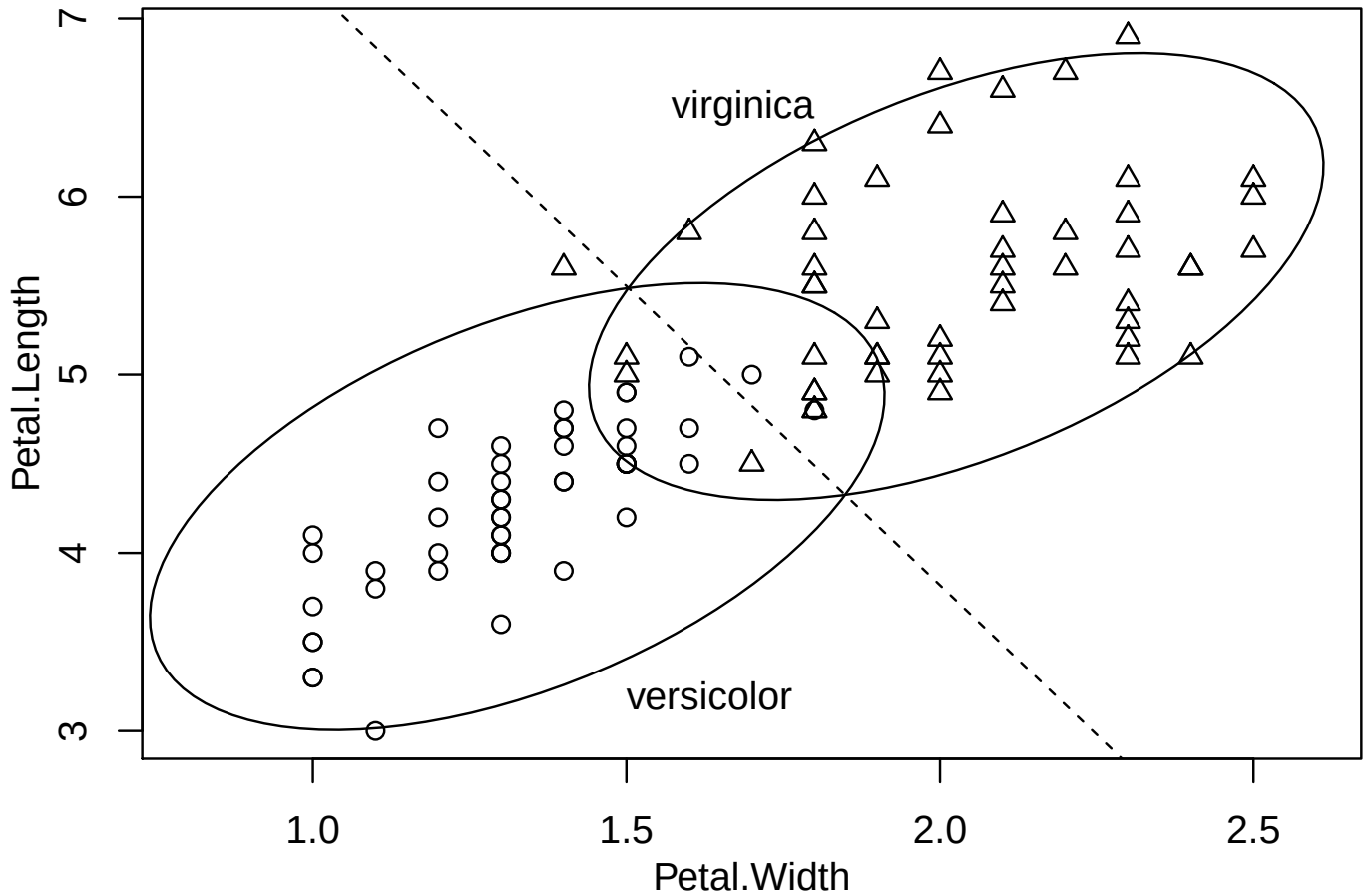
$\Rightarrow$ x_i is allocated to population 1

Geometric interpretation

The limit between the two populations is defined by the set of points which are at **equal (Mahalanobis') distance** from the centroïds of the populations.

This set of points draw a **straight line** between the populations, which pass through the **intersection of the ellipses** of equal Mahalanobis' distance, giving its name to the method (*linear* discriminant analysis).

Geometric interpretation



Linear predictive discriminant analysis

- Two populations/one variable
- Two populations/two variables
- g populations/p variables
- Error rates

g populations/p variables

Example

- I. versicolor, I. virginica, I. setosa
- petal length and petal width

Assumptions

- normality
- homogeneity of within group covariance matrix

Classification rules

- **Distance (for population h)**

$$d_{hi}^2 = [\mathbf{x}_i - \hat{\mathbf{m}}_h]' \, \hat{\mathbf{\Sigma}}^{-1} \, [\mathbf{x}_i - \hat{\mathbf{m}}_h]$$

- **Density (for population h)**

$$f_h(\mathbf{x}_i) = \frac{1}{\sqrt{(2\pi)^p |\hat{\mathbf{\Sigma}}|}} \exp\left[-\frac{1}{2} d_{hi}^2\right]$$

- **Posterior probability (for population h)**

$$\mathrm{P}(\mathrm{A}h \mid \mathbf{x}_i) = \frac{\exp\left(-\frac{1}{2} d_{hi}^2\right)}{\sum_{k=1}^{g} \exp\left(-\frac{1}{2} d_{ki}^2\right)}$$

Classification rules

- **Likelihood ratio**\

$$\frac{f_h(\mathbf{x})}{f_l(\mathbf{x})} = \frac{\left(1/\sqrt{(2\pi)^p|\hat{\mathbf{\Sigma}}|}\right)\exp\left[-\frac{1}{2}(x-\hat{m}_h)'\hat{\mathbf{\Sigma}}^{-1}(x-\hat{m}_h)\right]}{\left(1/\sqrt{(2\pi)^p|\hat{\mathbf{\Sigma}}|}\right)\exp\left[-\frac{1}{2}(x-\hat{m}_l)'\hat{\mathbf{\Sigma}}^{-1}(x-\hat{m}_l)\right]}$$

- **Log-likelihood ratio**\

$$\log_e(L_{hl} = \left(\hat{\mathbf{m}}_h\hat{\mathbf{\Sigma}}^{-1}\mathbf{x} - \frac{1}{2}\hat{\mathbf{m}}_h'\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{m}}_h\right) - \left(\hat{\mathbf{m}}_l\hat{\mathbf{\Sigma}}^{-1}\mathbf{x} - \frac{1}{2}\hat{\mathbf{m}}_l'\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{m}}_l\right)$$

Example with R

```
# load data (internal)
data(iris)

# select only petal length and width
iris4 <- subset(iris, select=3:5)

# load package
library(MASS)

# adjust the LDA
iris.lda <- lda(Species~., data=iris4)
```

Example with R

```
## Call:
## lda(Species ~ ., data = iris4)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Petal.Length Petal.Width
## setosa            1.462       0.246
## versicolor        4.260       1.326
## virginica         5.552       2.026
##
## Coefficients of linear discriminants:
##                    LD1       LD2
## Petal.Length 1.544371 -2.161222
## Petal.Width  2.402394  5.042599
##
## Proportion of trace:
##    LD1    LD2
```

Canonical discriminant analysis

Linear discriminant analysis can also be seen as a **factor analysis** (like PCA), which aims at creating linear combinations of the original variables that gives the best possible separation between the groups.

