FIT3152 Data analytics Lecture 10

Cluster analysis

- Supervised vs Unsupervised learning
- k-Means Clustering
- Hierarchical Clustering
- Cluster analysis in R

Week-by-week

Week Starting	Lecture	Topic	Tutorial	A1	A2
28/2/22	1	Intro to Data Science, review of basic statistics using R	...		
7/3/22	2	Exploring data using graphics in R	T1		
14/3/22	3	Data manipulation in R	T2	Released	
21/3/22	4	Data Science methodologies, dirty/clean/tidy data, data manipulation	T3		
28/3/22	5	Network analysis	T4		
4/4/22	6	Regression modelling	T5		
11/4/22	7	Classification using decision trees	T6		
		Mid-semester Break		Submitted	
25/4/22	8	Naïve Bayes, evaluating classifiers	T7		Released
2/5/22	9	Ensemble methods, artificial neural networks	T8		
9/5/22	10	Clustering	T9		
16/5/22	11	Text analysis	T10		Submitted
23/5/22	12	Review of course, Exam preparation	T11		

SETU

Student Evaluation of Teaching and Units (SETU) has opened for Semester 1.

- All students are encouraged to participate. Your feedback is very important.
- You will see a block in Moodle linking you to the survey.

End of semester exam

The end of semester exam:

- Will be online. The university is yet to make a formal announcement, but I expect on campus students will sit their exam at Monash. The university will advise you of the arrangements for sitting the exam.
- The exam is closed book. You are allowed two sheets of blank A4 paper for working.
- You may use a calculator: graphing, scientific, or CAS.
- A mock (practice) exam has been setup. There will be a link under the Assessments tile in Moodle from the end of Week 10. It is a good indicator of length and complexity.
- Solutions will be released in Week 12.

Assignment 2

FIT3152 Data analytics – 2022: Assignment 2

Your task	• The objective of this assignment is to gain familiarity with classification models using R. • This is an individual assignment.
Value	• This assignment is worth **20%** of your total marks for the unit. • It has 20 marks in total.
Suggested Length	• 4 – 6 A4 pages (for your report) + extra pages as appendix (for your code) • Font size 11 or 12pt, single spacing
Due Date	**11.55pm Friday 20th May 2022**
Submission	• PDF file only. Naming convention: *FirstnameSecondnameID.pdf* • Via Moodle Assignment Submission. • Turnitin will be used for similarity checking of all submissions.
Late Penalties	• 10% (2 mark) deduction per calendar day for up to one week. • Submissions more than 7 calendar days after the due date will receive a mark of zero (0) and no assessment feedback will be provided.

Assignment 2

Instructions and data

The objective of this assignment is to gain familiarity with classification models using R.
We want to obtain a model that may be used to predict whether tomorrow will be warmer than today for 10 locations in Australia.

You will be using a modified version of the Kaggle competition data: Predict rain tomorrow in Australia. https://www.kaggle.com/jsphyg/weather-dataset-rattle-package The data contains meteorological observations as attributes, and the class attribute "Warmer Tomorrow".

There are two options for compiling your report:
(1) You can submit a single pdf with R code pasted in as machine-readable text as an appendix, or
(2) As an R Markup document that contains the R code with the discussion/text interleaved. Render this as an HTML file and print off as a pdf and submit.

Regardless of which method you choose, you will submit a single pdf, and your R code will be machine readable text. We need to conform to this format as the university now requires all student submission to be processed by plagiarism detection software.

Submit your report as a single PDF with the file name ***FirstnameSecondnameID.pdf*** on Moodle.

Assignment 2

Creating your data set

Clear your workspace, set the number of significant digits to a sensible value, and use '**WAUS**' as the default data frame name for the whole data set. Read your data into R and create your individual data using the following code:

```
rm(list = ls())
WAUS <- read.csv("WarmerTomorrow2022.csv")
L <- as.data.frame(c(1:49))
set.seed(XXXXXXXX) # Your Student ID is the random seed
L <- L[sample(nrow(L), 10, replace = FALSE),] # sample 10 locations
WAUS <- WAUS[(WAUS$Location %in% L),]
WAUS <- WAUS[sample(nrow(WAUS), 2000, replace = FALSE),] # sample 2000 rows
```

Hint: code does not automatically convert strings to factors etc.

Assignment 2

Questions

1. Explore the data: What is the proportion of days when it is warmer than the previous day compared to those where it is cooler? Obtain descriptions of the predictor (independent) variables – mean, standard deviations, etc. for real-valued attributes. Is there anything noteworthy in the data? Are there any attributes you need to consider omitting from your analysis? **(1 Mark)**

2. Document any pre-processing required to make the data set suitable for the model fitting that follows. **(1 Mark)**

3. Divide your data into a 70% training and 30% test set by adapting the following code (written for the iris data). Use your student ID as the random seed.

```
set.seed(XXXXXXXX) #Student ID as random seed
train.row = sample(1:nrow(iris), 0.7*nrow(iris))
iris.train = iris[train.row,]
iris.test = iris[-train.row,]
```

Assignment 2

4. Implement a classification model using each of the following techniques. For this question you may use each of the R functions at their default settings if suitable. **(5 Marks)**

- Decision Tree
- Naïve Bayes
- Bagging
- Boosting
- Random Forest

5. Using the test data, classify each of the test cases as 'warmer tomorrow' or 'not warmer tomorrow'. Create a confusion matrix and report the accuracy of each model. **(1 Mark)**

6. Using the test data, calculate the confidence of predicting 'warmer tomorrow' for each case and construct an ROC curve for each classifier. You should be able to plot all the curves on the same axis. Use a different colour for each classifier. Calculate the AUC for each classifier. **(1 Mark)**

Assignment 2

7. Create a table comparing the results in parts 5 and 6 for all classifiers. Is there a single "best" classifier? **(1 Mark)**

8. Examining each of the models, determine the most important variables in predicting whether it will be warmer tomorrow or not. Which variables could be omitted from the data with very little effect on performance? Give reasons. **(2 Marks)**

9. Starting with one of the classifiers you created in Part 4, create a classifier that is simple enough for a person to be able to classify whether it will be warmer tomorrow or not by hand. Describe your model, either with a diagram or written explanation. How well does your model perform, and how does it compare to those in Part 4? What factors were important in your decision? State why you chose the attributes you used. (**2 Marks**)

Assignment 2

10. Create the best tree-based classifier you can. You may do this by adjusting the parameters, and/or cross-validation of the basic models in Part 4 or using an alternative tree-based learning algorithm. Show that your model is better than the others using appropriate measures. Describe how you created your improved model, and why you chose that model. What factors were important in your decision? State why you chose the attributes you used. **(3 Marks)**

11. Using the insights from your analysis so far, implement an Artificial Neural Network classifier and report its performance. Comment on attributes used and your data pre-processing required. How does this classifier compare with the others? Can you give any reasons? **(2 Marks)**

12. Write a brief report (suggested length 6 pages) summarizing your results in parts 1 – 11. Use commenting in your R script, where appropriate, to help a reader understand your code. Alternatively combine working, comments and reporting in R Markdown. **(1 Mark)**

Assignment 2

Description of the data

Attributes 1-3, Day, Month, Year	Day, Month, Year of the observation.
Attribute 4, Location	The location of the observation.
Attribute 5, MinTemp	The daily minimum temperature in degrees Celsius.
Attribute 6, MaxTemp	The daily maximum temperature in degrees Celsius.
Attribute 7, Rainfall	Rainfall recorded for the day in mm.
Attribute 8, Evaporation	The evaporation (mm) in the 24 hours to 9am.
Attribute 9, Sunshine	Hours of bright sunshine over the day.
Attribute 10, WindGustDir	Direction of strongest wind gust over the day.

Assignment 2

Attribute 11, WindGustSpeed	Speed (km/h) of the strongest wind gust over the day.
Attribute 12, WindDir9am	Direction of the wind at 9am.
Attribute 13, WindDir3pm	Direction of the wind at 3pm.
Attribute 14, WindSpeed9am	Speed (km/hr) averaged over 10 minutes prior to 9am.
Attribute 15, WindSpeed3pm	Speed (km/hr) averaged over 10 minutes prior to 3pm.
Attribute 16, Humidity9am	Humidity (percent) at 9am.
Attribute 17, Humidity3pm	Humidity (percent) at 3pm.
Attribute 18, Pressure9am	Atmospheric pressure (hpa) reduced to mean sea level at 9am.
Attribute 19, Pressure3pm	Atmospheric pressure (hpa) reduced to mean sea level at 3pm.

Assignment 2

Attribute 20, Cloud9am	Fraction of sky obscured by cloud at 9am. This is measured in "oktas", which are a unit of eigths. It records how many eigths of the sky are obscured by cloud. A 0 measure indicates completely clear sky whilst an 8 indicates that it is completely overcast.
Attribute 21, Cloud3pm	Fraction of sky obscured by cloud at 3pm.
Attribute 22, Temp9am	Temperature (degrees C) at 9am.
Attribute 23, Temp3pm	Temperature (degrees C) at 3pm.
Attribute 24, WarmerTomorrow	The target variable. Will tomorrow be warmer than today?

Quick revision from last week:

Question 1

A Bagging algorithm produces a classification by taking a/the _____ of the tree classifiers?

A. Average
B. Median
C. Vote
D. Minimum

Question 2

The Boosting algorithm is designed to improve the Bagging algorithm by building decision trees:

A. Using the original sample?
B. Using the original sample, weighted instances?
C. Weighted sampling with replacement?
D. Weighted sampling without replacement?

Question 3

The Random Forests algorithm is designed to improve the Bagging algorithm by:

A. Varying the size of trees?
B. Varying the number of trees?
C. Varying the attributes used to build trees?
D. Varying the depth of trees?

Question 4

To fit the Iris data, an Artificial Neural Network (ANN) requires:

A. 4 Input Nodes and 1 Output Node?
B. 4 Input Nodes and 2 Output Nodes?
C. 4 Input Nodes and 3 Output Nodes?
D. 4 Input Nodes and 4 Output Nodes?

