

An Introduction to Machine Learning

Sudhakaran Prabakaran, Matt Wayland and Chris Penfold

2017-08-30

Contents

1	About the course	5
1.1	Overview	5
1.2	Registration	5
1.3	Prerequisites	5
1.4	Github	6
1.5	License	6
1.6	Contact	6
1.7	Colophon	6
2	Introduction	7
3	Linear models and matrix algebra	9
3.1	Exercises	9
4	Linear and non linear logistic regression	11
4.1	Exercises	11
5	Nearest neighbours	13
5.1	Example one	13
5.2	Example two	13
5.3	Exercises	13
6	Decision trees and random forests	15
6.1	Exercises	15
7	Support vector machines	17
7.1	Exercises	17
8	Artificial neural networks	19
8.1	Exercises	19
9	Dimensionality reduction	21
9.1	Linear Dimensionality Reduction	21
9.2	Nonlinear Dimensionality Reduction	21
9.3	Exercises	21
10	Clustering	23
10.1	Introduction	23
10.2	Distance metrics	23
10.3	Hierarchic methods	23
10.4	Partitioning methods	30
10.5	Summary	34
10.6	Exercises	34

A Resources	35
A.1 Python	35
A.2 Machine learning data set repository	35
B Solutions ch. 3 - Linear models and matrix algebra	37
B.1 Exercise 1	37
B.2 Exercise 2	37
C Solutions ch. 4 - Linear and non-linear logistic regression	39
C.1 Exercise 1	39
C.2 Exercise 2	39
D Solutions ch. 5 - Nearest neighbours	41
D.1 Exercise 1	41
D.2 Exercise 2	41
E Solutions ch. 6 - Decision trees and random forests	43
E.1 Exercise 1	43
E.2 Exercise 2	43
F Solutions ch. 7 - Support vector machines	45
F.1 Exercise 1	45
F.2 Exercise 2	45
G Solutions ch. 8 - Artificial neural networks	47
G.1 Exercise 1	47
G.2 Exercise 2	47
H Solutions ch. 9 - Dimensionality reduction	49
H.1 Exercise 1	49
H.2 Exercise 2	49
I Solutions ch. 10 - Clustering	51
I.1 Exercise 1	51
I.2 Exercise 2	51

Chapter 1

About the course

1.1 Overview

Machine learning gives computers the ability to learn without being explicitly programmed. It encompasses a broad range of approaches to data analysis with applicability across the biological sciences. Lectures will introduce commonly used algorithms and provide insight into their theoretical underpinnings. In the practicals students will apply these algorithms to real biological data-sets using the R language and environment.

During this course you will learn about:

- Some of the core mathematical concepts underpinning machine learning algorithms: matrices and linear algebra; Bayes' theorem.
- Classification (supervised learning): partitioning data into training and test sets; feature selection; logistic regression; support vector machines; artificial neural networks; decision trees; nearest neighbours, cross-validation.
- Exploratory data analysis (unsupervised learning): dimensionality reduction, anomaly detection, clustering.

After this course you should be able to:

- Understand the concepts of machine learning.
- Understand the strengths and limitations of the various machine learning algorithms presented in this course.
- Select appropriate machine learning methods for your data.
- Perform machine learning in R.

1.2 Registration

Bioinformatics Training: An Introduction to Machine Learning

1.3 Prerequisites

- Some familiarity with R would be helpful.
- For an introduction to R see An Introduction to Solving Biological Problems with R course.

1.4 Github

[bioinformatics-training/intro-machine-learning](#)

1.5 License

GPL-3

1.6 Contact

If you have any **comments**, **questions** or **suggestions** about the material, please contact the authors: Sudhakaran Prabakaran, Matt Wayland and Chris Penfold.

1.7 Colophon

This book was produced using the **bookdown** package (Xie, 2017), which was built on top of R Markdown and **knitr** (Xie, 2015).