Canonical variables are then calculated by an similar procedure to PCA, but the criteria of maximum variance of the resulting components is replaced by the **maximum separation between the groups**.

$$F = \frac{\sigma^2_{Between}}{\sigma^2_{Within}}$$

Linear discriminant functions

```
# Linear discriminants coefficients
iris.lda$scaling
```

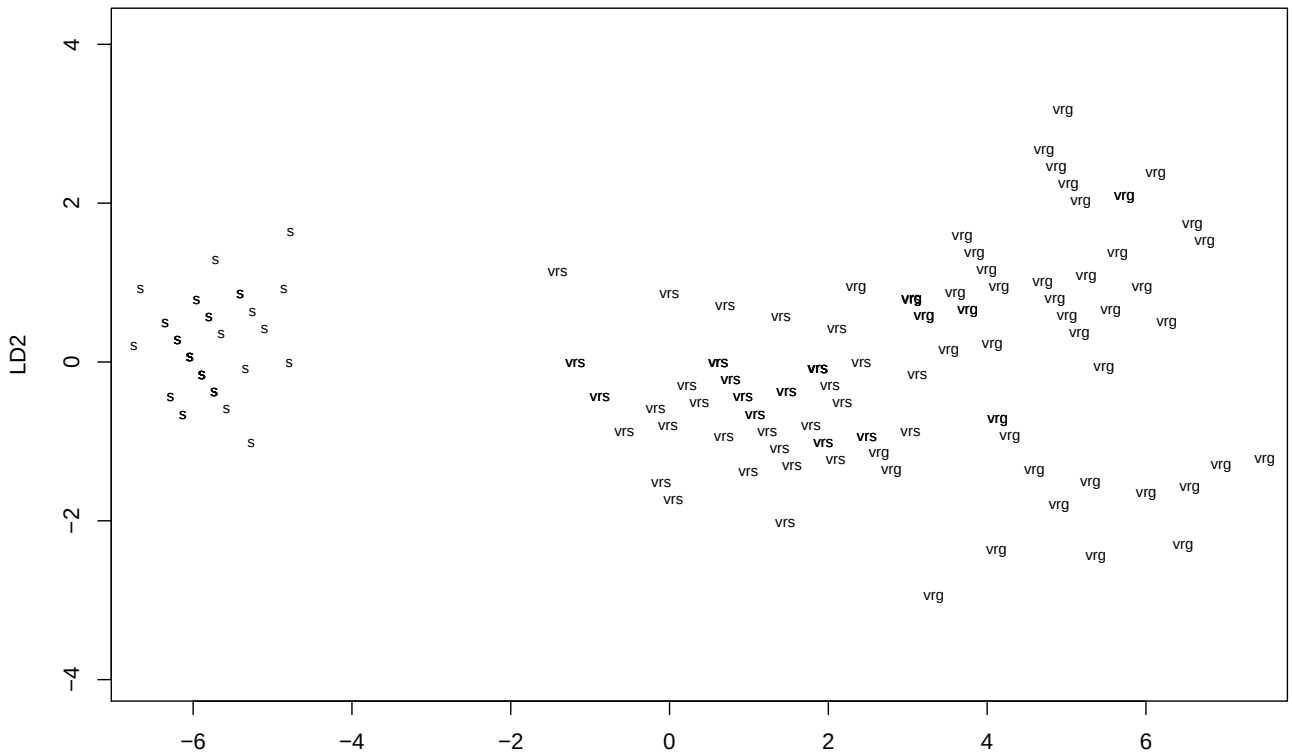
```
##                    LD1        LD2
## Petal.Length 1.544371 -2.161222
## Petal.Width  2.402394  5.042599
```

```
# Separation between populations
iris.lda$svd^2/sum(iris.lda$svd^2)
```

```
## [1] 0.99470499 0.00529501
```

Linear discriminant functions

```
plot(iris.lda, abbrev=1)
```



Linear discriminant scores

```
iris.pred <- predict(iris.lda)
# class prediction (class with maximum post prob)
head(iris.pred$class, n=5)
```

```
## [1] setosa setosa setosa setosa setosa
## Levels: setosa versicolor virginica
```

```
# posterior probability of each class
head(iris.pred$posterior, n=5)
```

```
##   setosa   versicolor    virginica
## 1      1 8.750491e-12 4.742801e-26
## 2      1 8.750491e-12 4.742801e-26
## 3      1 2.640992e-12 9.514213e-27
## 4      1 2.899331e-11 2.364269e-25
## 5      1 8.750491e-12 4.742801e-26
```

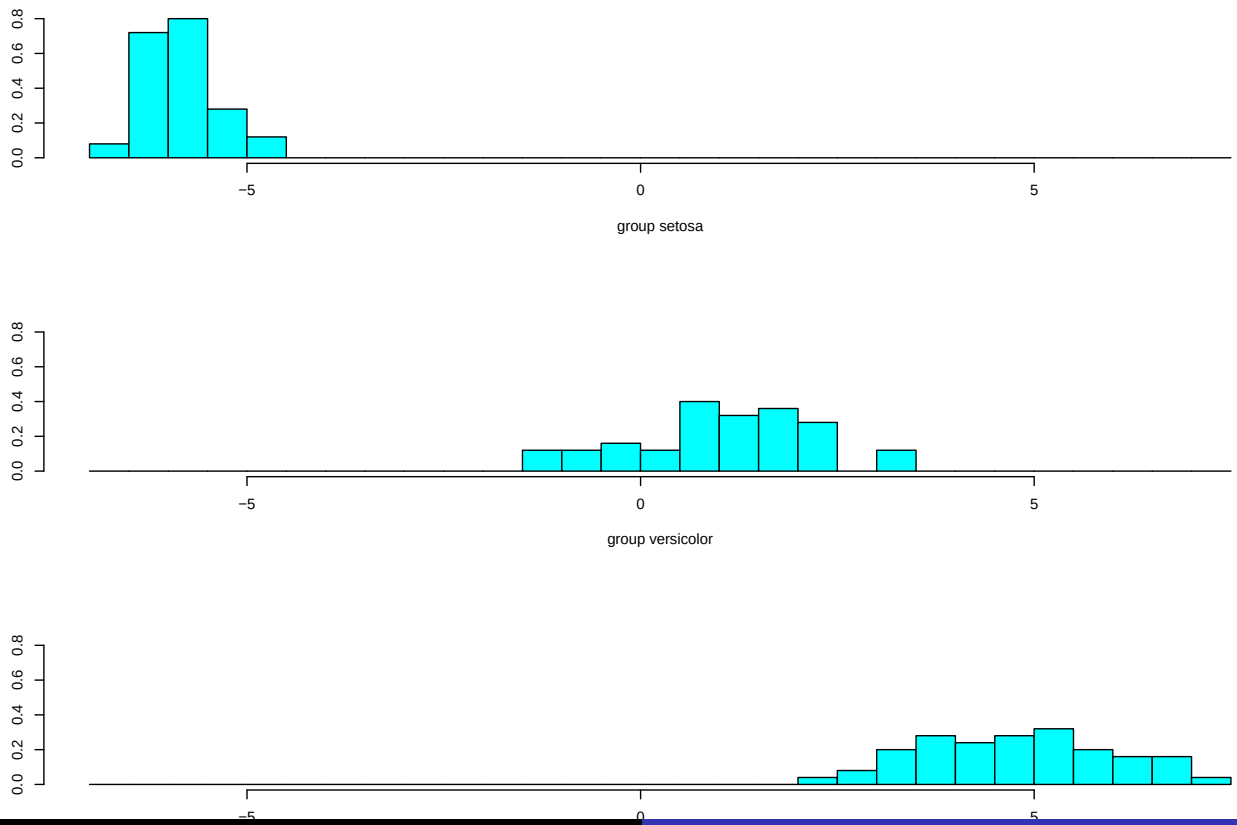
Linear discriminant scores

```
# Canonical scores
head(iris.pred$x, n=5)
```

```
##          LD1         LD2
## 1 -6.042418  0.05692487
## 2 -6.042418  0.05692487
## 3 -6.196856  0.27304711
## 4 -5.887981 -0.15919736
## 5 -6.042418  0.05692487
```

Linear discriminant scores

```
ldahist(iris.pred$x[,1], iris$Species)
```



Linear predictive discriminant analysis

- Two populations/one variable
- Two populations/two variables
- g populations/p variables
- Error rates

Definitions

Optimal error rate

Theoretical error rate when affectation rule is based on real population parameters. Function of Mahalanobis' distance between centroids of populations.

