

Data Clustering with R

Yanchang Zhao

<http://www.RDataMining.com>

R and Data Mining Course
Canberra, Australia

10 December 2018

Introduction

- Data Clustering with R
- The Iris Dataset

Partitioning Clustering

- The k -Means Clustering
- The k -Medoids Clustering

Hierarchical Clustering

Density-Based clustering

Cluster Validation

Further Readings and Online Resources

- ▶ Data clustering is to partition data into groups, where the data in the same group are similar to one another and the data from different groups are dissimilar [Han and Kamber, 2000].
- ▶ To segment data into clusters so that the *intra-cluster similarity* is maximized and that the *inter-cluster similarity* is minimized.
- ▶ The groups obtained are a partition of data, which can be used for customer segmentation, document categorization, etc.

- ▶ Partitioning Methods
 - ▶ *k*-means clustering: `stats::kmeans()` * and `fpc::kmeansruns()`
 - ▶ *k*-medoids clustering: `cluster::pam()` and `fpc::pamk()`
- ▶ Hierarchical Methods
 - ▶ Divisive hierarchical clustering: DIANA, `cluster::diana()`,
 - ▶ Agglomerative hierarchical clustering: `cluster::agnes()`, `stats::hclust()`
- ▶ Density based Methods
 - ▶ DBSCAN: `fpc::dbscan()`
- ▶ Cluster Validation
 - ▶ Packages *clValid*, *cclust*, *NbClust*


*`package_name::function_name()`

†Chapter 6 - Clustering, in *R and Data Mining: Examples and Case Studies*.

The iris dataset [Frank and Asuncion, 2010] consists of 50 samples from each of three classes of iris flowers. There are five attributes in the dataset:

- ▶ sepal length in cm,
- ▶ sepal width in cm,
- ▶ petal length in cm,
- ▶ petal width in cm, and
- ▶ class: Iris Setosa, Iris Versicolour, and Iris Virginica.

Detailed description of the dataset can be found at the UCI Machine Learning Repository [‡].

[‡]<https://archive.ics.uci.edu/ml/datasets/Iris> 

Below we have a look at the structure of the dataset with `str()`.

```
str(iris)
## 'data.frame': 150 obs. of 5 variables:
## $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1...
## $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1...
## $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0...
## $ Species      : Factor w/ 3 levels "setosa","versicolor",....
```

- ▶ 150 observations (records, or rows) and 5 variables (or columns)
- ▶ The first four variables are numeric.
- ▶ The last one, `Species`, is categoric (called as “factor” in R) and has three levels of values.

```
summary(iris)
```

```
##   Sepal.Length   Sepal.Width   Petal.Length   Petal.Wid...  
##   Min.      :4.300   Min.      :2.000   Min.      :1.000   Min.      :0....  
##   1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0....  
##   Median :5.800   Median :3.000   Median :4.350   Median :1....  
##   Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1....  
##   3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1....  
##   Max.    :7.900   Max.    :4.400   Max.    :6.900   Max.    :2....  
##           Species  
##   setosa      :50  
##   versicolor:50  
##   virginica  :50  
##  
##  
##
```

Introduction

Data Clustering with R

The Iris Dataset

Partitioning Clustering

The k -Means Clustering

The k -Medoids Clustering

Hierarchical Clustering

Density-Based clustering

Cluster Validation

Further Readings and Online Resources

- ▶ Partitioning the data into k groups first and then trying to improve the quality of clustering by moving objects from one group to another
- ▶ k -means [Alsabti et al., 1998, Macqueen, 1967]: randomly selects k objects as cluster centers and assigns other objects to the nearest cluster centers, and then improves the clustering by iteratively updating the cluster centers and reassigning the objects to the new centers.
- ▶ k -medoids [Huang, 1998]: a variation of k -means for categorical data, where the medoid (i.e., the object closest to the center), instead of the centroid, is used to represent a cluster.
- ▶ PAM and CLARA [Kaufman and Rousseeuw, 1990]
- ▶ CLARANS [Ng and Han, 1994]

- ▶ The result of partitioning clustering is dependent on the selection of initial cluster centers and it may result in a local optimum instead of a global one. (Improvement: run k-means multiple times with different initial centers and then choose the best clustering result.)
- ▶ Tends to result in sphere-shaped clusters with similar sizes
- ▶ Sensitive to outliers
- ▶ Non-trivial to choose an appropriate value for k

- ▶ k -means: a classic partitioning method for clustering
- ▶ First, it selects k objects from the dataset, each of which initially represents a cluster center.
- ▶ Each object is assigned to the cluster to which it is most similar, based on the distance between the object and the cluster center.
- ▶ The means of clusters are computed as the new cluster centers.
- ▶ The process iterates until the criterion function converges.