Chapter 2

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter ??.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))  
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(  
  head(iris, 20), caption = 'Here is a nice table!',  
  booktabs = TRUE  
)
```



Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Linear models and matrix algebra

3.1 Exercises

Solutions to exercises can be found in appendix B

Chapter 4

Linear and non linear logistic regression

4.1 Exercises

Solutions to exercises can be found in appendix C.

Chapter 5

Nearest neighbours

5.1 Example one

5.2 Example two

5.3 Exercises

Solutions to exercises can be found in appendix D.

Chapter 6

Decision trees and random forests

6.1 Exercises

Solutions to exercises can be found in appendix E.

Chapter 7

Support vector machines

7.1 Exercises

Solutions to exercises can be found in appendix F

Chapter 8

Artificial neural networks

8.1 Exercises

Solutions to exercises can be found in appendix G.

Chapter 9

Dimensionality reduction

9.1 Linear Dimensionality Reduction

9.1.1 Principle Component Analysis

9.1.2 Horeshoe effect

9.2 Nonlinear Dimensionality Reduction

9.2.1 t-SNE

9.2.2 Gaussian Process Latent Variable Models

9.2.3 GPLVMs with informative priors

9.3 Exercises

Solutions to exercises can be found in appendix H.

Chapter 10

Clustering

10.1 Introduction