Actual error rate

Observed error rate when affecting new individuals from the same mix of populations used to create affectation rules

Expected actual error rate

Mathematical expectation of the actual error rate

Parametric estimators

Only for some situations

- function of the classification rule
- function of the (unknown) parameters of the populations

Example : LCF, $g = 2, \quad p_1 = p_2$
Optimal error rate : $eo = \Phi(-\Delta/2)$

Non parametric estimators

Percent of misclassified observations

- resubstitution
- holdout
  - training sample
  - test sample
- leave-n-out
  - K-cross validation
  - jackknife
- bootstrap

Prediction error

Resubstitution confusion matrix

```
iris.err <- table(iris4$Species, iris.pred$class)
iris.err
```

```
##
##                setosa versicolor virginica
##    setosa          50          0         0
##    versicolor       0         48         2
##    virginica        0          4        46
```

```
# resubstitution error rate
1 - sum(diag(iris.err))/sum(iris.err)
```

```
## [1] 0.04
```

Prediction error

Cross validated confusion matrix

```
# compute lda with cross validation
iris.cv <- lda(Species~., data=iris4, CV=TRUE)

iris.ecv <- table(iris4$Species, iris.cv$class)
iris.ecv
```

```
##
##                setosa versicolor virginica
##    setosa          50          0         0
##    versicolor       0         48         2
##    virginica        0          4        46
```

Contents

1 Introduction

2 Linear predictive discriminant analysis

3 Nearest neighbours methods

4 Decision trees

5 Random Forest

6 Spatial smoothing

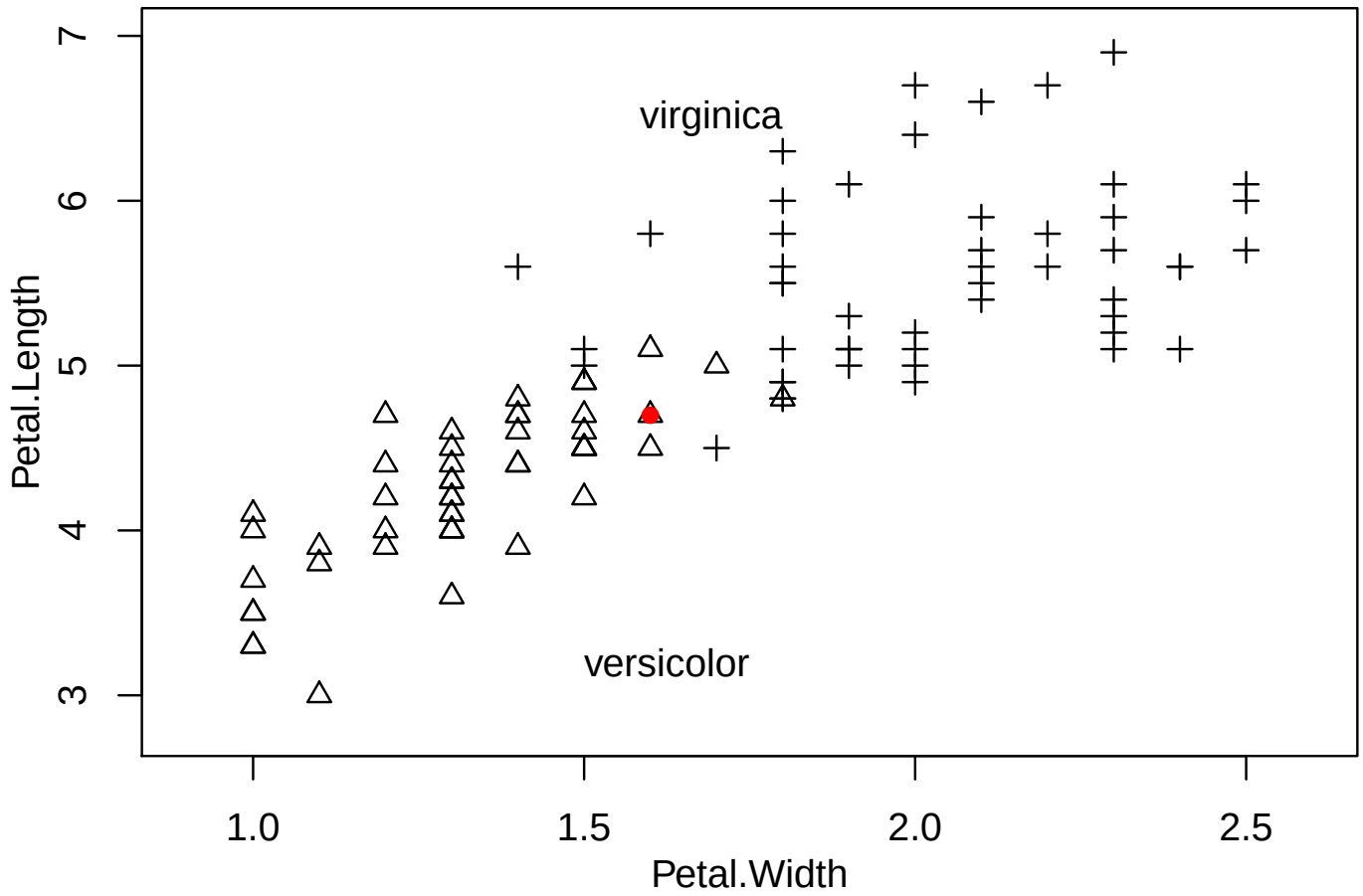
Basics

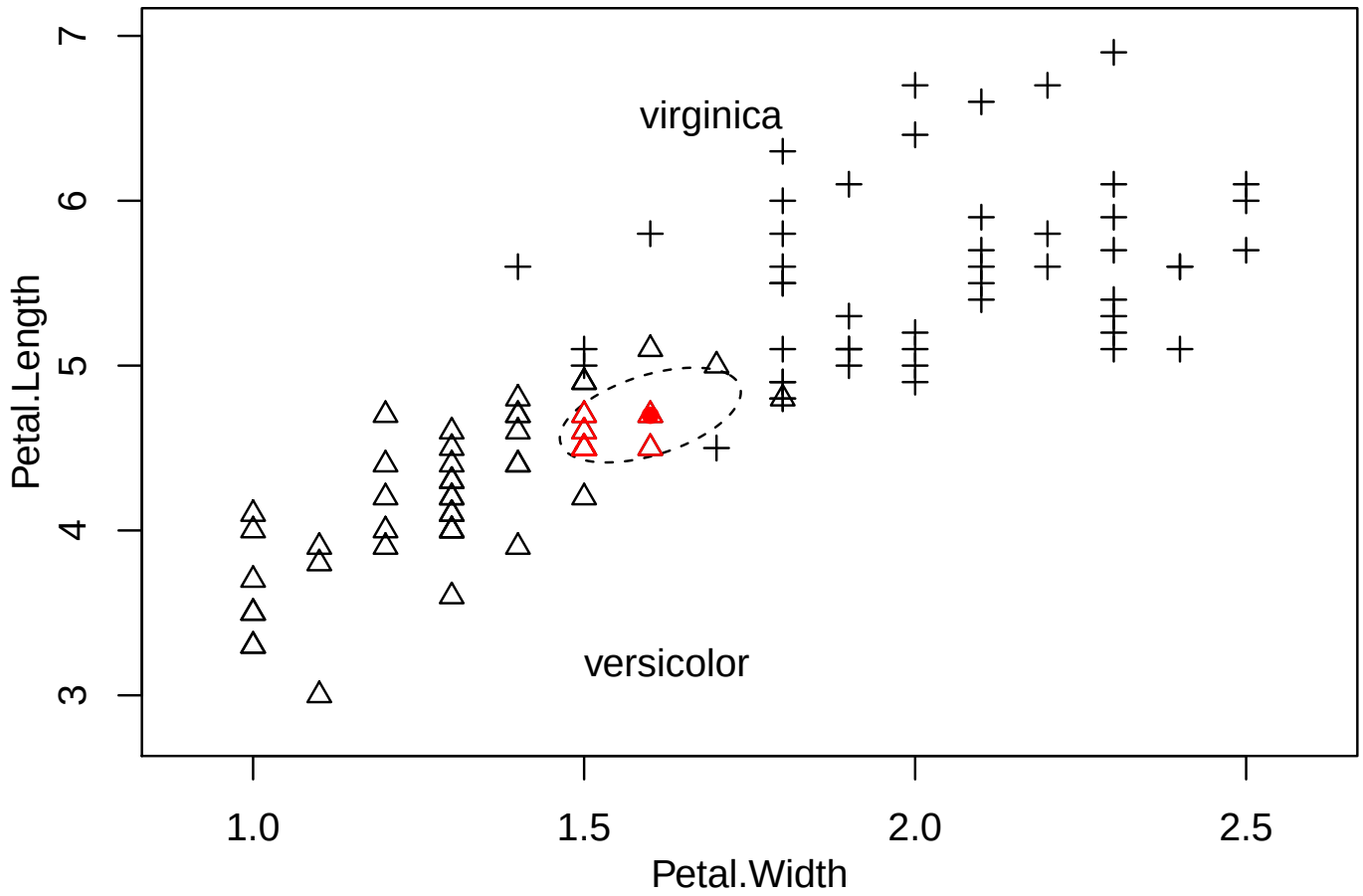
Classification rule

Affect observation i to the population h which is the most represented amongst the r nearest neighbours of this observation