A typical criterion function is the squared-error criterion, defined as

$$E = \sum_{i=1}^k \sum_{p \in C_i} \|p - m_i\|^2, \quad (1)$$

where E is the sum of square-error, p is a point, and m_i is the center of cluster C_i .

```
## set a seed for random number generation to make the results
## reproducible
set.seed(8953)
## make a copy of iris data
iris2 <- iris
## remove the class label, Species
iris2$Species <- NULL
## run kmeans clustering to find 3 clusters
kmeans.result <- kmeans(iris2, 3)
```

```
## print the clustering result
kmeans.result
```

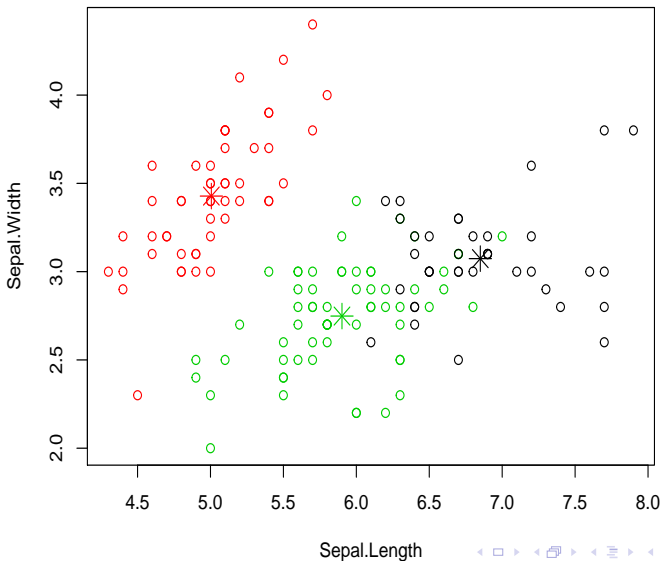
```
## K-means clustering with 3 clusters of sizes 38, 50, 62
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    6.850000    3.073684    5.742105    2.071053
## 2    5.006000    3.428000    1.462000    0.246000
## 3    5.901613    2.748387    4.393548    1.433871
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2...
##  [31] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3...
##  [61] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3...
##  [91] 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1 1 3 3 1 1...
## [121] 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1 1 3...
##
## Within cluster sum of squares by cluster:
## [1] 23.87947 15.15100 39.82097
## (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"...
## [5] "tot.withinss" "betweenss"    "size"         "iter"        ...
## [9] "ifault"
```

Check clustering result against class labels (Species)

```
table(iris$Species, kmeans.result$cluster)
##
##           1  2  3
##  setosa    0 50  0
##  versicolor 2  0 48
##  virginica 36  0 14
```

- ▶ Class “setosa” can be easily separated from the other clusters
- ▶ Classes “versicolor” and “virginica” are to a small degree overlapped with each other.

```
plot(iris2[, c("Sepal.Length", "Sepal.Width")],  
     col = kmeans.result$cluster)  
points(kmeans.result$centers[, c("Sepal.Length", "Sepal.Width")],  
       col = 1:3, pch = 8, cex=2) # plot cluster centers
```