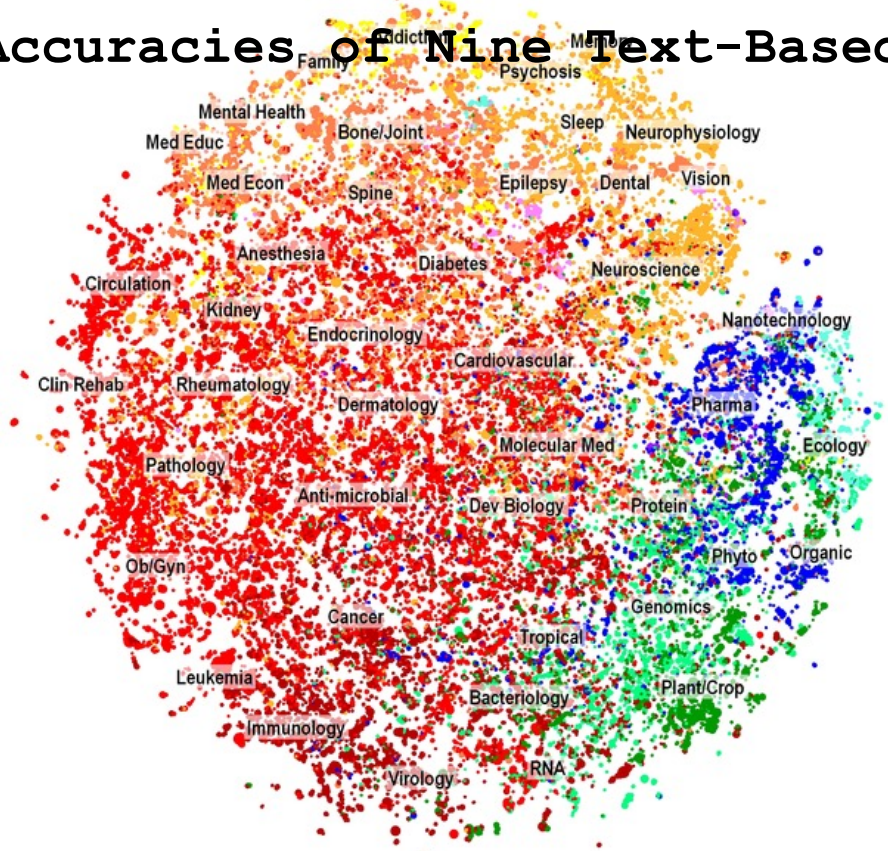
Cluster analysis

Food groups

https://www.muralsyourway.com/p/food-groups-mural/

Document clustering

Clustering More than Two Million Biomedical Publications: Comparing the Accuracies of Nine Text-Based Similarity Approaches

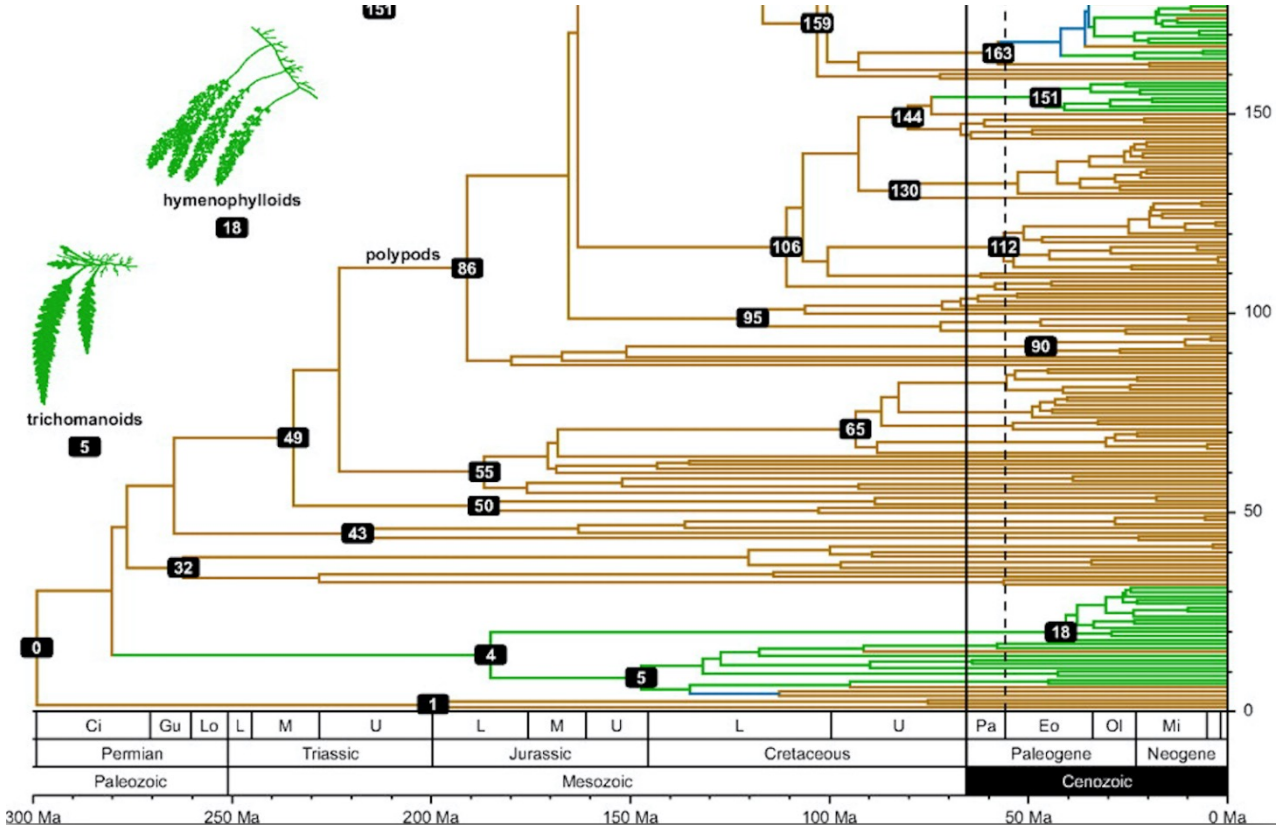


https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0018029

7 Australian political personas

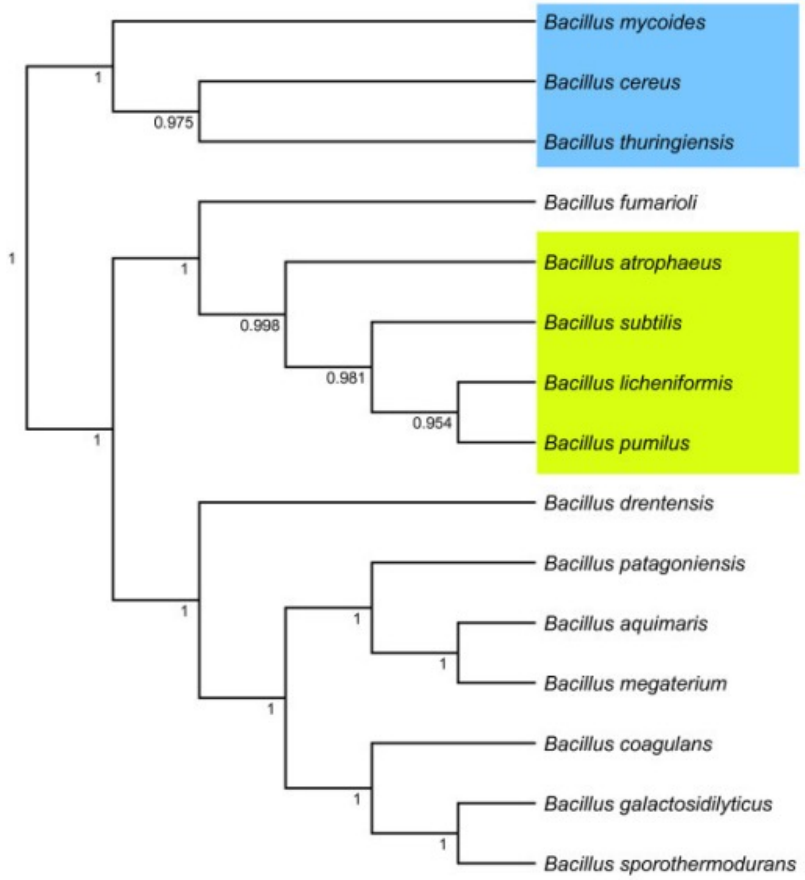
https://www.smh.com.au/

Phylogenetic tree, fern evolution



https://www.pnas.org/content/106/27/11200/F1.expansion.html

Phylogenetic tree, Bacillus species



https://openi.nlm.nih.gov/detailedresult.php?img=PMC2828439_1471-2105-11-69-1&req=4