Hierarchic (produce dendrogram) vs partitioning methods

10.2 Distance metrics

Minkowski distance:

$$distance(x, y, p) = \left(\sum_{i=1}^n abs(x_i - y_i)^p \right)^{1/p} \quad (10.1)$$

Graphical explanation of euclidean, manhattan and max (Chebyshev?)

10.2.1 Image segmentation

10.3 Hierarchic methods

10.3.1 Linkage algorithms

Make one section panel of three dendrograms one table

Single linkage - nearest neighbours linkage Complete linkage - furthest neighbours linkage Average linkage - UPGMA (Unweighted Pair Group Method with Arithmetic Mean)

Table 10.1: Example distance matrix

	A	B	C	D
B	2			
C	6	5		
D	10	10	5	
E	9	8	3	4

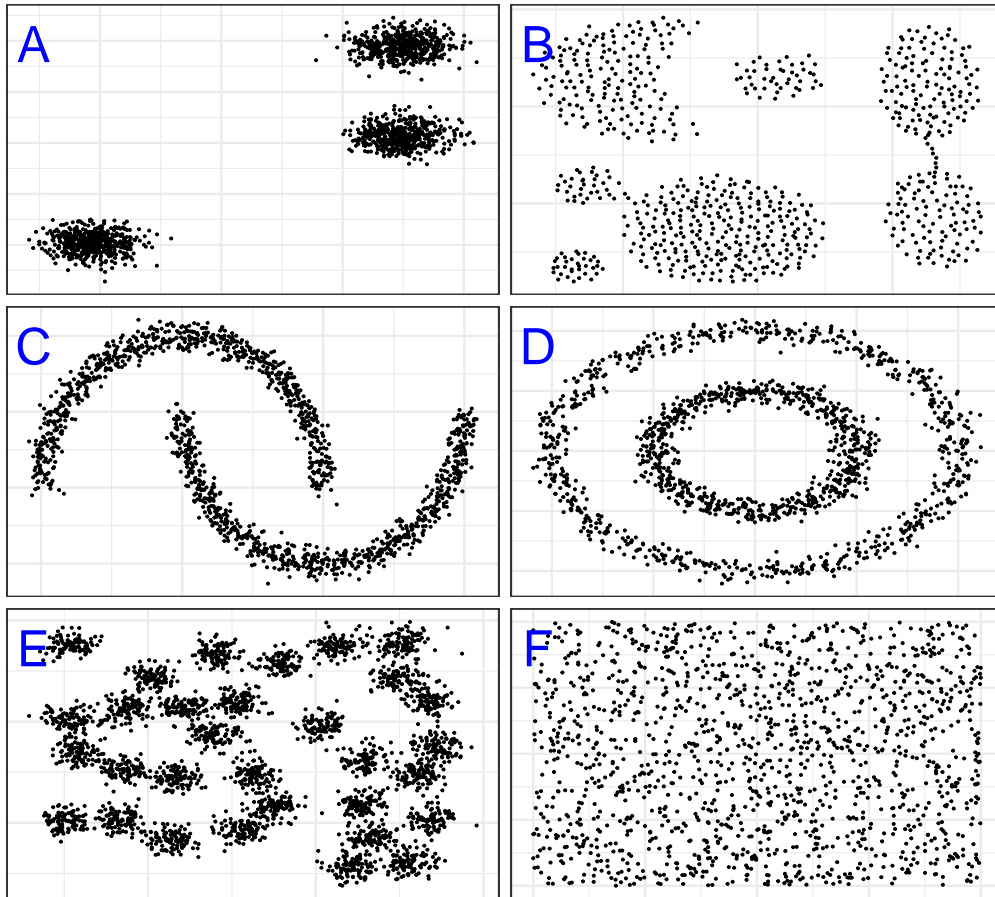
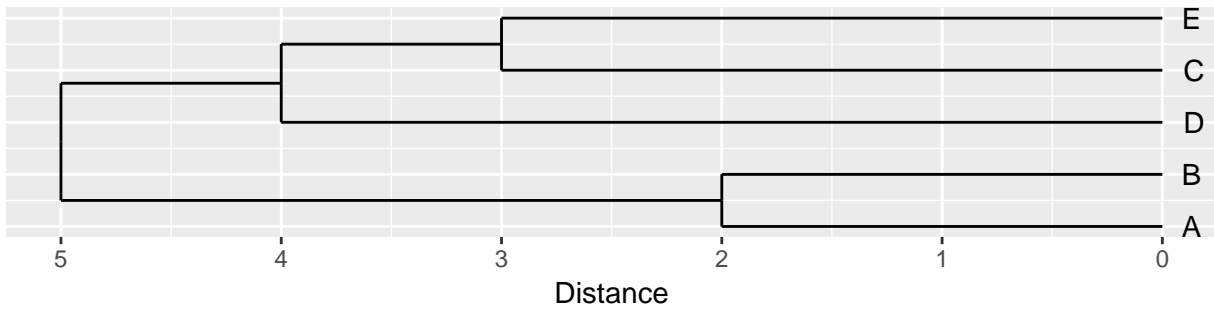


Figure 10.1: Example clusters. **A**, *blobs*; **B**, *aggregation* [Gionis2007]; **C**, *noisy moons*; **D**, *noisy circles*; **E**, *D31* [Veenman2002]; **F**, *no structure*.

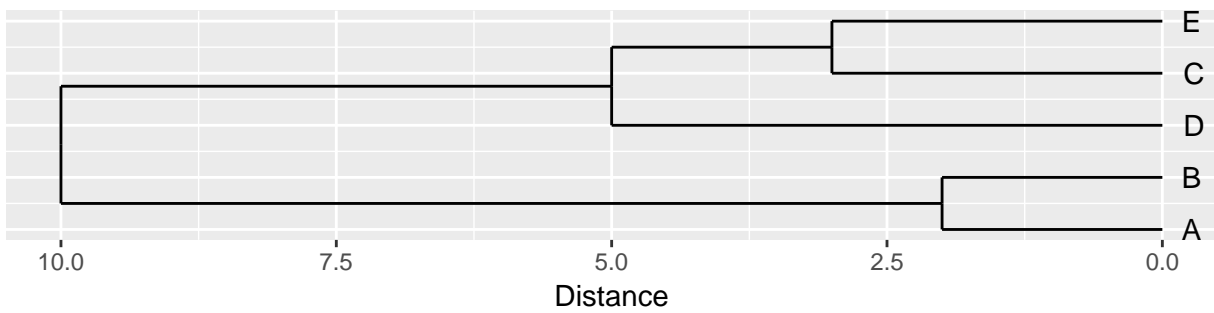
Table 10.2: Merge distances for objects in the example distance matrix using three different linkage methods.

Groups	Single	Complete	Average
A,B,C,D,E	0	0	0
(A,B),C,D,E	2	2	2
(A,B),(C,E),D	3	3	3
(A,B)(C,D,E)	4	5	4.5
(A,B,C,D,E)	5	10	8

Single linkage



Complete linkage



Average linkage

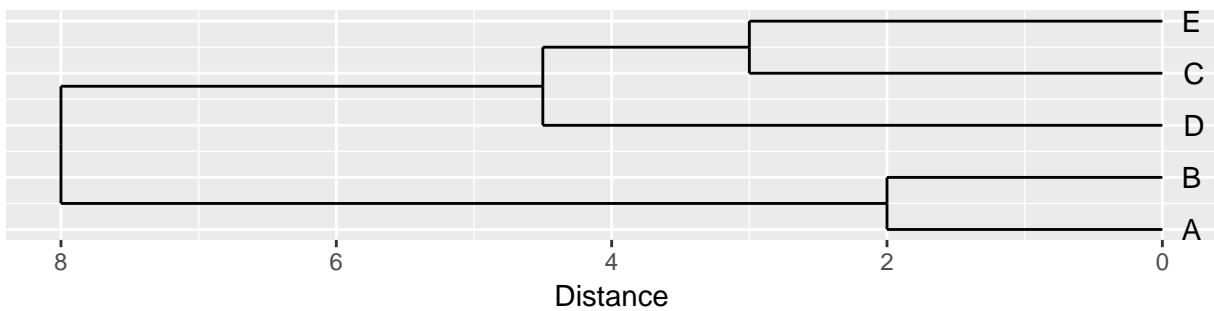


Figure 10.2: Dendrograms for the example distance matrix using three different linkage methods.

10.3.2 Example: clustering toy data sets