- ▶ `kmeansruns()` in package *fpc* [Hennig, 2014]
- ▶ calls `kmeans()` to perform k -means clustering
- ▶ initializes the k -means algorithm several times with random points from the data set as means
- ▶ estimates the number of clusters by Calinski Harabasz index or average silhouette width

```
library(fpc)
kmeansruns.result <- kmeansruns(iris2)
kmeansruns.result

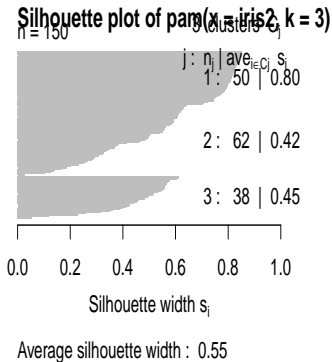
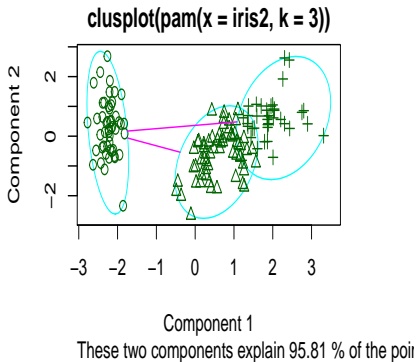
## K-means clustering with 3 clusters of sizes 62, 50, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1    5.901613    2.748387    4.393548    1.433871
## 2    5.006000    3.428000    1.462000    0.246000
## 3    6.850000    3.073684    5.742105    2.071053
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2...
##  [31] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1...
##  [61] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1...
##  [91] 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3 3 1 3 3 3 3 3 3 1 1 3 3...
## [121] 3 1 3 1 3 3 1 1 3 3 3 3 3 1 3 3 3 3 1 3 3 3 1 3 3 3 1...
##
## Within cluster sum of squares by cluster:
## [1] 39.82097 15.15100 23.87947
## (between_SS / total_SS =  88.4 %)
##
## Available components:
##
```

- ▶ Difference from k -means: a cluster is represented with its center in the k -means algorithm, but with the object closest to the center of the cluster in the k -medoids clustering.
- ▶ more robust than k -means in presence of outliers
- ▶ PAM (Partitioning Around Medoids) is a classic algorithm for k -medoids clustering.
- ▶ The CLARA algorithm is an enhanced technique of PAM by drawing multiple samples of data, applying PAM on each sample and then returning the best clustering. It performs better than PAM on larger data.
- ▶ Functions `pam()` and `clara()` in package *cluster* [Maechler et al., 2016]
- ▶ Function `pamk()` in package *fpc* does not require a user to choose k .

```
library(cluster)
# group into 3 clusters
pam.result <- pam(iris2, 3)
# check against actual class label
table(pam.result$clustering, iris$Species)
##
##      setosa versicolor virginica
## 1      50           0           0
## 2       0          48          14
## 3       0           2          36
```

Three clusters:

- ▶ Cluster 1 is species “setosa” and is well separated from the other two.
- ▶ Cluster 2 is mainly composed of “versicolor”, plus some cases from “virginica”.
- ▶ The majority of cluster 3 are “virginica”, with two cases from “versicolor”.



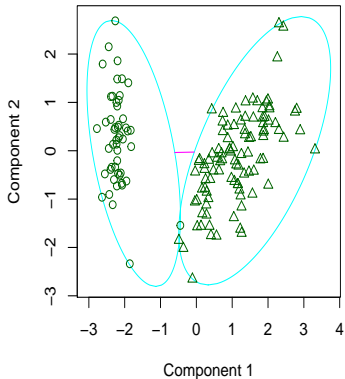
- ▶ The left chart is a 2-dimensional “clusplot” (clustering plot) of the three clusters and the lines show the distance between clusters.
- ▶ The right chart shows their silhouettes. A large s_i (almost 1) suggests that the corresponding observations are very well clustered, a small s_i (around 0) means that the observation lies between two clusters, and observations with a negative s_i are probably placed in the wrong cluster.
- ▶ Silhouette width of cluster 1 is 0.80, which means it is well clustered and separated from other clusters. The other two are of relatively low silhouette width (0.42 and 0.45), and they are somewhat overlapped with each other.

```
library(fpc)
pamk.result <- pamk(iris2)
# number of clusters
pamk.result$nc
## [1] 2

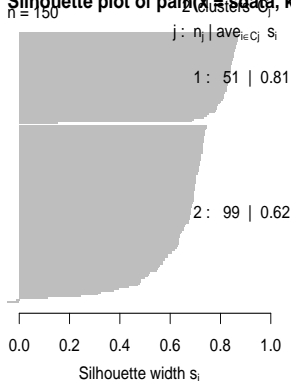
# check clustering against actual class label
table(pamk.result$pamobject$clustering, iris$Species)
##
##      setosa versicolor virginica
##  1      50           1           0
##  2       0          49          50
```

Two clusters:

- ▶ “setosa”
- ▶ a mixture of “versicolor” and “virginica”

`clusplot(pam(x = sdata, k = k, diss = diss))`

These two components explain 95.81 % of the poi

Silhouette plot of `pam(x = sdata, k = k, d`

Average silhouette width : 0.69

- ▶ In this example, the result of `pam()` seems better, because it identifies three clusters, corresponding to three species.