COVID-19

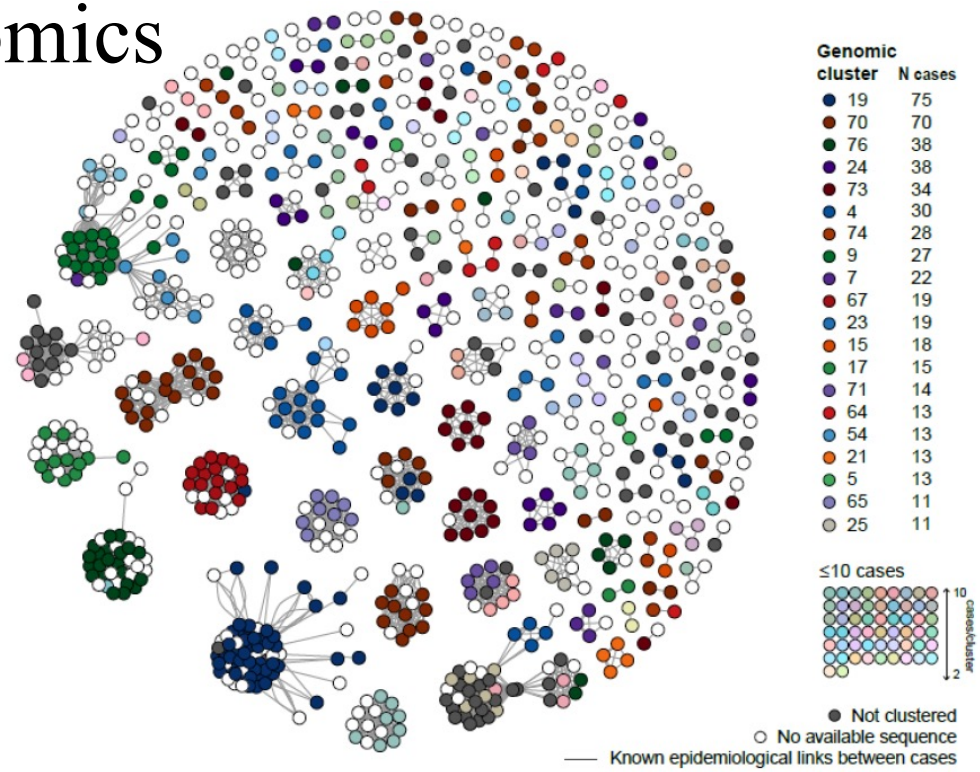
Tracking the COVID-19 pandemic in Australia using genomics

> **Sequenced samples from Australia were representative of the global diversity of SARS-CoV-2, ... In total, 76 distinct genomic clusters were identified; these included large clusters associated with social venues, healthcare facilities and cruise ships. Sequencing of sequential samples from 98 patients revealed minimal intra-patient SARS-CoV-2 genomic diversity.**

https://www.medrxiv.org/content/10.1101/2020.05.12.20099929v1

COVID-19

Tracking the COVID-19 pandemic in Australia using genomics



https://www.medrxiv.org/content/10.1101/2020.05.12.20099929v1

SARS-CoV-2 antigenic variants

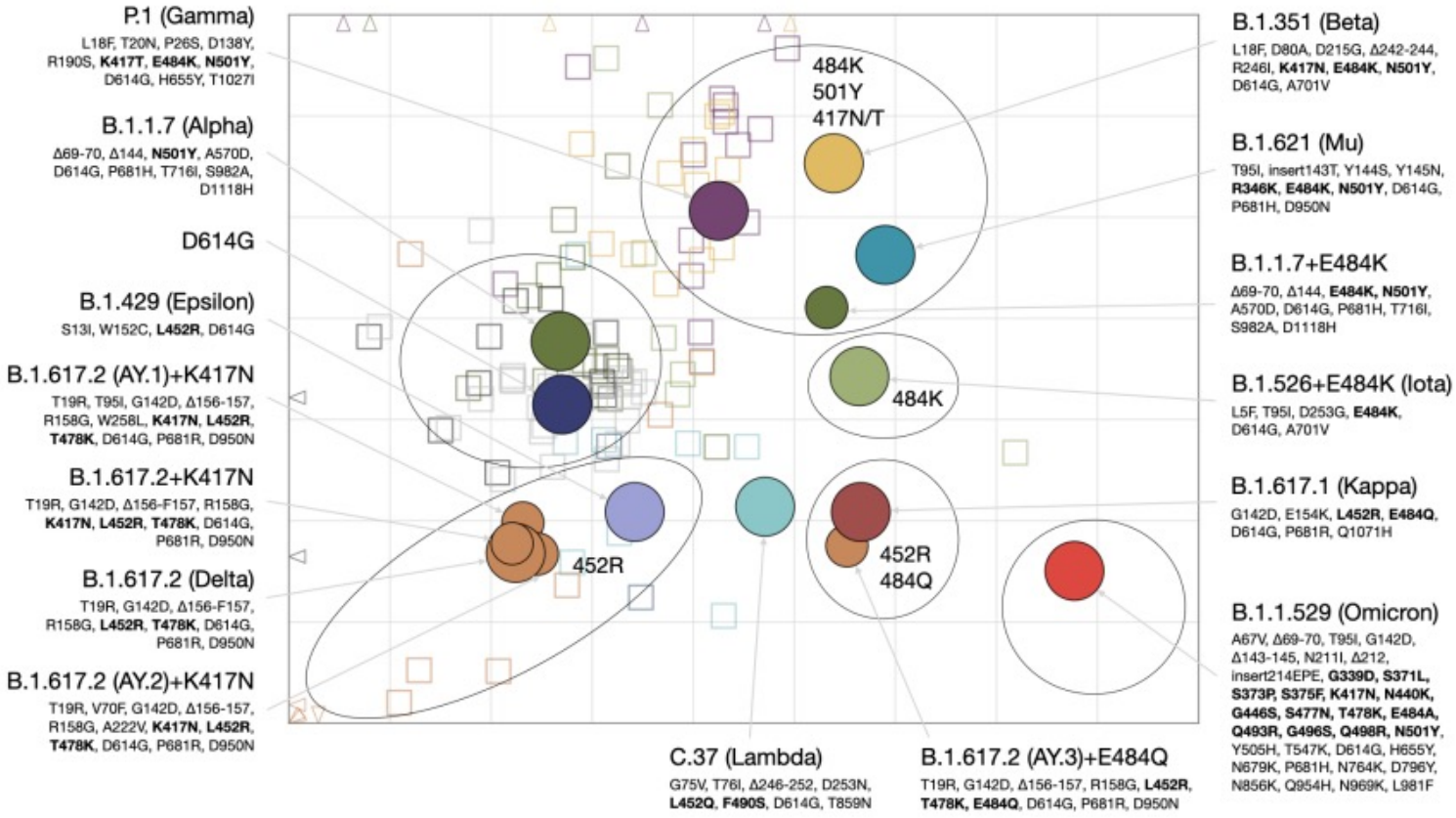


Fig. 2: Antigenic map of SARS-CoV-2 variants and selected substitutions. Variants are shown as circles, sera as squares.

https://www.biorxiv.org/content/10.1101/2022.01.28.477987v1.full.pdf

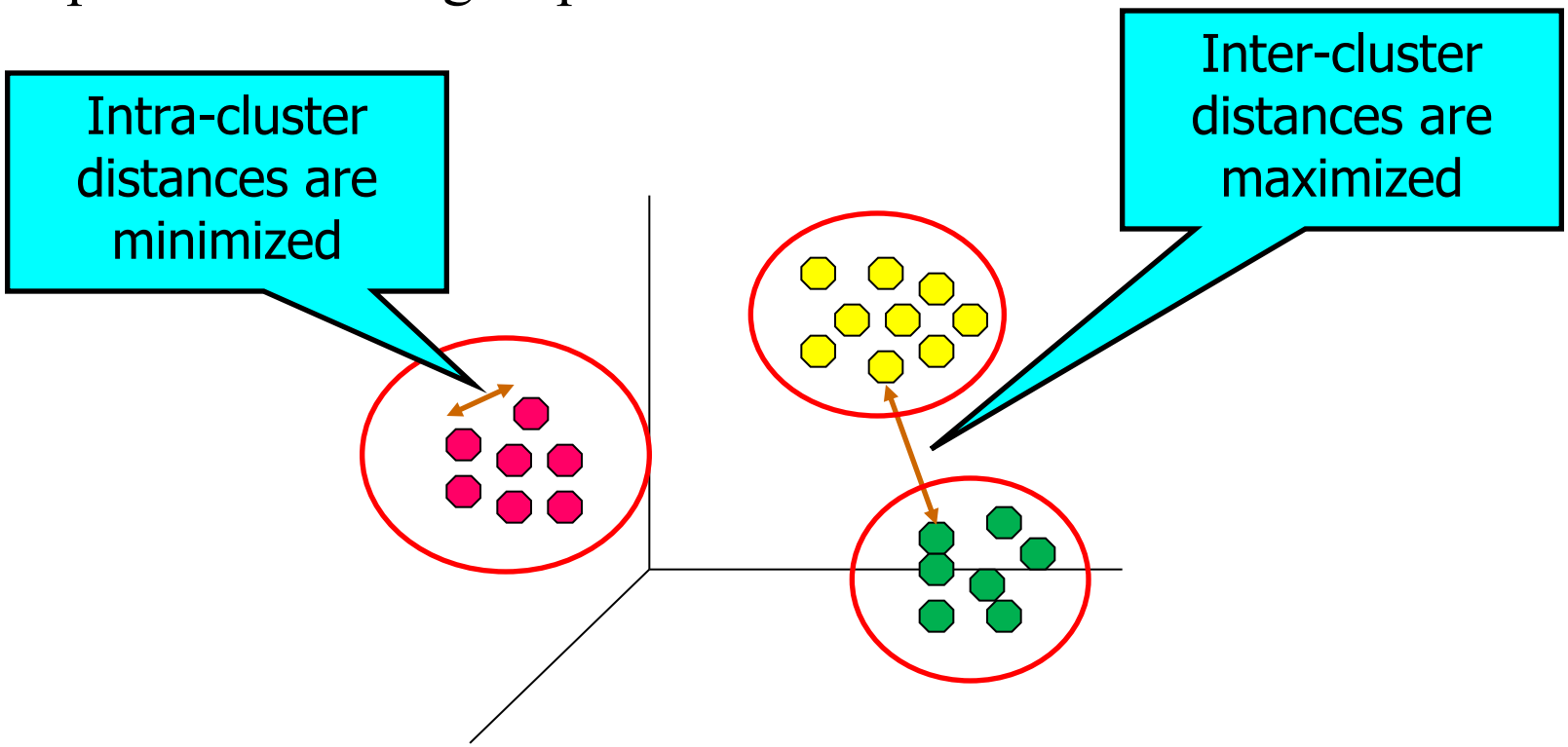
Supervised *vs* unsupervised learning

There are two main approaches to machine learning:

- Supervised learning algorithms:
  - > Algorithms are given labelled examples (target class) for the various types of data that need to be learned.
  - > For example: classification algorithms such as decision trees, artificial neural networks, Bayesian classifiers.
- Unsupervised learning algorithms:
  - > Data is unlabeled (has no predefined classes) and the learning algorithms attempt to find patterns within the data to put into groups or sets.
  - > For example, clustering algorithms.