Distance

- euclidian
- Mahalanobis'


virginica
versicolor
Petal.Length
Petal.Width
3
4
5
6
7
1.0
1.5
2.0
2.5


virginica
versicolor
Petal.Length
Petal.Width
3
4
5
6
7
1.0
1.5
2.0
2.5

Pros/cons

- **Pros**
  - no assumptions about distributions
  - easily adapt to complex concepts
- **Cons**
  - no model $\Rightarrow$ need a new computation for each new prediction
  - can be slow for very high number of individuals

Example with R

```
library(class)

# separate data set into training and test sets
set.seed(123)
train <- sample(1:150, 125, replace=FALSE)
iris.trn <- iris4[train,]
iris.tst <- iris4[-train,]

# predict class for test set
cltest <- knn(train=iris.trn[,1:2],
              test=iris.tst[,1:2],
              cl=iris.trn$Species, k = 5)
```

Example with R

```
# confusion matrix
knn.cm <- table(iris.tst$Species, cltest)
knn.cm
```

```
##                cltest
##                 setosa versicolor virginica
##    setosa            9          0         0
##    versicolor        0         11         0
##    virginica         0          0         5
```

```
# actual error rate
1 - sum(diag(knn.cm))/sum(knn.cm)
```

```
## [1] 0
```

Contents

1 Introduction

2 Linear predictive discriminant analysis

3 Nearest neighbours methods

4 Decision trees

5 Random Forest

6 Spatial smoothing

Basics

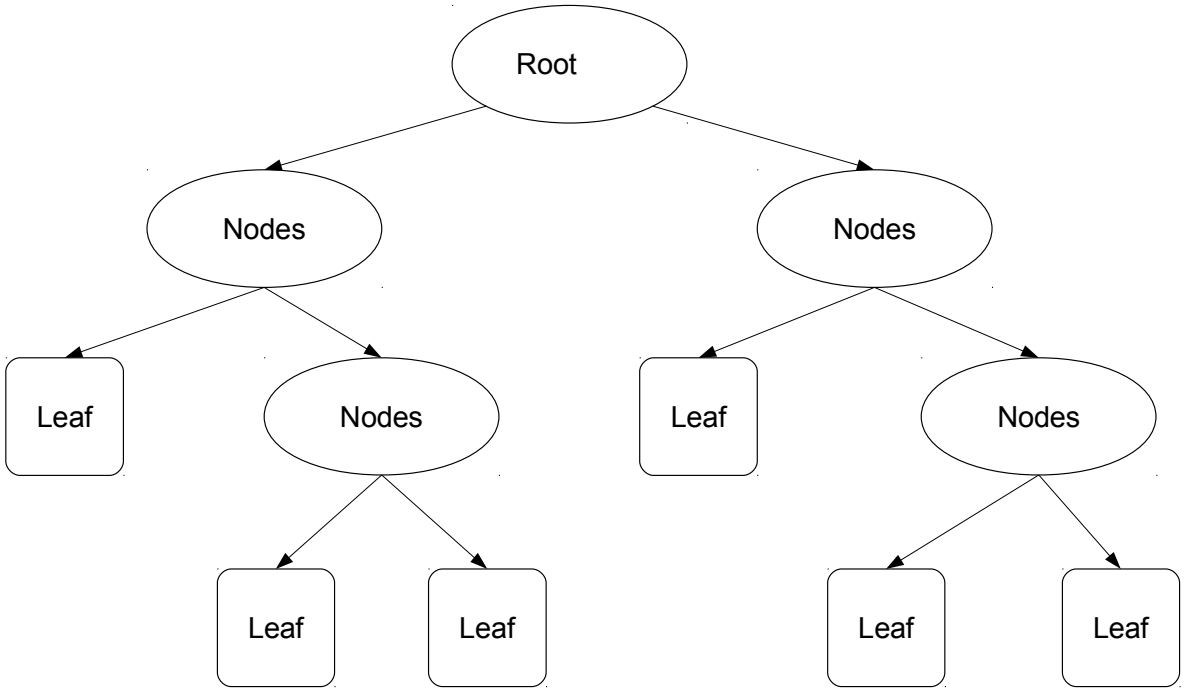
Construction of the classifier

- Recursively partition the observations into groups of increasing homogeneity regarding their populations' distribution.

= decision tree

Classification rules

- Follow the decision path from the *root* of the tree to its *leaves*
- Assign observation to the population with the highest estimated posterior probability


Root
Nodes
Nodes
Leaf
Nodes
Leaf
Nodes
Leaf
Leaf
Leaf
Leaf

Methods

- Number of choices at each node
- Type of decision (univariate, multivariate)
- Homogeneity criteria
- Stopping rule

CART method

- Dichotomic univariate choices
- Homogeneity/splitting criteria : Entropy/Shannon's Index or Gini's index

Spliting criteria

Entropy or Shannon's index

$$I_{Shannon}(E) = \sum_{j=1}^{g} -\frac{n_{j.}}{n_{..}} \log_2 \frac{n_{j.}}{n_{..}}$$

Gini's index

$$I_{Gini}(E) = \sum_{j=1}^{g} \frac{n_{j.}}{n_{..}}(1 - \frac{n_{j.}}{n_{..}})$$

Choose the split which maximize the information gain :