```

library(RColorBrewer)
library(dendextend)

##
## -----
## Welcome to dendextend version 1.5.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----

##
## Attaching package: 'dendextend'

## The following object is masked from 'package:ggdendro':
##
##     theme_dendro

## The following object is masked from 'package:stats':
##
##     cutree

library(ggplot2)
library(GGally)

cluster_colours <- brewer.pal(8,"Dark2")

blobs <- read.csv("data/example_clusters/blobs.csv", header=F)

aggregation <- read.table("data/example_clusters/aggregation.txt")
noisy_moons <- read.csv("data/example_clusters/noisy_moons.csv", header=F)
noisy_circles <- read.csv("data/example_clusters/noisy_circles.csv", header=F)
no_structure <- read.csv("data/example_clusters/no_structure.csv", header=F)

hclust_plots <- function(data_set, n){
  d <- dist(data_set[,1:2])
  dend <- as.dendrogram(hclust(d, method="average"))
  clusters <- cutree(dend,n,order_clusters_as_data=F)
  dend <- color_branches(dend, clusters=clusters, col=cluster_colours[1:n])
  clusters <- clusters[order(as.numeric(names(clusters)))]
  labels(dend) <- rep("", length(data_set[,1]))
  ggd <- as.ggdend(dend)
  ggd$nodes <- ggd$nodes[!(1:length(ggd$nodes[,1])),]
  plotPair <- list(ggplot(ggd),
    ggplot(data_set, aes(V1,V2)) + geom_point(col=cluster_colours[clusters], size=0.2))
  return(plotPair)
}

```

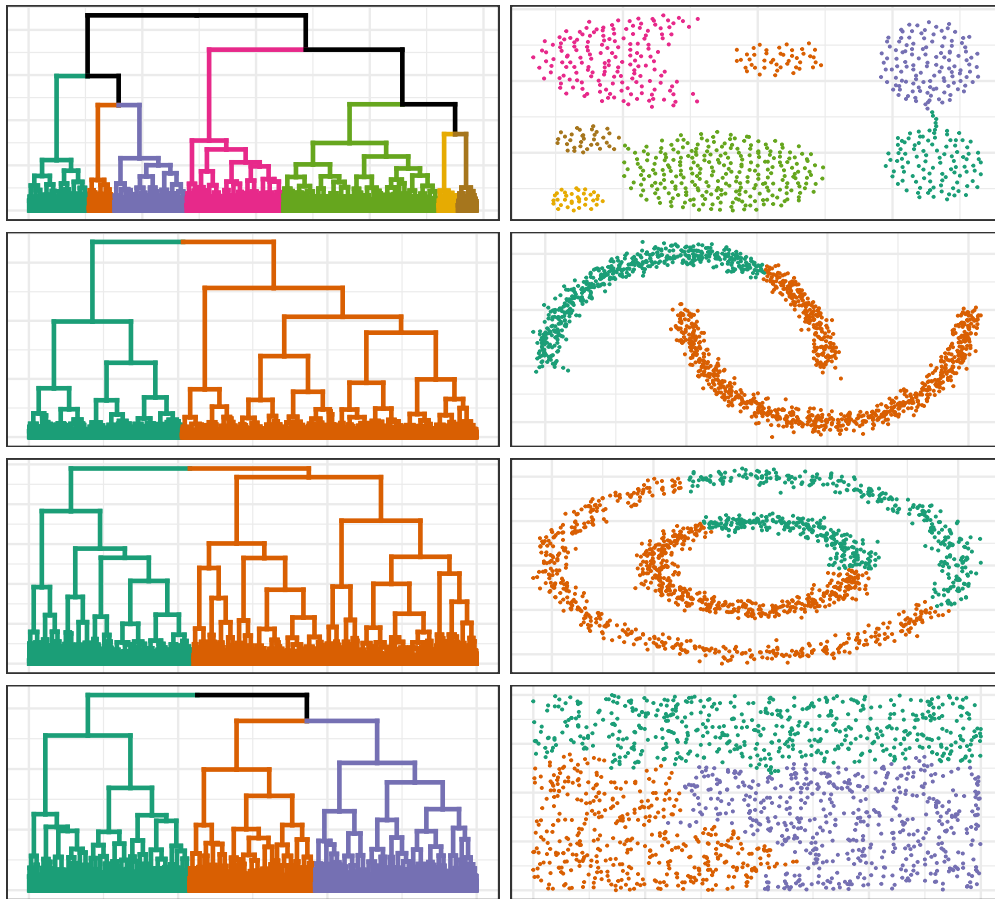


Figure 10.3: Hierarchical clustering of toy data-sets.