What is Cluster Analysis?

Finding groups of points such that the points in a group will be similar (or related) to one another and different from (or unrelated to) the points in other groups



Clustering – applications

Examples:

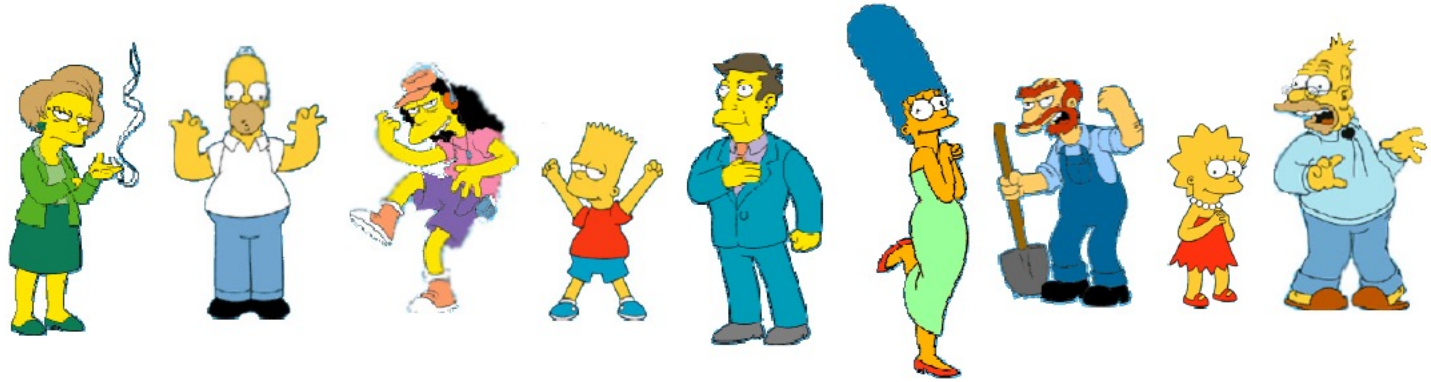
- Segment customer database based on similar buying patterns.
- Group houses in a town into neighborhoods based on similar features.
- Identify similar Internet usage patterns.
- Clustering articles/emails by content area.
- Gene clustering in biology.
- Group together documents/web sites that have similar content.

Are these clusters pre-defined?

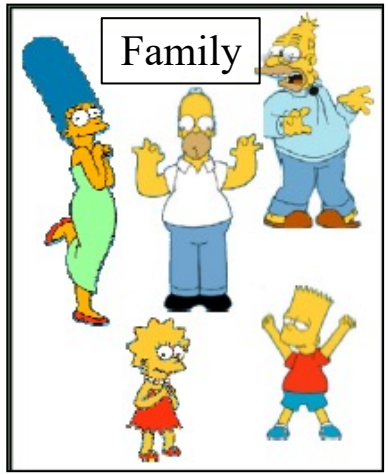
- No, it depends on the data and what you want to do with it.
- There are no class labels.

Illustrating clustering

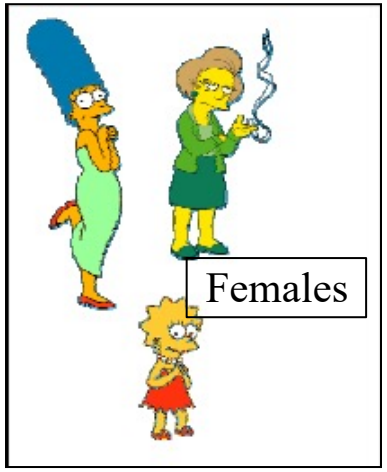
Are there natural groupings amongst this group?

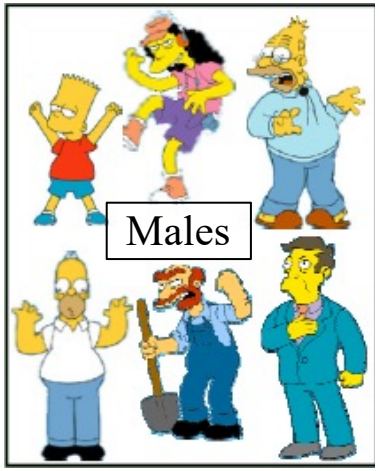


Possible clusters



or





Clustering Definition

- Clustering identifies natural groups in a data set:
  - > Given a set of data points, each having a set of attributes, and a similarity measure, find clusters such that:
  - > Data points in one cluster are more similar to one another.
  - > Data points in separate clusters are less similar to one another.
- Similarity Measures:
  - > Euclidean Distance (think Pythagoras' theorem).
  - > Other distance-based measures (for example, Manhattan).
  - > (Other measures if the attribute values are not continuous, for example cosine distance for text (*next week's lecture*)).

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

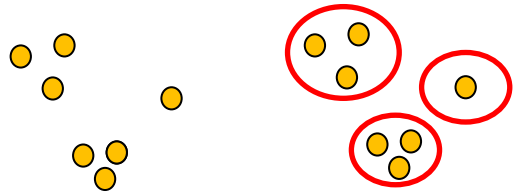
Two Clusters

Four Clusters

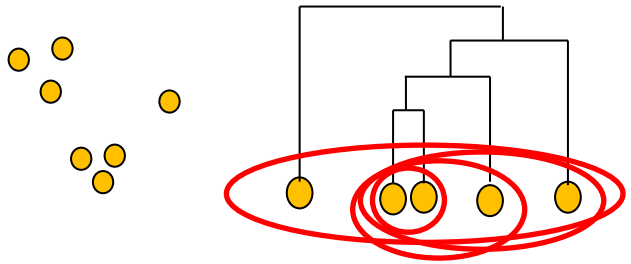
Types of clustering

Two main approaches: partitional and hierarchical.

- Partitional: the division of data points into non-overlapping subsets (clusters) such that each data point is in exactly one subset.



- Hierarchical: a set of nested clusters organized as a hierarchical tree.



k-Means clustering

k-Means Clustering

Partitional clustering approach

Each cluster is associated with a **centroid** (center point)

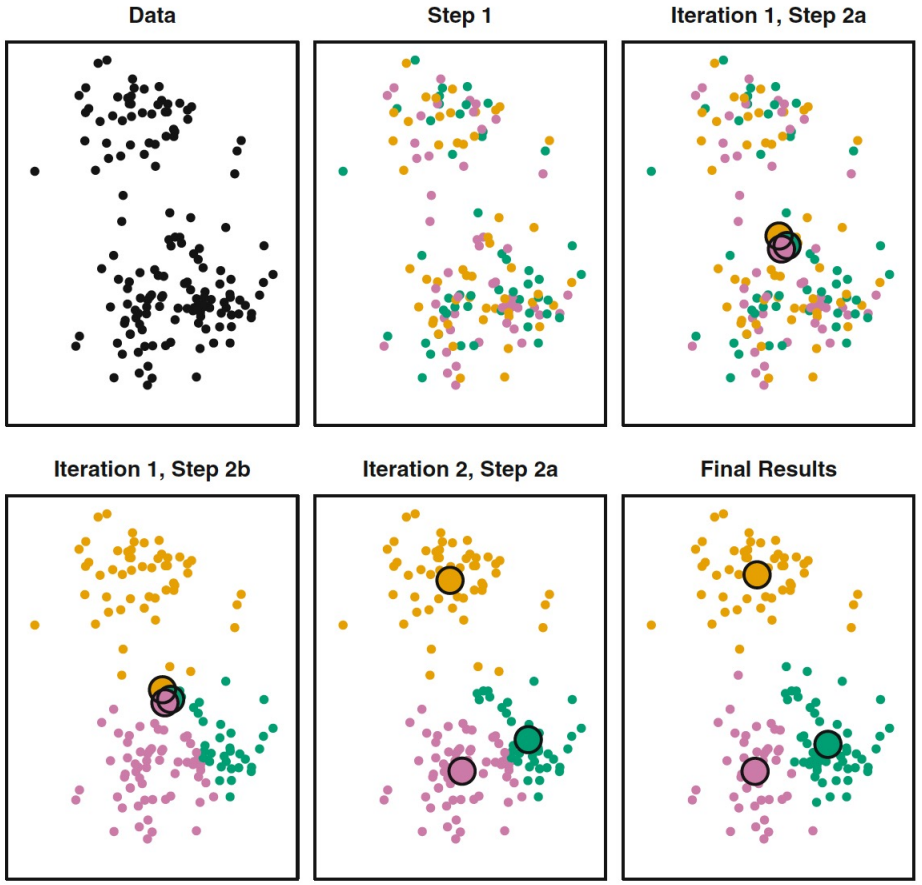
Each point is assigned to the cluster with the closest centroid

Number of clusters, *k*, must be specified

The basic algorithm is very simple:

1. Select *k* points (at random) as the initial centroids
2. **Repeat**
    3. Form *k* clusters by assigning all points to the closest centroid
    4. Re-compute the centroid of each cluster
5. **Until** the centroids don't change

k-Means demonstration



In Step 1 each point is randomly assigned to a cluster. Top right: in Step 2(a), the cluster centroids are computed. In Step 2(b), each point is assigned to the nearest centroid. Bottom center: Step 2(a) is once again performed, leading to new cluster centroids. Bottom right: the results obtained after ten iterations.

James et al., An Introduction to Statistical Learning

Finding the centroids

How do we decide which is the **closest centroid**?