$$gain(E, A) = \Delta I = I(E) - \sum_{i=1}^{p} \frac{n_{.i}}{n_{..}} I(E_i)$$

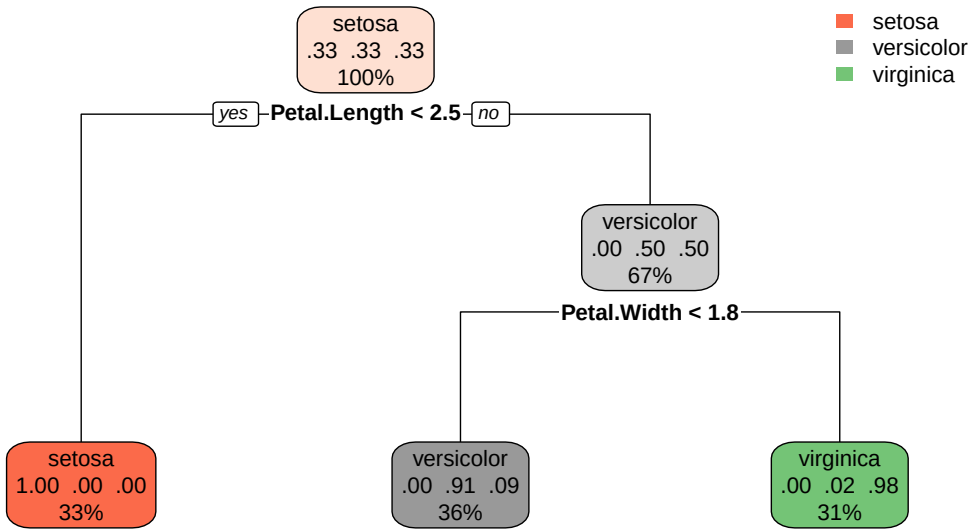
Example with R

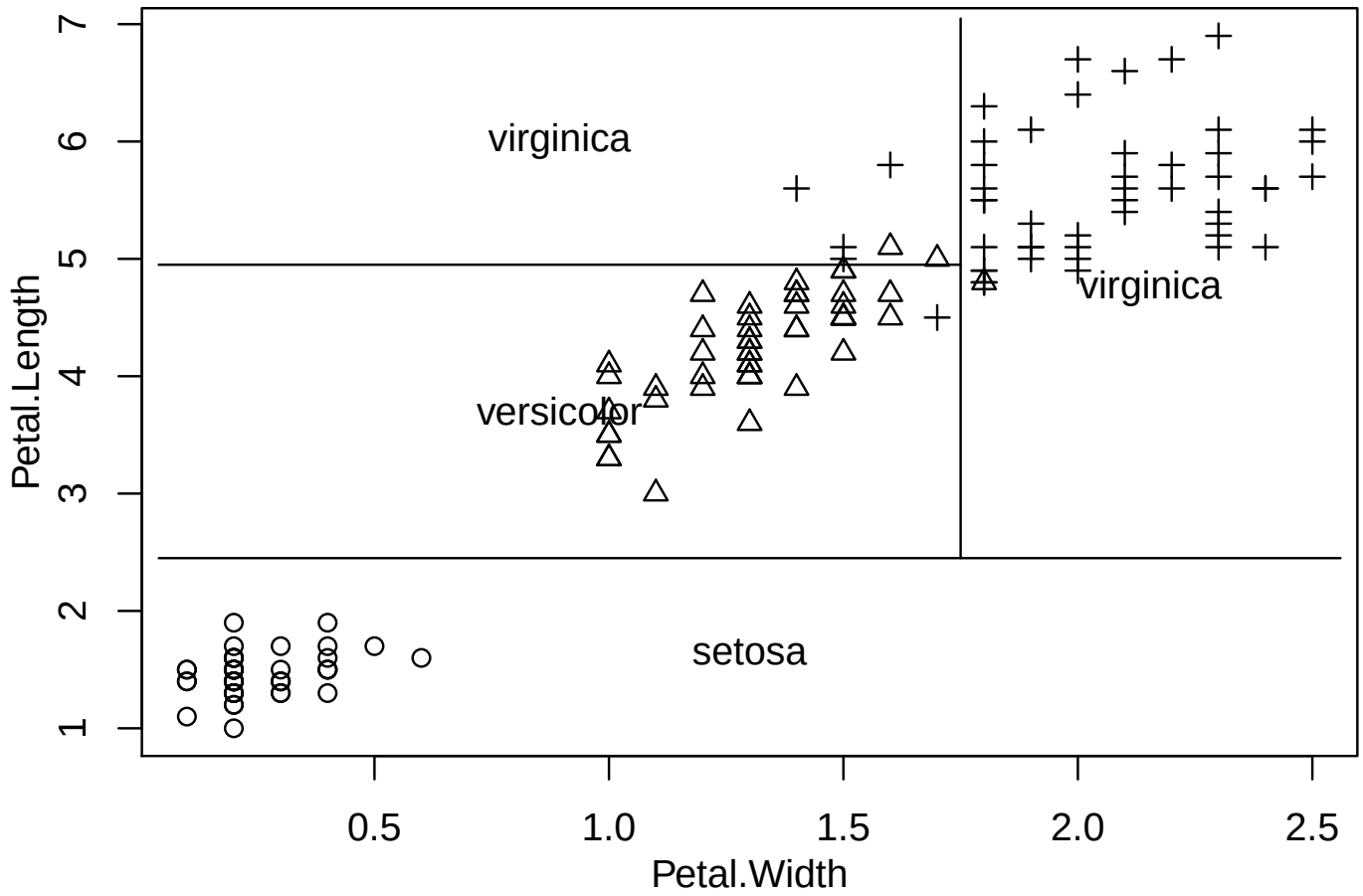
```
library(rpart)
iris.rp <- rpart(Species~., data=iris)
iris.rp
```

```
## n= 150
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)
##   2) Petal.Length< 2.45 50   0 setosa (1.00000000 0.00000000
##   3) Petal.Length>=2.45 100  50 versicolor (0.00000000 0.5000
##     6) Petal.Width< 1.75 54   5 versicolor (0.00000000 0.9074
##     7) Petal.Width>=1.75 46   1 virginica (0.00000000 0.02173
```

Plotting the tree

```
library(rpart.plot)
rpart.plot(iris.rp)
```




virginica
versicolor
setosa
virginica
Petal.Length
Petal.Width
1
2
3
4
5
6
7
0.5
1.0
1.5
2.0
2.5

Optimal size

Fully grown trees tend to overfit data (produce non significant splits), which lower global the prediction performance.

- Stopping rules
  - minimal size
  - minimal gain
  - significance test (χ^2)
- Pruning
  - cut parts of a full grown tree to improve expected error
  - based on penalties on error

rpart R command

```
rpart(formula, data, parms, control, ...)
```

formula : $Y \sim X_1 + X_2 + \ldots + X_p$

parms : list with

- prior component (vector of prior probabilities)
- split component (splitting criteria, gini or information)

control : list with

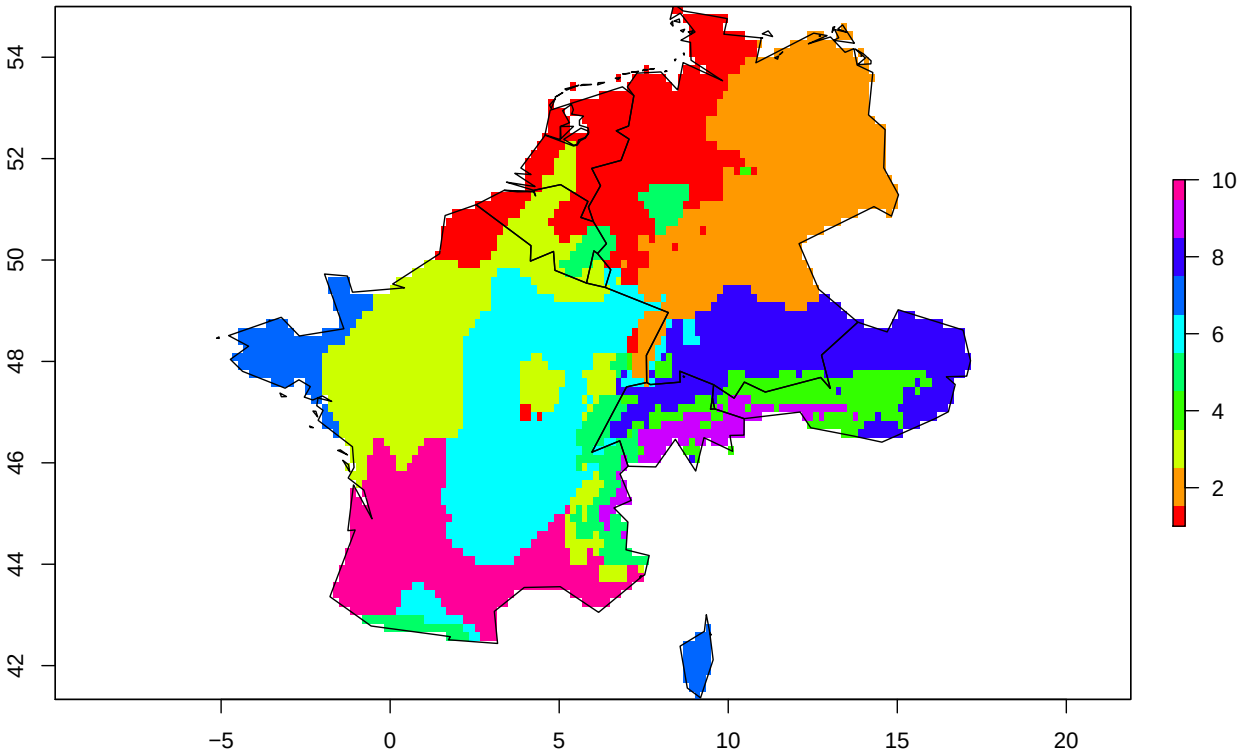
- minsplit minimum number of observations in a node to split it
- minbucket minimum number of observations in any terminal node
- cp complexity parameter. minimum improvement of the splitting criteria to attempt a split