- ▶ In this example, the result of `pam()` seems better, because it identifies three clusters, corresponding to three species.
- ▶ Note that we cheated by setting $k = 3$ when using `pam()`, which is already known to us as the number of species.

Introduction

- Data Clustering with R
- The Iris Dataset

Partitioning Clustering

- The k -Means Clustering
- The k -Medoids Clustering

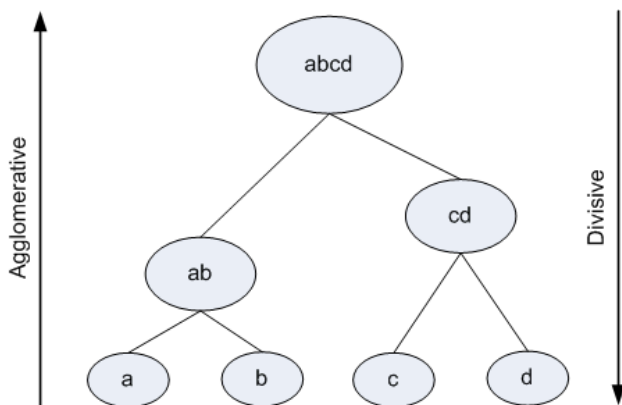
Hierarchical Clustering

Density-Based clustering

Cluster Validation

Further Readings and Online Resources

- ▶ With hierarchical clustering approach, a hierarchical decomposition of data is built in either bottom-up (agglomerative) or top-down (divisive) way.
- ▶ Generally a dendrogram is generated and a user may select to cut it at a certain level to get the clusters.

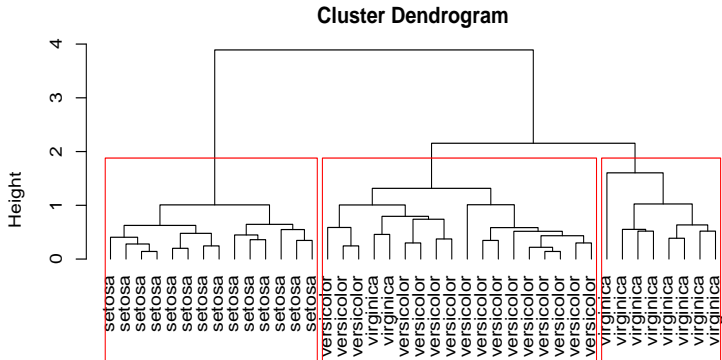


- ▶ With agglomerative clustering, every single object is taken as a cluster and then iteratively the two nearest clusters are merged to build bigger clusters until the expected number of clusters is obtained or when only one cluster is left.
 - ▶ AGENS [Kaufman and Rousseeuw, 1990]
- ▶ Divisive clustering works in an opposite way, which puts all objects in a single cluster and then divides the cluster into smaller and smaller ones.
 - ▶ DIANA [Kaufman and Rousseeuw, 1990]
 - ▶ BIRCH [Zhang et al., 1996]
 - ▶ CURE [Guha et al., 1998]
 - ▶ ROCK [Guha et al., 1999]
 - ▶ Chameleon [Karypis et al., 1999]

In hierarchical clustering, there are four different methods to measure the distance between clusters:

- ▶ *Centroid distance* is the distance between the centroids of two clusters.
- ▶ *Average distance* is the average of the distances between every pair of objects from two clusters.
- ▶ *Single-link distance*, a.k.a. *minimum distance*, is the distance between the two nearest objects from two clusters.
- ▶ *Complete-link distance*, a.k.a. *maximum distance*, is the distance between the two objects which are the farthest from each other from two clusters.

```
set.seed(2835)
# draw a sample of 40 records from the iris data, so that the
# clustering plot will not be over crowded
idx <- sample(1:dim(iris)[1], 40)
iris3 <- iris[idx, ]
# remove class label
iris3$Species <- NULL
# hierarchical clustering
hc <- hclust(dist(iris3), method = "ave")
# plot clusters
plot(hc, hang = -1, labels = iris$Species[idx])
# cut tree into 3 clusters
rect.hclust(hc, k = 3)
# get cluster IDs
groups <- cutree(hc, k = 3)
```

```
dist(iris3)  
hclust (*, "average")
```