```
plotList <- c(
  hclust_plots(aggregation, 7),
  hclust_plots(noisy_moons, 2),
  hclust_plots(noisy_circles, 2),
  hclust_plots(no_structure, 3)
)

pm <- ggmatrix(
  plotList, nrow=4, ncol=2, showXAxisPlotLabels = F, showYAxisPlotLabels = F
) + theme_bw()

pm
```

10.3.3 Example: gene expression profiling of human tissues

Load required libraries

```
library(RColorBrewer)
library(dendextend)
```

Load data

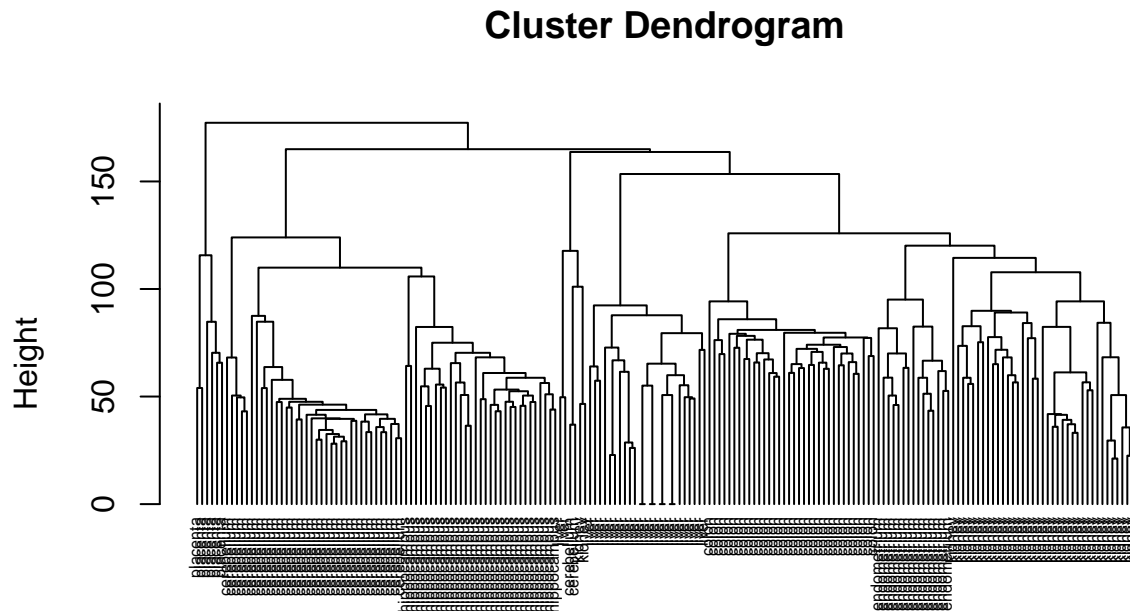


Figure 10.4: Clustering of tissue samples based on gene expression profiles.

```
load("data/tissues_gene_expression/tissuesGeneExpression.rda")
```

Inspect data

```
table(tissue)
```

```
## tissue
## cerebellum      colon endometrium hippocampus      kidney      liver
##           38          34          15          31          39          26
## placenta
##           6
```

```
dim(e)
```

```
## [1] 22215  189
```

Compute distance between each sample

```
d <- dist(t(e))
```

perform hierarchical clustering

```
hc <- hclust(d, method="average")
plot(hc, labels=tissue, cex=0.5, hang=-1, xlab="", sub="")
```

use dendextend library to plot dendrogram with colour labels

```
tissue_type <- unique(tissue)
dend <- as.dendrogram(hc)
dend_colours <- brewer.pal(length(unique(tissue)), "Dark2")
names(dend_colours) <- tissue_type
labels(dend) <- tissue[order.dendrogram(dend)]
labels_colors(dend) <- dend_colours[tissue][order.dendrogram(dend)]
labels_cex(dend) = 0.5
plot(dend, horiz=T)
```

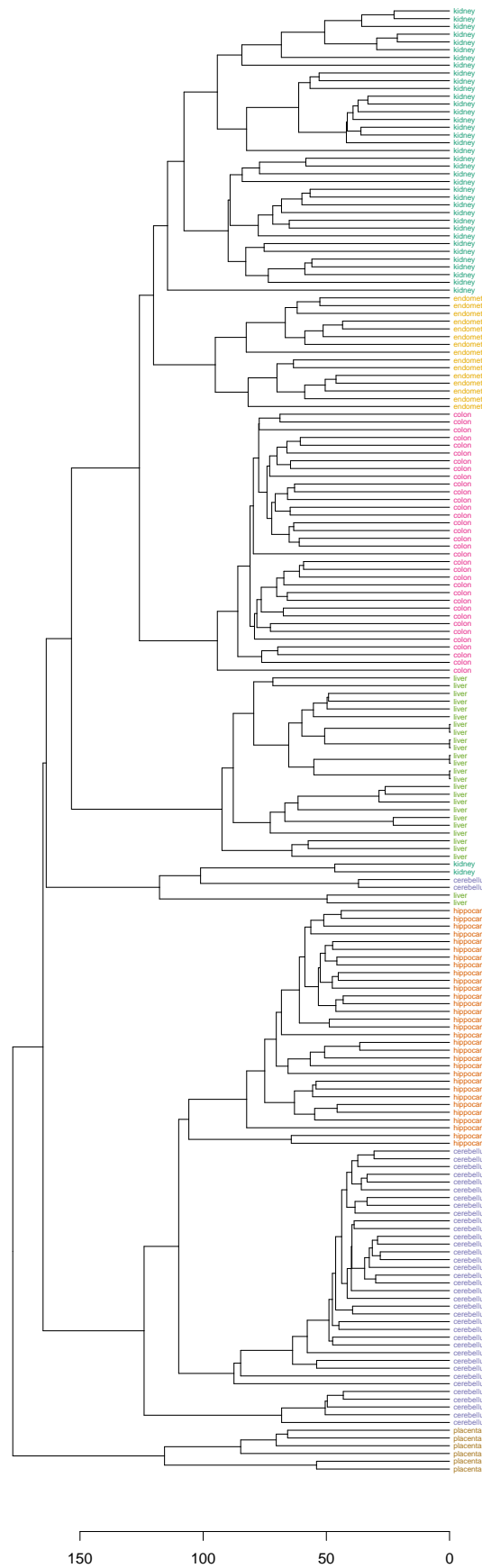


Figure 10.5: Clustering of tissue samples based on gene expression profiles with labels coloured by tissue type.

Define clusters by cutting tree at a specific height