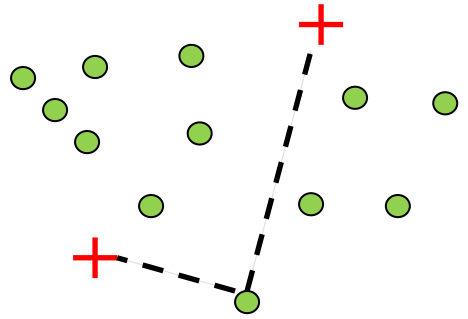
We need to find the ‘distance’ between each point and all the centroids

What does ‘distance’ mean?

There are many ways of defining ‘distance’. We need to use a distance metric.

Data points

Centroids

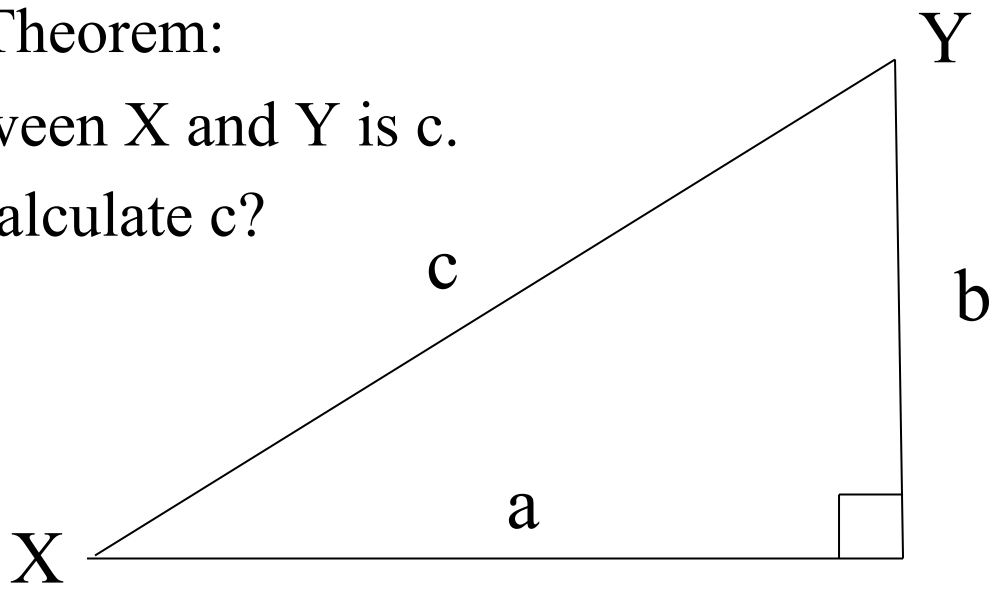


What is Euclidean distance?

Pythagoras' Theorem:

Distance between X and Y is c.

How do we calculate c?



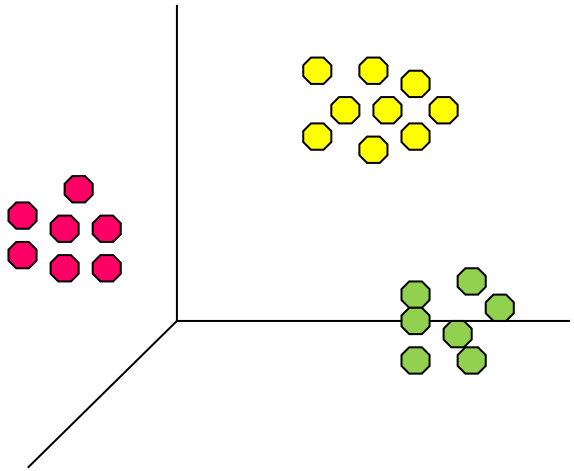
$$c^2 = a^2 + b^2 \quad therefore \quad c = \sqrt{(a^2 + b^2)}$$

This model can be applied to multiple dimensions!

What is k-Means aiming to do

We can now specify the objective of the k-Means algorithm in terms of the distance metric. We are trying to minimise the total squared distance of each point to its centroid:

$\sum_{i=1}^{k} \sum_{j=1}^{n} d\left(c_i, x_{i,j}\right)^2$ where:



- k is the number of clusters
- c_i is the centroid of each cluster for i=1,….k
- n_i is the number in cluster i
- $x_{i,j}$ is the jth point of cluster i
- $d(c_i, x_{i,j})$ is the distance between c_i and $x_{i,j}$.

Evaluating k-Means Clusters

Most common measure is Sum of Squared Error (SSE)

- For each point, the error is the distance to the nearest cluster.
- To get SSE, we square these errors and sum them.
- x_i is a data point in cluster C_i and c_i is the centroid of cluster C_i .
- From previous slide: SSE = $\sum_{i=1}^{k} \sum_{j=1}^{n} d\left(c_i, x_{i,j}\right)^2$
- Given two sets of clusters, we can choose the one with the smallest error.
- One easy way to reduce SSE is to increase *k*, the number of clusters.
- Note: A good clustering with smaller *k* can have a lower SSE than a poor clustering with higher *k*.

Normalising attributes

It is a good idea to normalise the data before clustering, otherwise large valued attributes will exert greater influence on the clustering.

This is achieved by rescaling each attribute to fit within the same range (for example, between 0 and 1). To normalize attribute A:

MaxA and *MinA* are the maximum and minimum of *A*. Then, the normalized values of *A* are: $x_{new}=\frac{x-MinA}{MaxA-MinA}$

R software has a function (scale) which performs a similar – but not identical function.

Pre-processing and post-processing

Pre-processing

- Normalise the data
- Eliminate outliers

Post-processing

- Eliminate small clusters that may represent outliers
- Split ‘loose’ clusters, i.e., clusters with relatively high SSE.
- Merge clusters that are ‘close’ and that have relatively low SSE.

k-Means clustering in R

The k-Means function is built into the Stats package, which is loaded by default.

Using the iris data:

```
> set.seed(9999)
> data("iris")
```

k-Means clustering in R

Using sepals (Cols 1 & 2), create 3 clusters, choose the best out of 20 starting configurations.

```
> ikfit = kmeans(iris[,1:2], 3, nstart = 20)
> ikfit
> table(actual = iris$Species, fitted = ikfit$cluster)
```

```
            fitted
actual        1  2  3
  setosa      0 50  0
  versicolor 12  0 38
  virginica  35  0 15
```

k-Means clustering in R

Looking at the ikfit object:

> ikfit

```
K-means clustering with 3 clusters of sizes 47,
50, 53

Cluster means:  Sepal.Length Sepal.Width
          1     6.812766    3.074468
          2     5.006000    3.428000
          3     5.773585    2.692453

Clustering vector:  [1] 2 2 2 2 2 2 ...
```

k-Means clustering in R

Looking at the ikfit object:

```
...
Within cluster sum of squares by cluster:
[1] 12.6217 13.1290 11.3000
(between_SS / total_SS =  71.6 %)

Available components:
[1] "cluster"      "centers"      "totss"
[4] "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

k-Means clustering in R

Looking at the sums of squares calculations:

> ikfit$totss Total SS from a single centroid (treats data as one cluster).

```
[1] 130.4753
```

> ikfit$withinss SS within each cluster.

```
[1] 12.6217 13.1290 11.3000
```

> ikfit$tot.withinss Total within clusters.

```
[1] 37.0507
```

> ikfit$betweenss Total sum of squares - total SS within clusters.

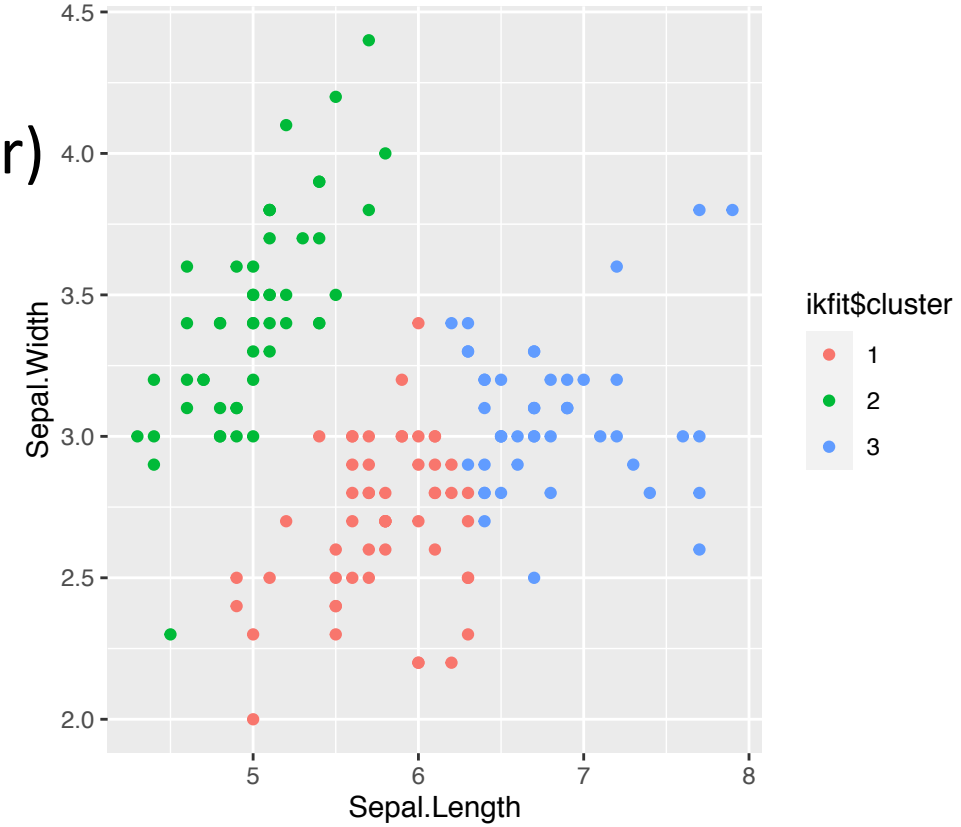
```
[1] 93.42456
```

k-Means clustering in R

Plotting the clusters:

> ikfit$cluster = as.factor(ikfit$cluster)

> ggplot(iris, aes(Sepal.Length, Sepal.Width, color = ikfit$cluster)) + geom_point()



? kmeans

- Description

  Perform k-means clustering on a data matrix.

- Usage