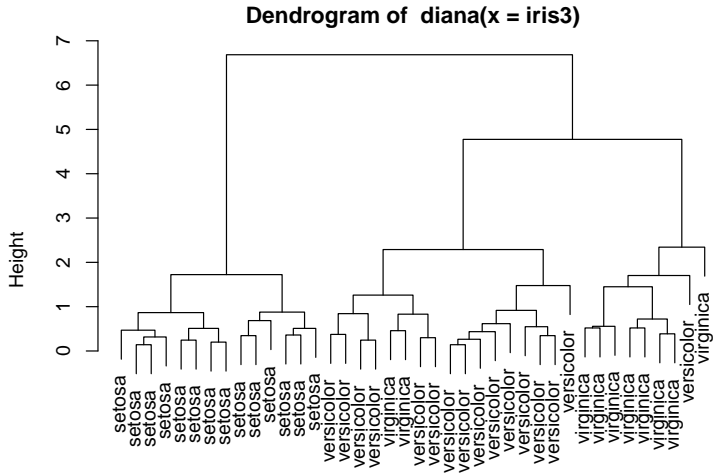
```
hclust(d, method = "complete", members = NULL)
```

- ▶ `method = "ward.D"` or `"ward.D2"`: Ward's minimum variance method aims at finding compact, spherical clusters [R Core Team, 2015].
- ▶ `method = "complete"`: complete-link distance; finds similar clusters.
- ▶ `method = "single"`: single-link distance; adopts a “friends of friends” clustering strategy.
- ▶ `method = "average"`: average distance
- ▶ `method = "centroid"`: centroid distance
- ▶ `method = "median"`:
- ▶ `method = "mcquitty"`:

- ▶ DIANA [Kaufman and Rousseeuw, 1990]: divisive hierarchical clustering
- ▶ Constructs a hierarchy of clusterings, starting with one large cluster containing all observations.
- ▶ Divides clusters until each cluster contains only a single observation.
- ▶ At each stage, the cluster with the largest diameter is selected. (The diameter of a cluster is the largest dissimilarity between any two of its observations.)
- ▶ To divide the selected cluster, the algorithm first looks for its most disparate observation (i.e., which has the largest average dissimilarity to other observations in the selected cluster). This observation initiates the “splinter group”. In subsequent steps, the algorithm reassigns observations that are closer to the “splinter group” than to the “old party”. The result is a division of the selected cluster into two new clusters.

```
library(cluster)
diana.result <- diana(iris3)
```

```
plot(diana.result, which.plots = 2, labels = iris$Species[idx])
```



iris3

Divisive Coefficient = 0.93

Introduction

Data Clustering with R

The Iris Dataset

Partitioning Clustering

The k -Means Clustering

The k -Medoids Clustering

Hierarchical Clustering

Density-Based clustering

Cluster Validation

Further Readings and Online Resources

- ▶ The rationale of density-based clustering is that a cluster is composed of well-connected dense region, while objects in sparse areas are removed as noises.
- ▶ DBSCAN is a typical density-based clustering algorithm, which works by expanding clusters to their dense neighborhood [Ester et al., 1996].
- ▶ Other density-based clustering techniques: OPTICS [Ankerst et al., 1999] and DENCLUE [Hinneburg and Keim, 1998]
- ▶ The advantage of density-based clustering is that it can filter out noise and find clusters of arbitrary shapes (as long as they are composed of connected dense regions).