Bigger example : back to climatic typology

```
# get worldwide bioclimatic data
library(raster)
wclim <- getData('worldclim', res=10, var='bio', path="../")

# loading countries borders
library(maptools)
data(wrld_simpl) # simplified world contries borders
# selection of west european region
eur_simpl <- wrld_simpl[wrld_simpl@data$SUBREGION==155,]

# crop climatic data for western europe
climEur <- crop(wclim, bbox(eur_simpl))
climEur <- mask(climEur, eur_simpl)

# extract data for further analysis
climdat <- getValues(climEur)
climdat <- na.omit(as.data.frame(climdat))
```

Bigger example : back to climatic typology

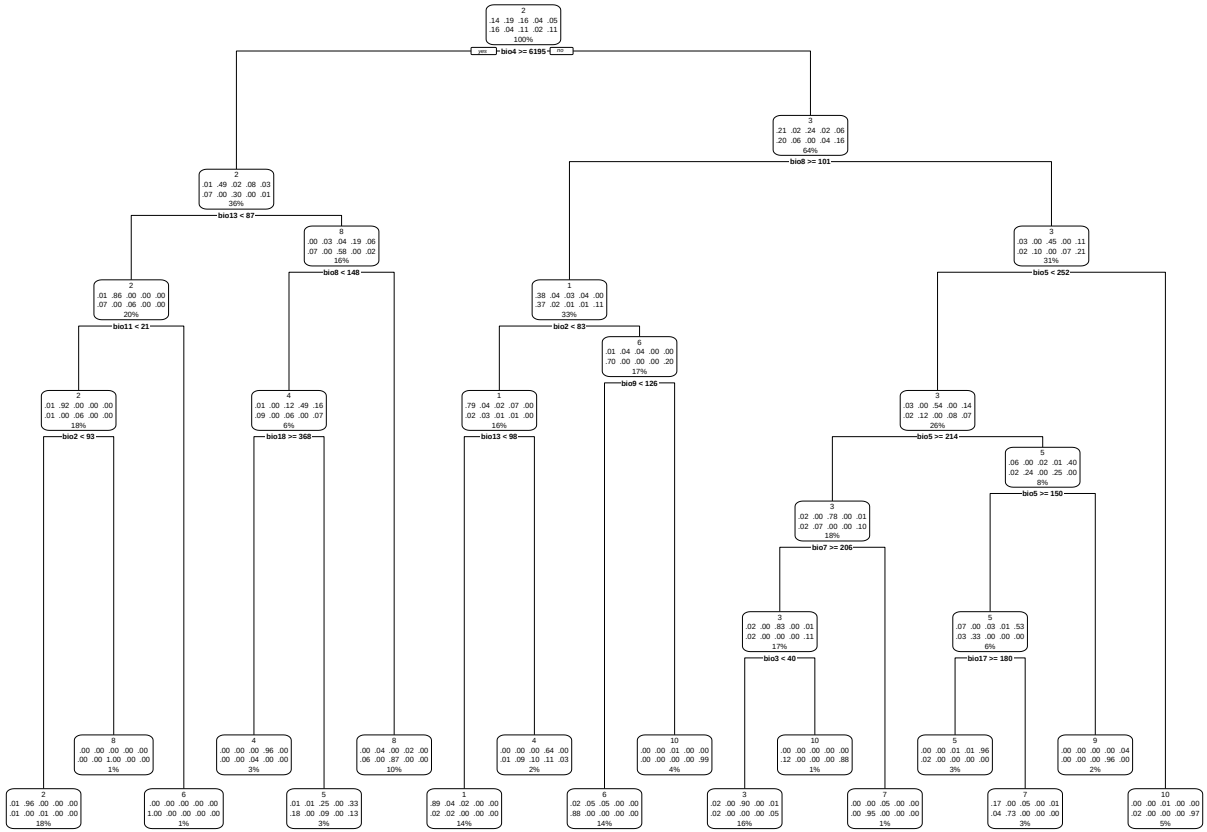
```
climClust <- raster("climTypo10.tif")

# extract groups data
groups <- getValues(climClust)
groups <- na.omit(groups)

plot(climClust, col=rainbow(length(unique(groups))))
plot(eur_simpl, add=TRUE)
```


54
52
50
48
46
44
42
−5
0
5
10
15
20
10
8
6
4
2

```
clim.rp <- rpart(as.factor(groups)~., data=climdat)

rpart.plot(clim.rp)
```

```
# resubstitution confusion matrix
clim.pred <- predict(clim.rp, type="class")
table(groups, clim.pred)
```

```
##        clim.pred
## groups   1   2   3   4   5   6   7   8   9  10
##      1  595   9  17   0   2  12  22   0   0   0
##      2   29 813   0   0   1  31   0  22   0   0
##      3   15   1 666   0  35  32  10   0   0   3
##      4    0   0   0 190   2   2   0  11   0   0
##      5    0   0   9   0 201   2   1   0   4   0
##      6   13  11  12   1  27 636   5  32   0  12
##      7   16   0   2   8   0   0 151   0   0   0
##      8    0   9   0  14  12   1   0 482   0   0
##      9    0   0   0  10   0   0   0   0 100   0
##      10   0   0  38   3  18   0   0   0   0 449
```


bio4 >= 6195
yes
no
bio13 < 87
bio8 >= 101
bio8 < 148
bio5 < 252
bio11 < 21
bio2 < 83
bio9 < 126
bio5 >= 214
bio5 >= 150
bio7 >= 206
bio3 < 40
bio17 >= 180
bio2 < 93
bio18 >= 368
bio13 < 98

Pruning the tree

Cost-complexity pruning evaluate the performance of sequences of (sub)trees, defined by an increasing cost-complexity penalty α.

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|$$

with $R(T)$ the resubstitution error of the tree T and $|\tilde{T}|$ the number of terminal nodes.

For each α the algorithme search the subtree with the minimum $R_\alpha(T)$ and estimate by cross validation the associated error.

The cost-complexity value associated with the **minimal CV error** is then used to prune the initial tree to a smaller one with a better expected error rate.

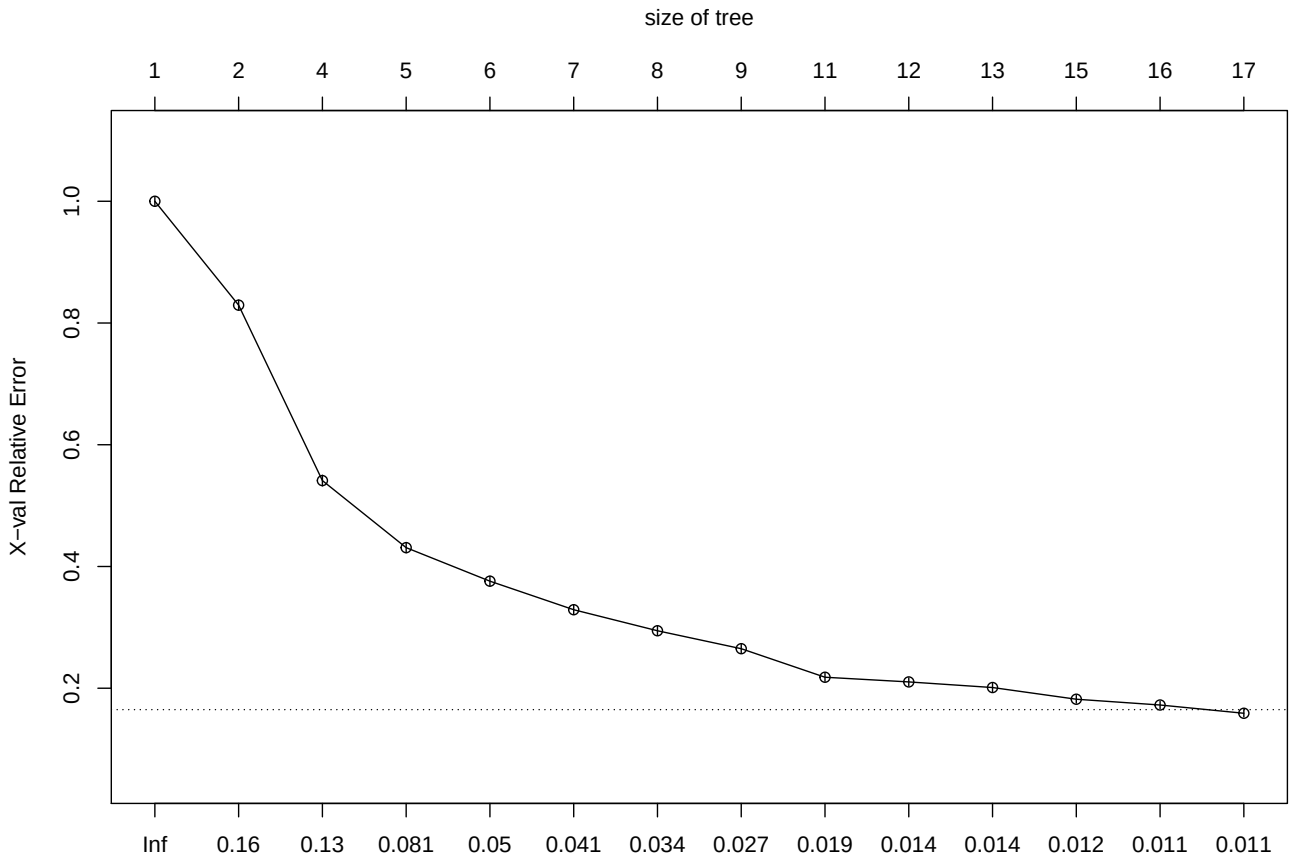
Cost-complexity pruning

```
printcp(clim.rp)
```

```
##
## Classification tree:
## rpart(formula = as.factor(groups) ~ ., data = climdat)
##
## Variables actually used in tree construction:
##  [1] bio11 bio13 bio17 bio18 bio2  bio3  bio4  bio5  bio7  bio8  bio
##
## Root node error: 3903/4799 = 0.81329
##
## n= 4799
##
##          CP nsplit rel error  xerror      xstd
## 1  0.170894      0   1.00000 1.00000 0.0069164
## 2  0.144248      1   0.82911 0.82936 0.0083164
## 3  0.110684      3   0.54061 0.54112 0.0088106
```