```
kmeans(x, centers, iter.max = 10, nstart = 1,
algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
"MacQueen"), trace=FALSE)
x           data
centers     number of clusters (k)
nstart      random starting positions to test
iter.max    maximum number of iterations
...
```

k-Means clustering in R

Note that using both petals and sepals improves the accuracy of the clustering for these data:

```
> ikfit = kmeans(iris[,1:4], 3, nstart = 20)
> ttable(actual = iris$Species, fitted = ikfit$cluster)
             fitted
actual         1  2  3
  setosa       0  0 50
  versicolor   2 48  0
  virginica   36 14  0
> ikfit$tot.withinss
[1] 78.85144
```

k-Means for classification…

From previous slide, using both petals and sepals for the clustering:

> ttable(actual = iris$Species, fitted = ikfit$cluster)

```
             fitted
actual         1  2  3
  setosa       0  0 50
  versicolor   2 48  0
  virginica   36 14  0
```

- If we classify Setosa = Group 3, Versicolor = Group 2 and Virginica = Group 1, this has an accuracy of:

> (50 + 48 + 36)/150 = 0.89.

k-Means clustering in R

But the number of clusters is arbitrary, for example:

> ikfit = kmeans(iris[,1:4], 5, nstart = 20)

> ttable(actual = iris$Species, fitted = ikfit$cluster)

```
                Fitted
actual           1  2  3  4  5
  setosa         0  0  0  0 50
  versicolor    26  0 24  0  0
  virginica     13 24  1 12  0
```

> ikfit$tot.withinss

```
[1] 46.44618
```

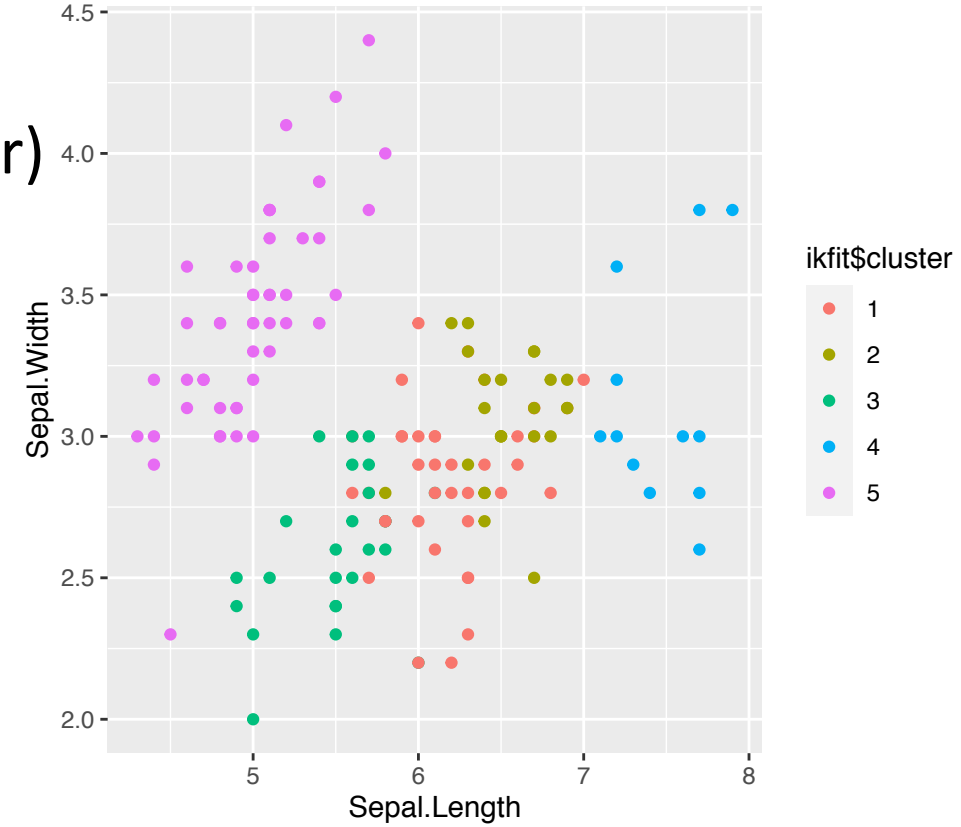
Compared to 78.85 for 3 clusters!

k-Means clustering in R

Plotting the clusters:

> ikfit$cluster = as.factor(ikfit$cluster)

> ggplot(iris, aes(Sepal.Length, Sepal.Width, color = ikfit$cluster)) + geom_point()



Countries data

Sample socio-economic data for 19 countries.

Country	Per capita income	Literacy	Infant mortality	Life expectancy
Brazil	10326	90	23.6	75.4
Germany	39650	99	4.08	79.4
Mozambique	830	38.7	95.9	42.1
Australia	43000	99	4.57	81.2
China	5300	90.9	23	73
Argentina	13308	97.2	13.4	75.3
United Kingdom	34105	99	5.01	79.4
South Africa	10600	82.4	44.8	49.3
Zambia	1000	68	92.7	42.4
Namibia	5249	85	42.3	52.9
Georgia	4200	100	17.36	71
Pakistan	3320	49.9	67.5	65.5
India	2972	61	55	64.7
Turkey	12888	88.7	27.5	71.8
Sweden	34735	99	3.2	80.9
Lithuania	19730	99.6	8.5	73
Greece	36983	96	5.34	79.5
Italy	26760	98.5	5.94	80
Japan	34099	99	3.2	82.6

Countries data: scaling

Sc
co

```
> summary(CD)
      Country   Per.capita.income    Literacy      Infant.mortality Life.expectancy
 Argentina: 1   Min.   :  830     Min.   : 38.70   Min.   : 3.200   Min.   :42.10
 Australia: 1   1st Qu.: 4724     1st Qu.: 83.70   1st Qu.: 5.175   1st Qu.:65.10
 Brazil   : 1   Median :12888     Median : 96.00   Median :17.360   Median :73.00
 China    : 1   Mean   :17845     Mean   : 86.36   Mean   :28.574   Mean   :69.44
 Georgia  : 1   3rd Qu.:34102     3rd Qu.: 99.00   3rd Qu.:43.550   3rd Qu.:79.45
 Germany  : 1   Max.   :43000     Max.   :100.00   Max.   :95.900   Max.   :82.60
 (Other)  :13

> # scale numerical data
> CD[,2:5] = scale(CD[,2:5])

> summary(CD)
      Country   Per.capita.income    Literacy      Infant.mortality  Life.expectancy
 Argentina: 1   Min.   :-1.1367   Min.   :-2.5773   Min.   :-0.8459   Min.   :-2.0659
 Australia: 1   1st Qu.:-0.8765   1st Qu.:-0.1440   1st Qu.:-0.7800   1st Qu.:-0.3281
 Brazil   : 1   Median :-0.3312   Median : 0.5211   Median :-0.3738   Median : 0.2688
 China    : 1   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 Georgia  : 1   3rd Qu.: 1.0861   3rd Qu.: 0.6833   3rd Qu.: 0.4993   3rd Qu.: 0.7562
 Germany  : 1   Max.   : 1.6805   Max.   : 0.7374   Max.   : 2.2444   Max.   : 0.9942
 (Other)  :13
```

Countries data: k-Means

k-Means for the scaled data set

```
> set.seed(9999)
> CD <- read.csv("CountriesData.csv")
> # scale numerical data
> CD[,2:5] = scale(CD[,2:5])
> CDkfit = kmeans(CD[,2:5], 3, nstart = 20)
> CDkfittable(actual = CD$Country, fitted =
  CDkfit$cluster)
```

Non–scaled *v* scaled clusters

Not-scaled

actual	fitted 1	2	3
Argentina	1	0	0
Australia	0	0	1
Brazil	1	0	0
China	1	0	0
Georgia	1	0	0
Germany	0	0	1
Greece	0	0	1
India	0	1	0
Italy	0	0	1
Japan	0	0	1
Lithuania	1	0	0
Mozambique	0	1	0
Namibia	0	1	0
Pakistan	0	1	0
South Africa	0	1	0
Sweden	0	0	1
Turkey	1	0	0
United Kingdom	0	0	1
Zambia	0	1	0

Scaled

actual	fitted 1	2	3
Argentina	1	0	0
Australia	0	0	1
Brazil	1	0	0
China	0	1	0
Georgia	0	1	0
Germany	0	0	1
Greece	0	0	1
India	0	1	0
Italy	0	0	1
Japan	0	0	1
Lithuania	1	0	0
Mozambique	0	1	0
Namibia	0	1	0
Pakistan	0	1	0
South Africa	1	0	0
Sweden	0	0	1
Turkey	1	0	0
United Kingdom	0	0	1
Zambia	0	1	0

Scaling changes the clusters of these countries.

k-Means: some considerations

The location of the initial centroids influences final clusters, some ways to address this:

- Multiple runs (which k-Means does), or
- Select more than *k* initial centroids and then select among these initial centroids.

k-Means: some considerations

How do we decide which k to use?

- Trial and error
- There is no single best way of doing this. One reference with some good approaches is https://uc-r.github.io/kmeans_clustering.
- The following shows one method, the average silhouette, adapted from Giordani et al., An Introduction to Clustering with R.

K-Means: Silhouette

- The average silhouette calculates how well each data point sits within its cluster. It is a proxy measure for the quality of the clustering. For each point, i ,

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}, i = 1, 2, 3, \ldots$$

- where a_i is the average distance between that point and all other points in the same cluster, and
- b_i is smallest average distance to any cluster it does not belong to. Ideally a_i is small and b_i is large.
- The average s can then be evaluated across all i. at different values of k.

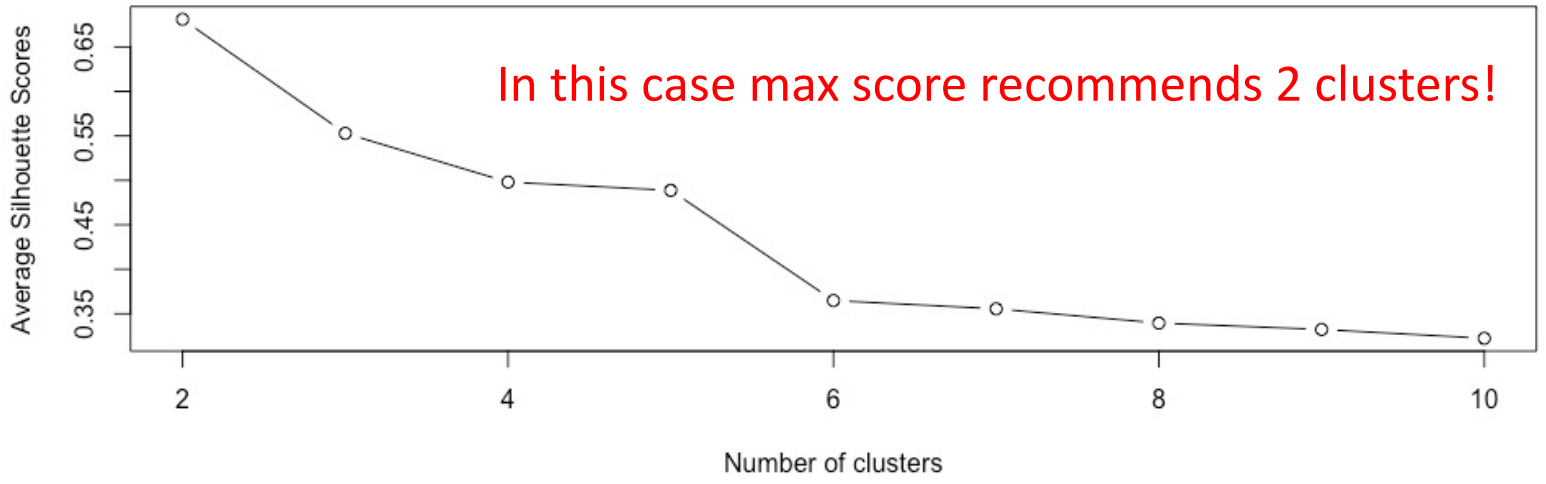
K-Means: Silhouette in R

For the iris data, using sepals and petals:

```
> library(cluster)
> #make function to get average silhouette score
> i_silhouette_score <- function(k){
>   km <- kmeans(iris[,1:4], centers = k, nstart=25)
>   ss <- silhouette(km$cluster, dist(iris[,1:4]))
>   mean(ss[, 3])
> }
```

K-Means: Silhouette in R

```
> #calc and plot average silhouette for 2-10 clusters
> k <- 2:10
> avg_sil <- sapply(k, i_silhouette_score)
> plot(k, type='b', avg_sil, xlab='Number of clusters',
  ylab='Average Silhouette Scores')
```

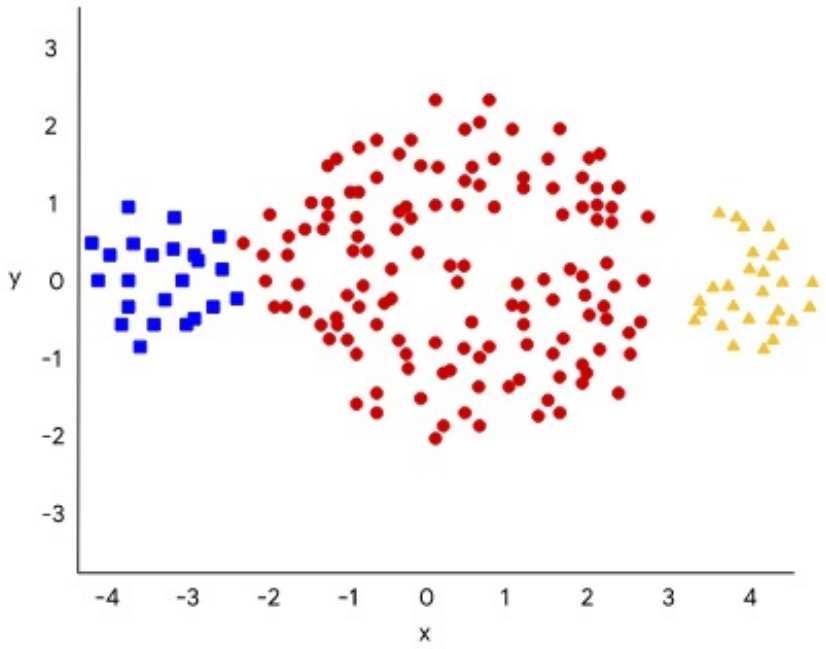


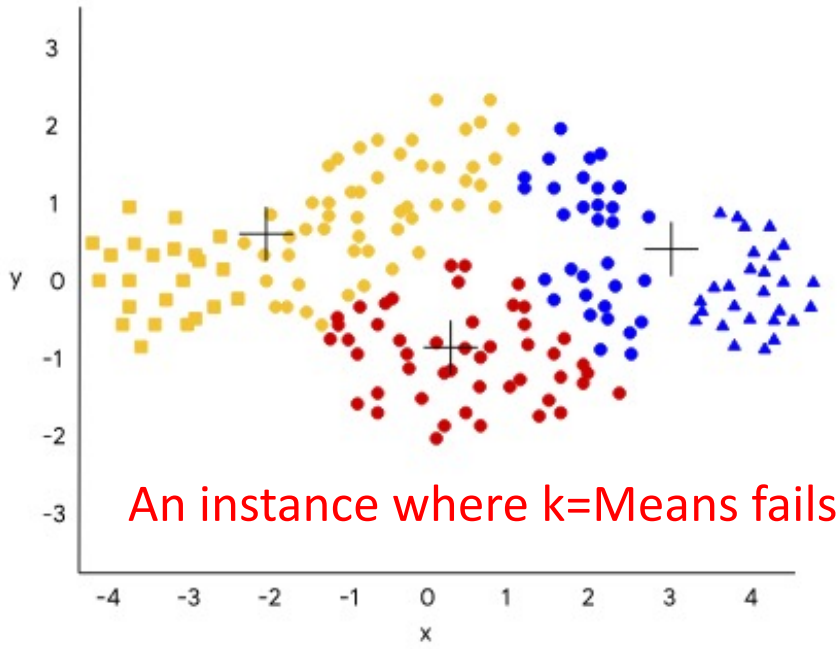
K-Means: further thoughts…

- Advantages:
  - > Relatively simple to implement. Scales to large data sets. Guarantees convergence. Can warm-start the positions of centroids. Easily adapts to new examples. Generalizes to clusters of different shapes and sizes.
- Disadvantages:
  - > Dependent on initial values. Clusters of varying sizes and density. Centroids can be dragged by outliers. Outliers might get their own cluster. Scaling with number of dimensions. Non-globular clusters.

https://developers.google.com/

K-Means: further thoughts…





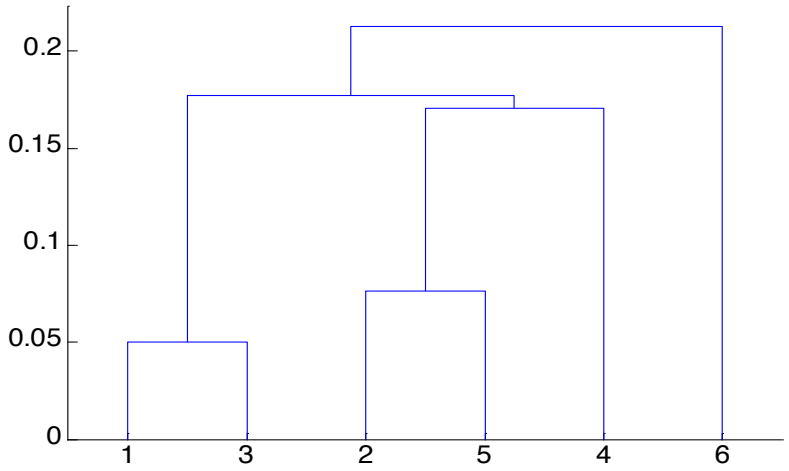
An instance where k=Means fails.

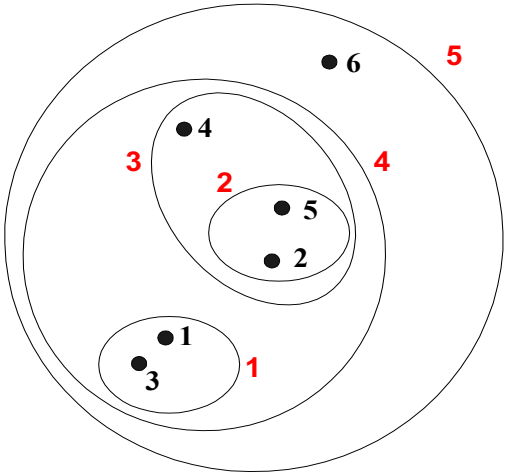
https://developers.google.com/

Hierarchical clustering

Hierarchical clustering

- Creates a set of nested clusters organized as a hierarchical tree that:
  - > Records the sequences of merges or splits
  - > Can be visualized as a dendrogram





Advantages of hierarchical clustering

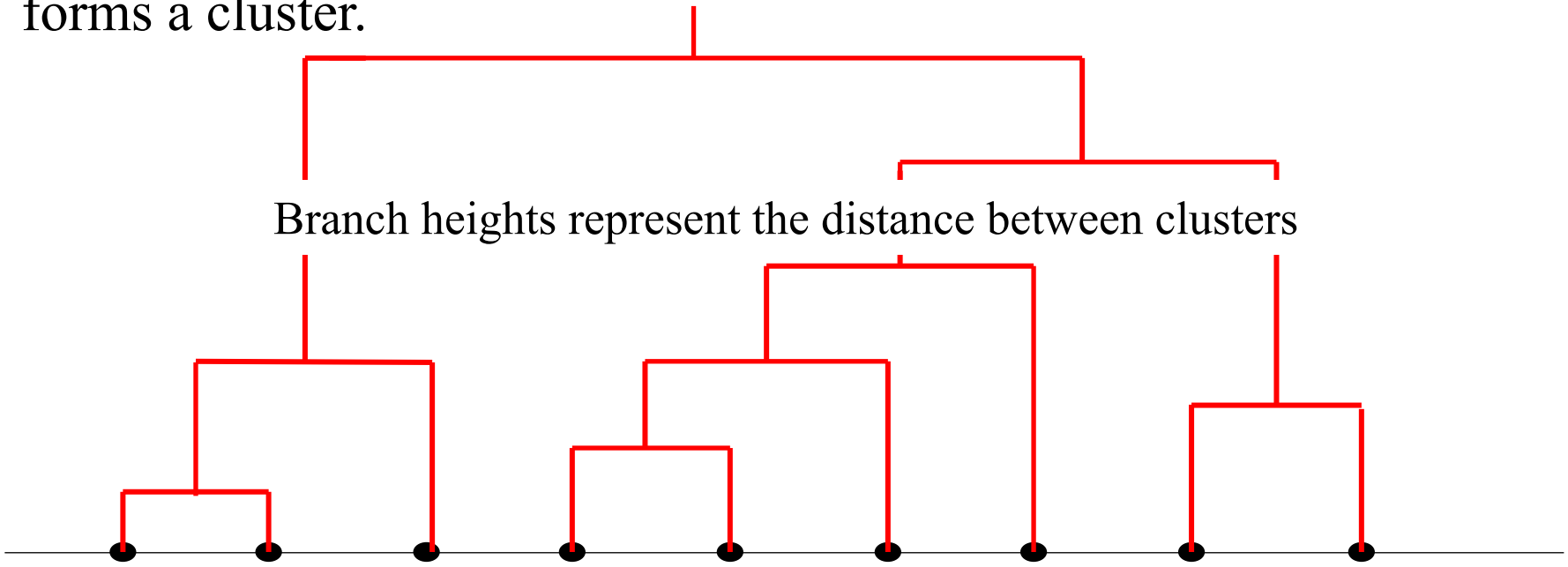
- Do not have to assume any particular number of clusters:
  - > Any desired number of clusters can be obtained by 'cutting' the dendrogram at the appropriate level.