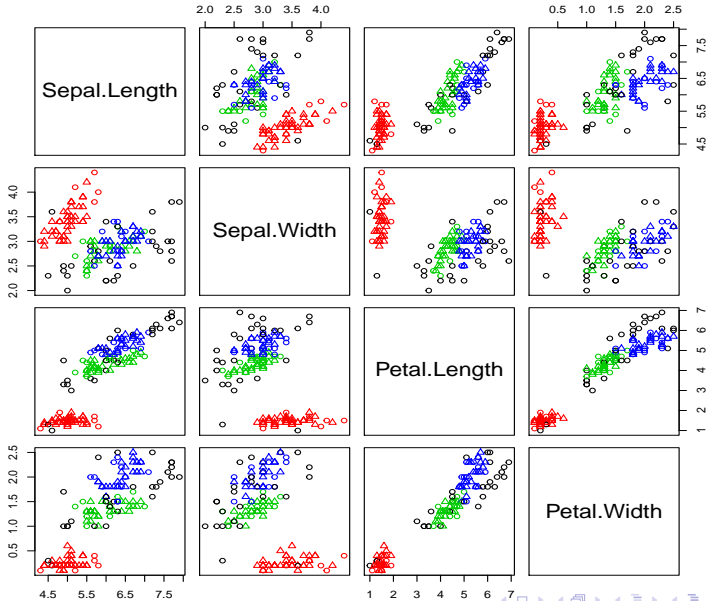
- ▶ Group objects into one cluster if they are connected to one another by densely populated area
- ▶ The DBSCAN algorithm from package *fpc* provides a density-based clustering for numeric data.
- ▶ Two key parameters in DBSCAN:
 - ▶ `eps`: reachability distance, which defines the size of neighborhood; and
 - ▶ `MinPts`: minimum number of points.
- ▶ If the number of points in the neighborhood of point α is no less than `MinPts`, then α is a *dense point*. All the points in its neighborhood are *density-reachable* from α and are put into the same cluster as α .
- ▶ Can discover clusters with various shapes and sizes
- ▶ Insensitive to noise

Density-based Clustering of the iris data

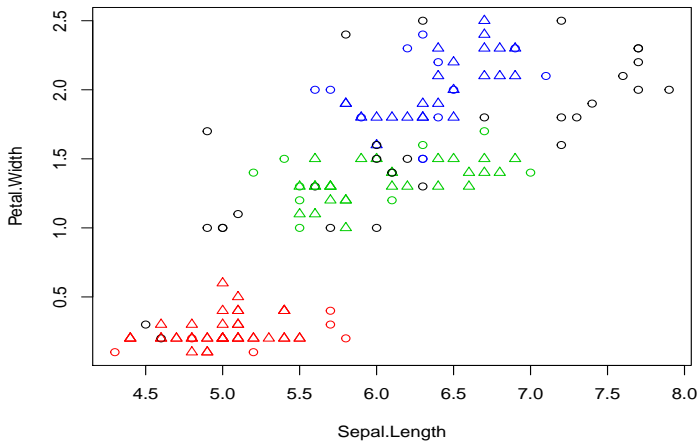
```
library(fpc)
iris2 <- iris[-5] # remove class tags
ds <- dbscan(iris2, eps = 0.42, MinPts = 5)
ds
## dbscan Pts=150 MinPts=5 eps=0.42
##          0  1  2  3
## border 29  6 10 12
## seed    0 42 27 24
## total  29 48 37 36
```

```
# compare clusters with actual class labels
table(ds$cluster, iris$Species)
##
##      setosa versicolor virginica
## 0         2          10         17
## 1        48           0           0
## 2         0          37           0
## 3         0           3          33
```

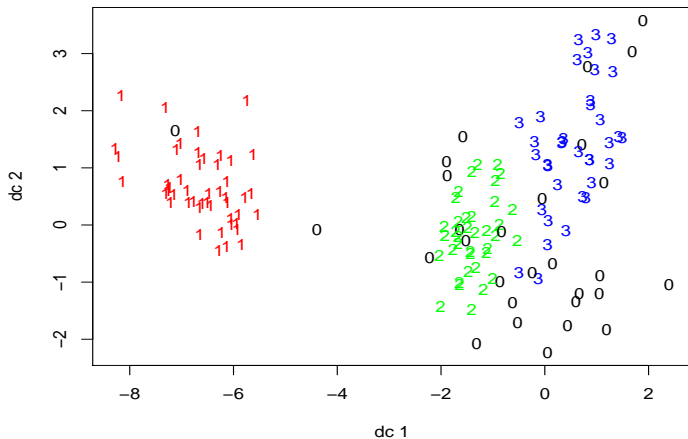
- ▶ 1 to 3: identified clusters
- ▶ 0: noises or outliers, i.e., objects that are not assigned to any clusters



```
plot(ds, iris2[, c(1, 4)])
```



```
plotcluster(iris2, ds$cluster)
```