```
plot(dend, horiz=T)
abline(v=125, lwd=2, lty=2, col="blue")
```

```
hclusters <- cutree(dend, h=125)
table(tissue, cluster=hclusters)
```

```
##           cluster
## tissue      1  2  3  4  5  6
## cerebellum  0 36  0  0  2  0
## colon       0  0 34  0  0  0
## endometrium 15  0  0  0  0  0
## hippocampus 0 31  0  0  0  0
## kidney      37  0  0  0  2  0
## liver       0  0  0 24  2  0
## placenta    0  0  0  0  0  6
```

Select a specific number of clusters.

```
plot(dend, horiz=T)
abline(v = heights_per_k.dendrogram(dend)["8"], lwd = 2, lty = 2, col = "blue")
```

```
hclusters <- cutree(dend, k=8)
table(tissue, cluster=hclusters)
```

```
##           cluster
## tissue      1  2  3  4  5  6  7  8
## cerebellum  0 31  0  0  2  0  5  0
## colon       0  0 34  0  0  0  0  0
## endometrium 0  0  0  0  0 15  0  0
## hippocampus 0 31  0  0  0  0  0  0
## kidney      37  0  0  0  2  0  0  0
## liver       0  0  0 24  2  0  0  0
## placenta    0  0  0  0  0  0  0  6
```

10.4 Partitioning methods

10.4.1 K-means

Pseudocode

to illustrate range of different types of data that can be clustered - image segmentation