Cost-complexity pruning

```
plotcp(clim.rp)
```



Pros & cons

Pros

- Quick even on big data sets
- Easily readable and interpretable
- Classification rules $\rightarrow$ simple logical rules

Cons

- Instability $\Rightarrow$ Random forests

Contents

1 Introduction

2 Linear predictive discriminant analysis

3 Nearest neighbours methods

4 Decision trees

5 Random Forest

6 Spatial smoothing

Basics

CART tree node selection is sensitive to variation in the training data.

Instead of pruning the tree to remove this instability, Random Forests promote this instability by using **resampling of the sample and of the attributes** to build multiple tree predictors (a forest) which are **pooled to increase the robustness** of the prediction.

Algorithm

Starting with a training sample with n observations, p descriptive attributes and a class variable Y

1. For each tree, a bootstrap sample of the original data is used
2. At each node, the attributes selection starts with a random choice of *mtry* attributes, followed by a classical selection based on the partition performance
3. All this is repeated to build *ntree* trees

Prediction is done by aggregating the results of the *ntree* trees (mean for quantitative Y, mode for qualitative Y).

Example with R

```
library(randomForest)
clim.rf <- randomForest(as.factor(groups)~.,
                        importance=TRUE,
                        ntree=1000,
                        data=climdat)

clim.rf
```

```
##
## Call:
##  randomForest(formula = as.factor(groups) ~ ., data = climdat,
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 1.29%
## Confusion matrix:
##      1   2   3   4   5   6   7   8   9  10 class.error
## 1  646   7   1   0   0   3   0   0   0   0 0.016742770
## 2    2 890   0   0   0   2   0   2   0   0 0.006696429
## 3    2   0 752   0   4   1   1   0   0   2 0.013123360
## 4    0   0   0 200   0   0   0   3   2   0 0.024390244
## 5    0   0   3   0 212   0   0   0   2   0 0.023041475
## 6    2   2   0   0   1 741   0   2   0   1 0.010680908
## 7    0   0   2   0   0   0 175   0   0   0 0.011299435
## 8    0   4   0   3   0   1   0 510   0   0 0.015444015
## 9    0   0   0   1   0   0   0   0 109   0 0.009090909
```

Pros & cons

Pros

- Lose the simplicity and readability of single trees
- can be computer intensive for big datasets

Cons

- better robustness and prediction performance
- new information available
  - out-of-bag (OOB) error
  - variable importance

Out-of-bag (OOB) error

Each tree of the random forest is build with a training set obtained by bootstrapping (sample of the same size with replacement) the originial dataset.

In such sample, some observations are present more than one time, and conversely some observations are absent (out-of-bag) from the bootstrap sample (about 33% on average)

Those OOB observations are used as a test set for the corresponding tree to estimate the OOB error rate.

Variable importance

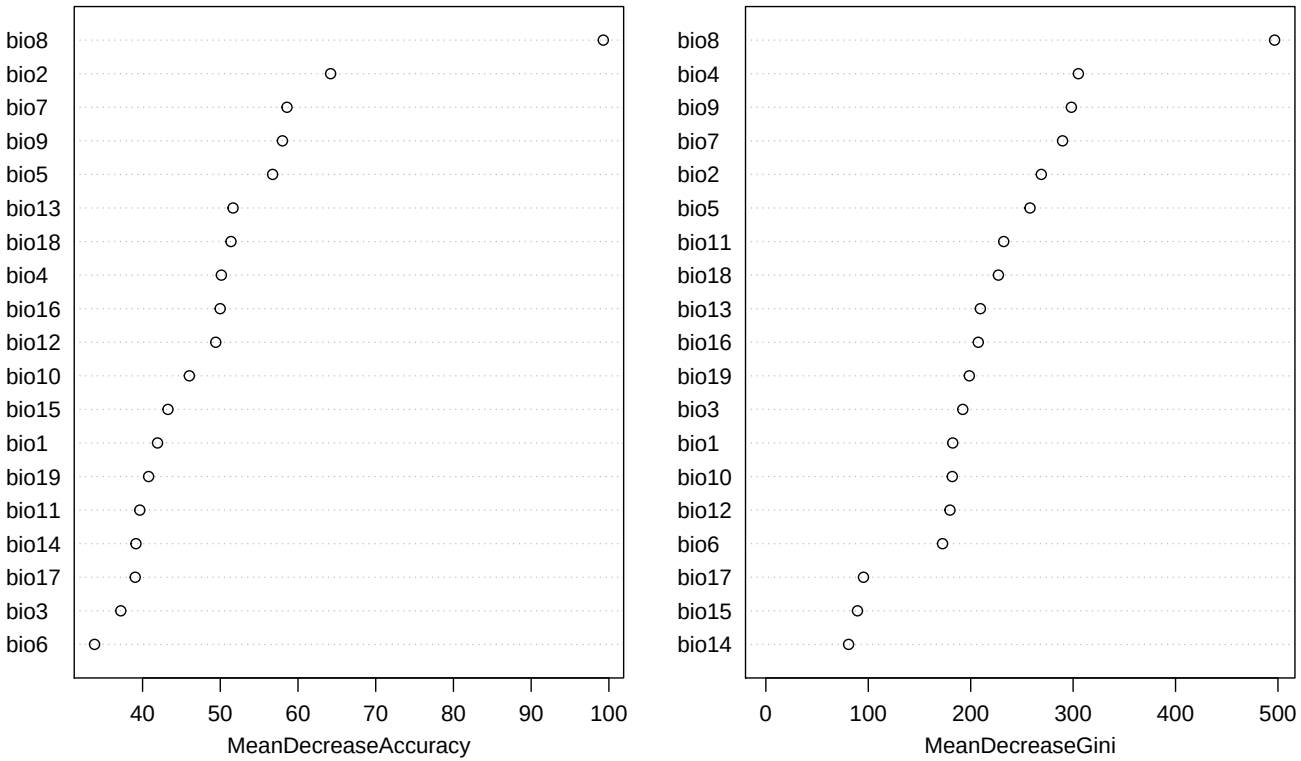
The importance (influence) of a variable in the RF prediction process is estimated by two different methods