- ▶ Label new data, based on their similarity with the clusters
- ▶ Draw a sample of 10 objects from `iris` and add small noises to them to make a new dataset for labeling
- ▶ Random noises are generated with a uniform distribution using function `runif()`.

```
# create a new dataset for labeling
set.seed(435)
idx <- sample(1:nrow(iris), 10)
# remove class labels
new.data <- iris[idx,-5]
# add random noise
new.data <- new.data + matrix(runif(10*4, min=0, max=0.2),
                             nrow=10, ncol=4)

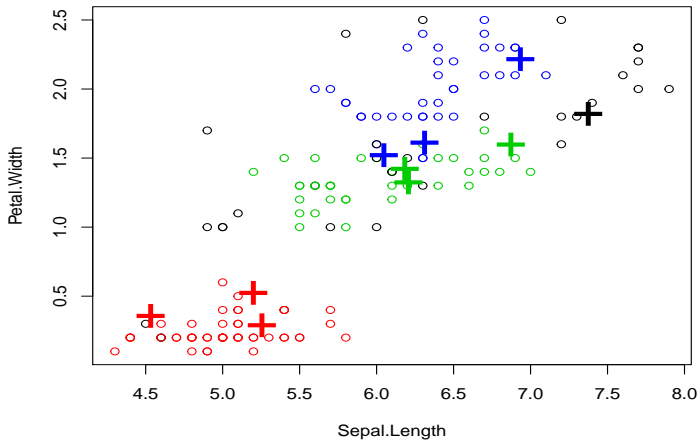
# label new data
pred <- predict(ds, iris2, new.data)
```

```
table(pred, iris$Species[idx]) # check cluster labels
##
## pred setosa versicolor virginica
##    0     0         0         1
##    1     3         0         0
##    2     0         3         0
##    3     0         1         2
```

```
table(pred, iris$Species[idx]) # check cluster labels
##
## pred setosa versicolor virginica
##    0     0         0         1
##    1     3         0         0
##    2     0         3         0
##    3     0         1         2
```

Eight(=3+3+2) out of 10 objects are assigned with correct class labels.


```
plot(iris2[, c(1, 4)], col = 1 + ds$cluster)
points(new.data[, c(1, 4)], pch = "+", col = 1 + pred, cex = 3,
```



Introduction

- Data Clustering with R
- The Iris Dataset

Partitioning Clustering

- The k -Means Clustering
- The k -Medoids Clustering

Hierarchical Clustering

Density-Based clustering

Cluster Validation

Further Readings and Online Resources

- ▶ `silhouette()` compute or extract silhouette information (*cluster*)
- ▶ `cluster.stats()` compute several cluster validity statistics from a clustering and a dissimilarity matrix (*fpc*)
- ▶ `clValid()` calculate validation measures for a given set of clustering algorithms and number of clusters (*clValid*)
- ▶ `clustIndex()` calculate the values of several clustering indexes, which can be independently used to determine the number of clusters existing in a data set (*cclust*)
- ▶ `NbClust()` provide 30 indices for cluster validation and determining the number of clusters (*NbClust*)

Introduction

- Data Clustering with R
- The Iris Dataset

Partitioning Clustering

- The k -Means Clustering
- The k -Medoids Clustering

Hierarchical Clustering

Density-Based clustering

Cluster Validation

Further Readings and Online Resources

- ▶ A brief overview of various approaches for clustering
Yanchang Zhao, et al. "Data Clustering." In Ferragline et al. (Eds.), Handbook of Research on Innovations in Database Technologies and Applications, Feb 2009. <http://yanchang.rdatamining.com/publications/Overview-of-Data-Clustering.pdf>
- ▶ Cluster Analysis & Evaluation Measures
https://en.wikipedia.org/wiki/Cluster_analysis
- ▶ Detailed review of algorithms for data clustering
Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM Computing Surveys, 31(3), 264-323.
Berkhin, P. (2002). Survey of Clustering Data Mining Techniques. Accrue Software, San Jose, CA, USA.
<http://citeseer.ist.psu.edu/berkhin02survey.html>.
- ▶ A comprehensive textbook on data mining
Han, J., & Kamber, M. (2000). Data mining: concepts and techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

- ▶ Data Mining Algorithms In R: Clustering

https:

[//en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering](https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering)