10.4.2 DBSCAN

Density-based spatial clustering of applications with noise

10.4.3 Gene expression

tissue types?

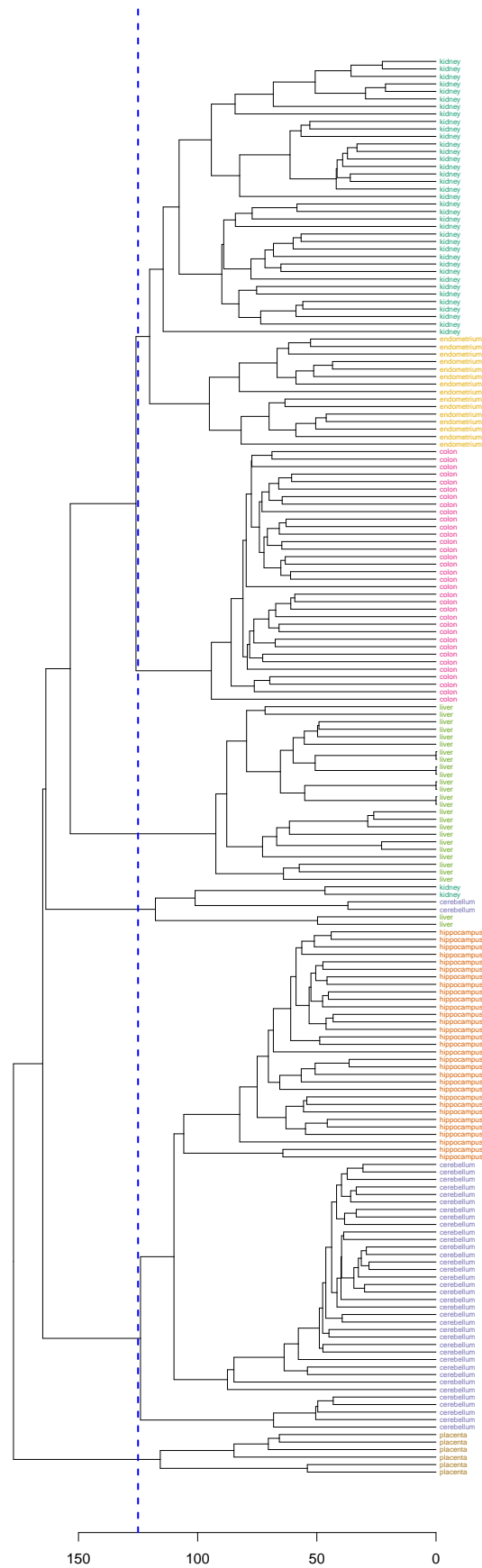


Figure 10.6: Clusters found by cutting tree at a height of 125

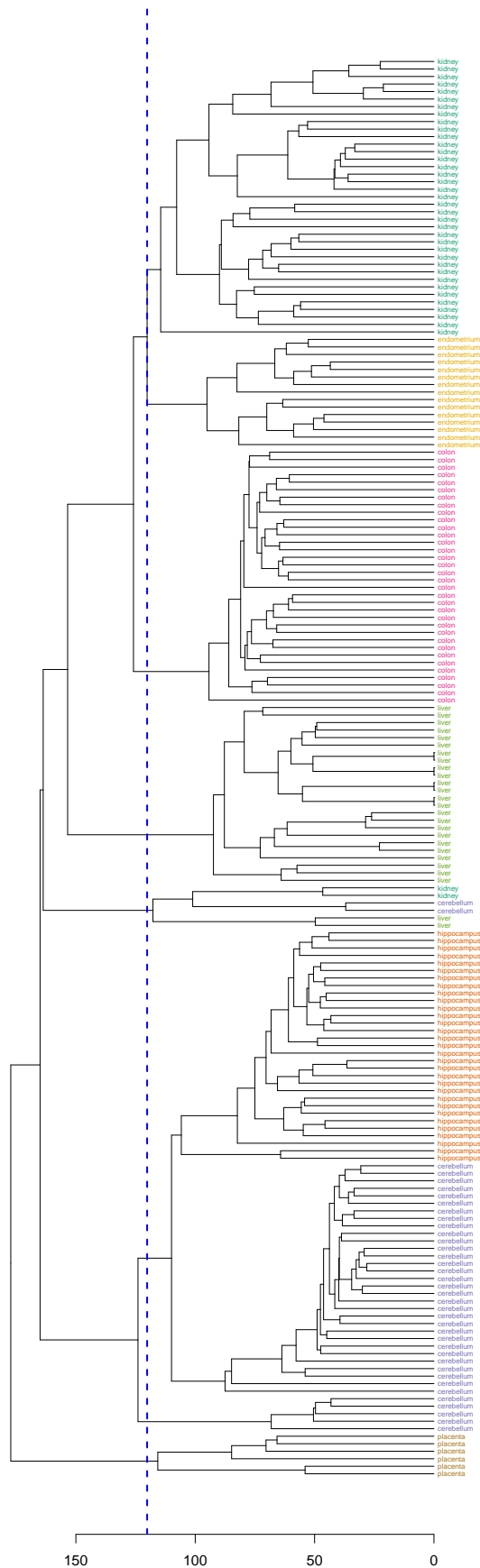


Figure 10.7: Selection of eight clusters from the dendrogram

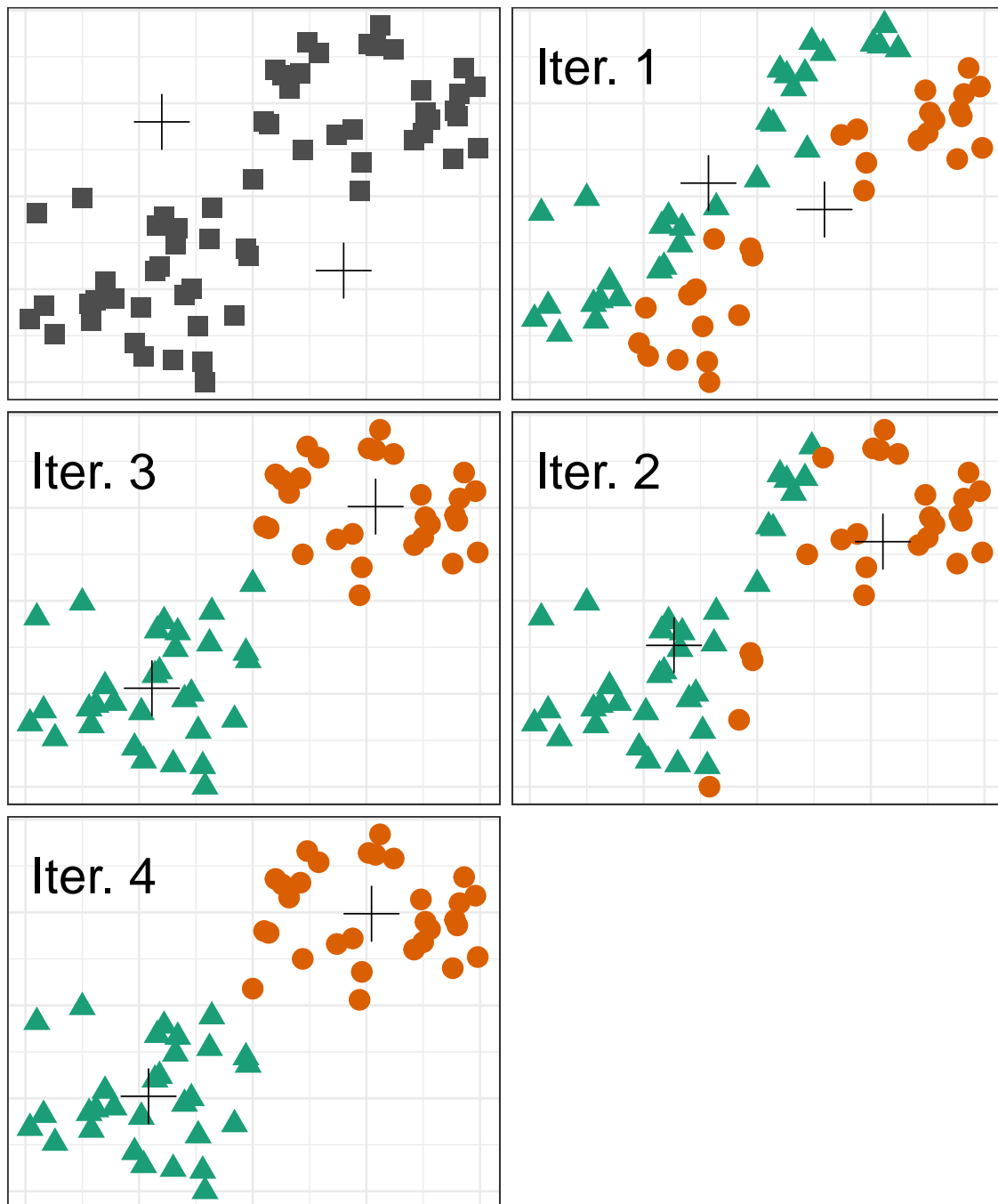


Figure 10.8: Iterations of the k-means algorithm

10.5 Summary

10.5.1 Applications

10.5.2 Strengths

10.5.3 Limitations

10.6 Exercises

Exercise solutions: I

Solutions to exercises can be found in appendix I.

Appendix A

Resources

A.1 Python

[scikit-learn](#)

A.2 Machine learning data set repository

[mldata.org](#)

This repository manages the following types of objects:

- Data Sets - Raw data as a collection of similarly structured objects.
- Material and Methods - Descriptions of the computational pipeline.
- Learning Tasks - Learning tasks defined on raw data.
- Challenges - Collections of tasks which have a particular theme.

Appendix B

Solutions ch. 3 - Linear models and matrix algebra