- by comparing the OOB performance on the original data and a sample where the value of the variable are randomly permutated (*MeanDeacreaseAccuracy*);
- by summing all the contribution of the variable to the decrease of the splitting criteria (*MeanDecreaseGini*)

Example with R

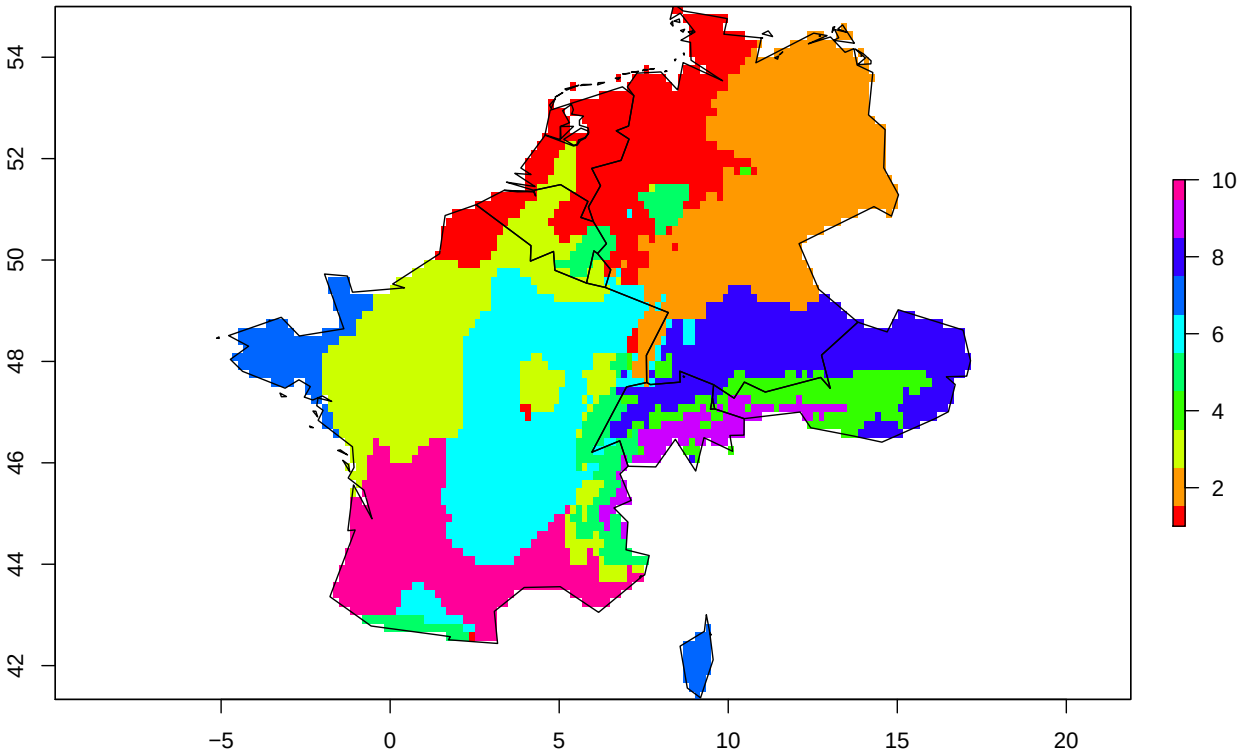
```
# need to use to argument importance=TRUE
# while building the random forest
varImpPlot(clim.rf)
```

clim.rf


bio8
bio2
bio7
bio9
bio5
bio13
bio18
bio4
bio16
bio12
bio10
bio15
bio1
bio19
bio11
bio14
bio17
bio3
bio6
40
50
60
70
80
90
100
MeanDecreaseAccuracy
bio8
bio4
bio9
bio7
bio2
bio5
bio11
bio18
bio13
bio16
bio19
bio3
bio1
bio10
bio12
bio6
bio17
bio15
bio14
0
100
200
300
400
500
MeanDecreaseGini

Back to spatial

```
climRF <- raster(climEur)
values(climRF)[-nai] <- predict(clim.rf)
plot(climRF, col=rainbow(length(unique(groups))))
plot(eur_simpl, add=TRUE)
```


54
52
50
48
46
44
42
−5
0
5
10
15
20
10
8
6
4
2

Contents

1 Introduction

2 Linear predictive discriminant analysis

3 Nearest neighbours methods

4 Decision trees

5 Random Forest

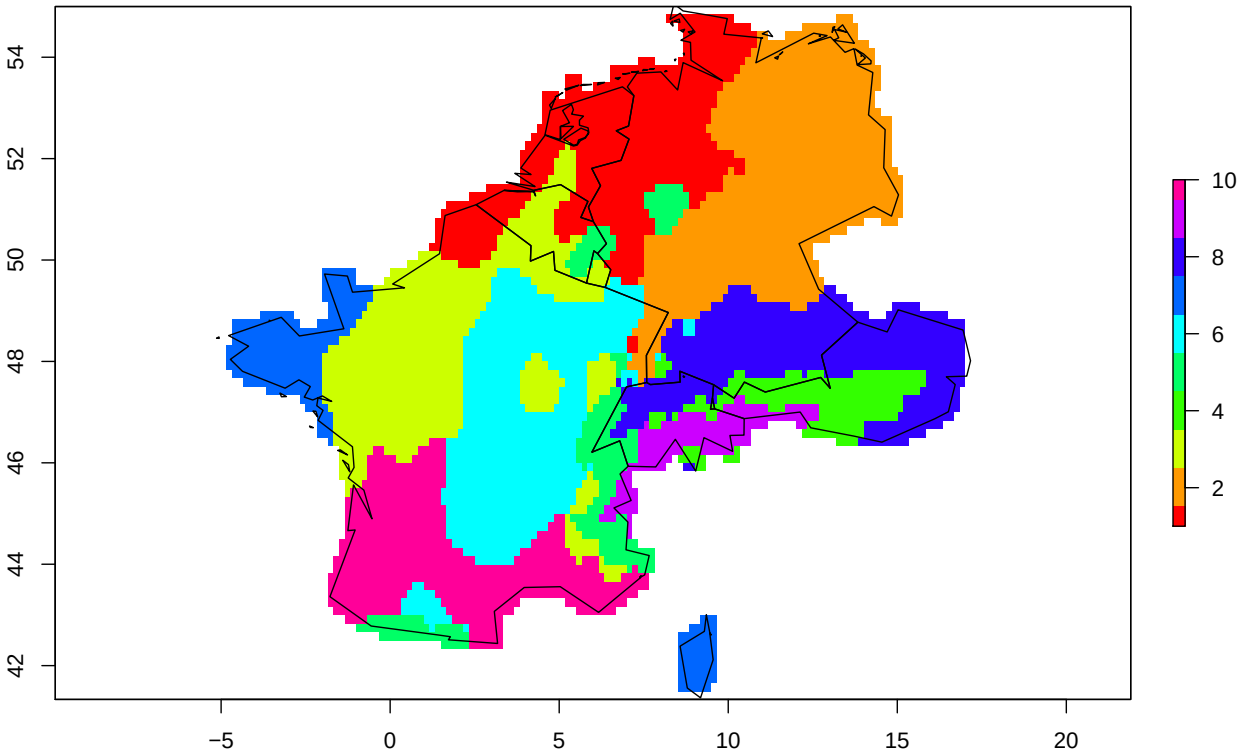
6 Spatial smoothing

Spatial Smoothing of the prediction

As classical classification methods don't use spatial information to predict the classes, you can get local artefacts (lone pixel of one class surrounded by another class), especially in noisy environment.

You can process the raw prediction a posteriori to limit those artifacts by a spatial smoothing. A spatial smoothing apply a function to a defined neighborhood of a point to restimate the value of the point

```
# definition of the smoothing window
# 3 x 3 square, constant weight
windw <- matrix(rep(1, 9), nrow=3)

# take the majority class in the window for each pixel
climRF.sm <- focal(climRF, w=windw, fun=modal,
                   na.rm=T, pad=F)
plot(climRF.sm, col=rainbow(length(unique(groups))))
plot(eur_simpl, add=TRUE)
```


54
52
50
48
46
44
42
−5
0
5
10
15
20
10
8
6
4
2