- ▶ Data Mining Algorithms In R: k-Means Clustering

https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/K-Means

- ▶ Data Mining Algorithms In R: k-Medoids Clustering

[https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Partitioning_Around_Medoids_\(PAM\)](https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Partitioning_Around_Medoids_(PAM))

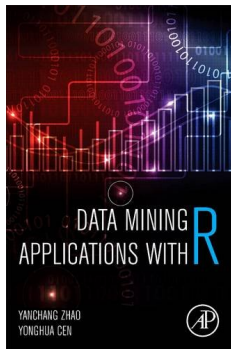
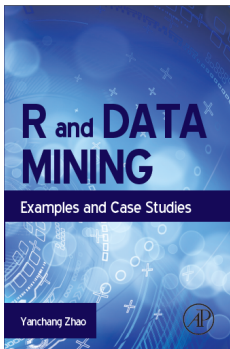
- ▶ Data Mining Algorithms In R: Hierarchical Clustering

https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Hierarchical_Clustering

- ▶ Data Mining Algorithms In R: Density-Based Clustering

https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Density-Based_Clustering

- ▶ Chapter 6 - Clustering, in book
R and Data Mining: Examples and Case Studies
<http://www.rdatamining.com/docs/RDataMining-book.pdf>
- ▶ RDataMining Reference Card
<http://www.rdatamining.com/docs/RDataMining-reference-card.pdf>
- ▶ Free online courses and documents
<http://www.rdatamining.com/resources/>
- ▶ RDataMining Group on LinkedIn (26,000+ members)
<http://group.rdatamining.com>
- ▶ Twitter (3,300+ followers)
@RDataMining



Thanks!

Email: [yanchang\(at\)RDataMining.com](mailto:yanchang(at)RDataMining.com)

Twitter: @RDataMining



Alsabti, K., Ranka, S., and Singh, V. (1998).

An efficient k-means clustering algorithm.

In *Proc. the First Workshop on High Performance Data Mining*, Orlando, Florida.



Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999).

OPTICS: ordering points to identify the clustering structure.

In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 49–60, New York, NY, USA. ACM Press.



Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996).

A density-based algorithm for discovering clusters in large spatial databases with noise.

In *KDD*, pages 226–231.



Frank, A. and Asuncion, A. (2010).

UCI machine learning repository. university of california, irvine, school of information and computer sciences.

<http://archive.ics.uci.edu/ml>.



Guha, S., Rastogi, R., and Shim, K. (1998).

CURE: an efficient clustering algorithm for large databases.

In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 73–84, New York, NY, USA. ACM Press.



Guha, S., Rastogi, R., and Shim, K. (1999).

ROCK: A robust clustering algorithm for categorical attributes.

In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Australia*, pages 512–521. IEEE Computer Society.



Han, J. and Kamber, M. (2000).

Data Mining: Concepts and Techniques.

Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.



Hennig, C. (2014).

fpc: Flexible procedures for clustering.
R package version 2.1-9.



Hinneburg, A. and Keim, D. A. (1998).

An efficient approach to clustering in large multimedia databases with noise.
In *KDD*, pages 58–65.
DENCLUE.



Huang, Z. (1998).

Extensions to the k-means algorithm for clustering large data sets with categorical values.
Data Min. Knowl. Discov., 2(3):283–304.



Karypis, G., Han, E.-H., and Kumar, V. (1999).

Chameleon: hierarchical clustering using dynamic modeling.
Computer, 32(8):68–75.



Kaufman, L. and Rousseeuw, P. J. (1990).

Finding groups in data. an introduction to cluster analysis.
Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York:
Wiley, 1990.



Macqueen, J. B. (1967).

Some methods of classification and analysis of multivariate observations.
In *the Fifth Berkeley Symposium on Mathematical Statistics and Probability.*



Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2016).

cluster: Cluster Analysis Basics and Extensions.
R package version 2.0.4 — For new features, see the 'Changelog' file (in the package source).



Ng, R. T. and Han, J. (1994).

Efficient and effective clustering methods for spatial data mining.

In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 144–155, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.



R Core Team (2015).

R: A Language and Environment for Statistical Computing.

R Foundation for Statistical Computing, Vienna, Austria.



Zhang, T., Ramakrishnan, R., and Livny, M. (1996).

BIRCH: an efficient data clustering method for very large databases.

In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 103–114, New York, NY, USA. ACM Press.