Solutions to exercises of chapter 3.

B.1 Exercise 1

B.2 Exercise 2

Appendix C

Solutions ch. 4 - Linear and non-linear logistic regression

Solutions to exercises of chapter 4.

C.1 Exercise 1

C.2 Exercise 2

Appendix D

Solutions ch. 5 - Nearest neighbours

Solutions to exercises of chapter 5.

D.1 Exercise 1

D.2 Exercise 2

Appendix E

Solutions ch. 6 - Decision trees and random forests

Solutions to exercises of chapter 6.

E.1 Exercise 1

E.2 Exercise 2

Appendix F

Solutions ch. 7 - Support vector machines

Solutions to exercises of chapter 7.

F.1 Exercise 1

F.2 Exercise 2

Appendix G

Solutions ch. 8 - Artificial neural networks

Solutions to exercises of chapter 8.

G.1 Exercise 1

G.2 Exercise 2

Appendix H

Solutions ch. 9 - Dimensionality reduction

Solutions to exercises of chapter 9.

H.1 Exercise 1

H.2 Exercise 2

Appendix I

Solutions ch. 10 - Clustering

Solutions to exercises of chapter 10.

I.1 Exercise 1

I.2 Exercise 2

Bibliography

- Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.
- Xie, Y. (2017). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.4.