- They may correspond to meaningful taxonomies:
  - > For example, in biological sciences (e.g., plant and animal kingdoms).

Dendrogram and hierarchies

Decompose data points into a several levels of nested partitioning (**tree** of clusters), called a **dendrogram**.

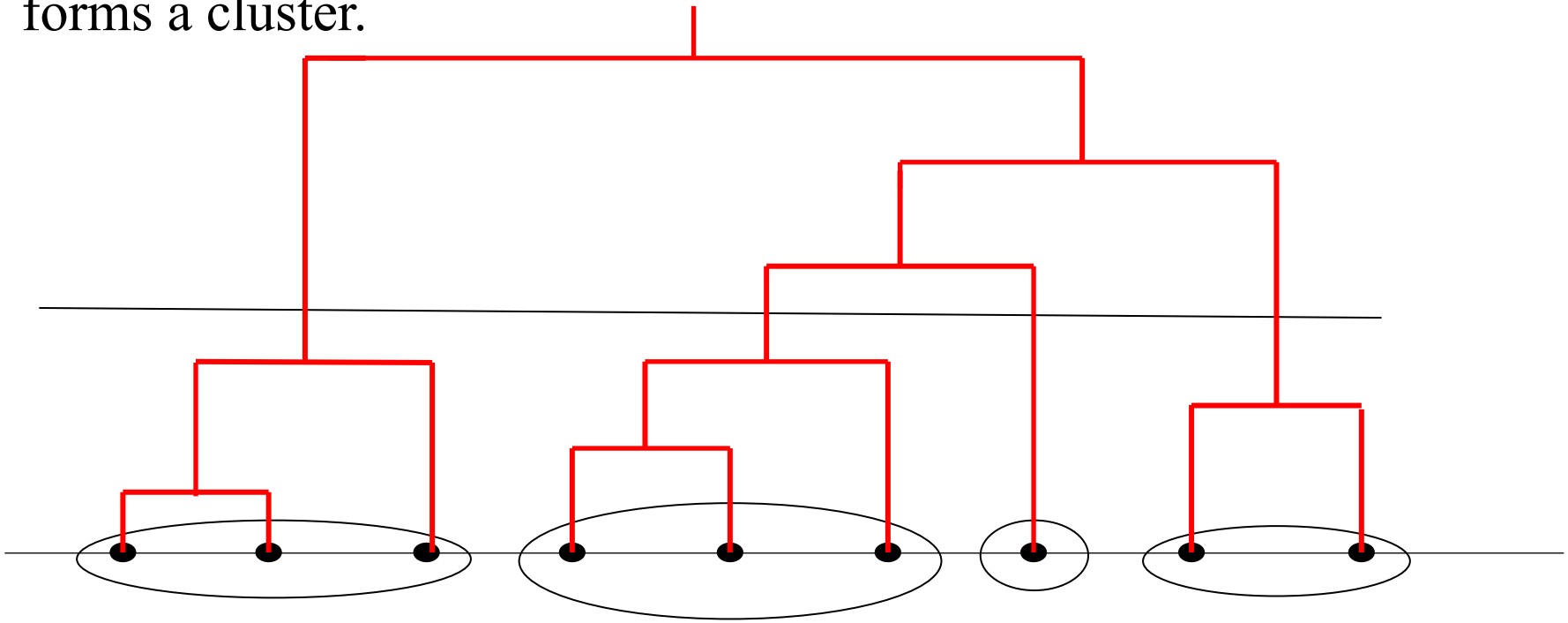
A **clustering** of the data points is obtained by **cutting** the dendrogram at the desired level, then each **connected component** forms a cluster.



Dendrogram and hierarchies

Decompose data points into a several levels of nested partitioning (**tree** of clusters), called a **dendrogram**.

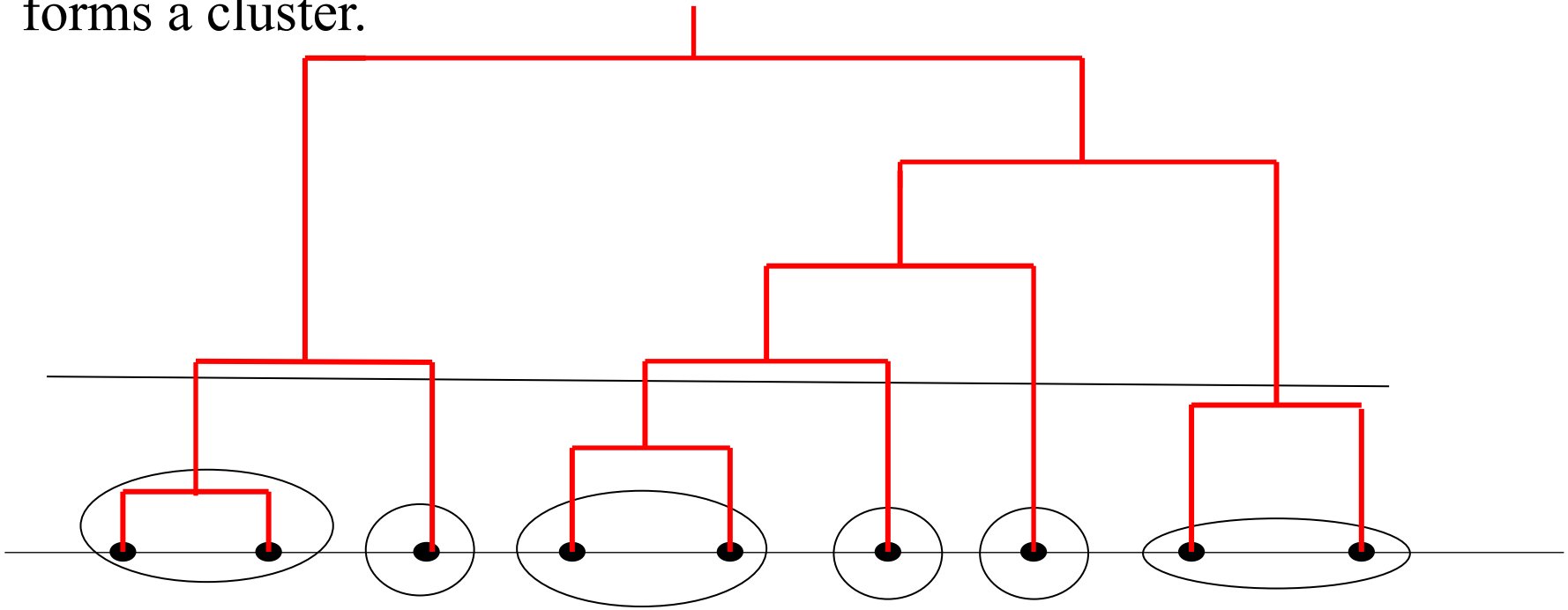
A **clustering** of the data points is obtained by **cutting** the dendrogram at the desired level, then each **connected component** forms a cluster.



Dendrogram and hierarchies

Decompose data points into a several levels of nested partitioning (**tree** of clusters), called a **dendrogram**.

A **clustering** of the data points is obtained by **cutting** the dendrogram at the desired level, then each **connected component** forms a cluster.



Hierarchical Clustering

Two main types of hierarchical clustering:

- Agglomerative:
  - > Start with the points as individual clusters
  - > At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- Divisive:
  - > Start with one, all-inclusive cluster
  - > At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix and merge or split one cluster at a time

Agglomerative Clustering Algorithm

More popular hierarchical clustering technique

Distance matrix stores the distances between each cluster

Basic algorithm is straightforward

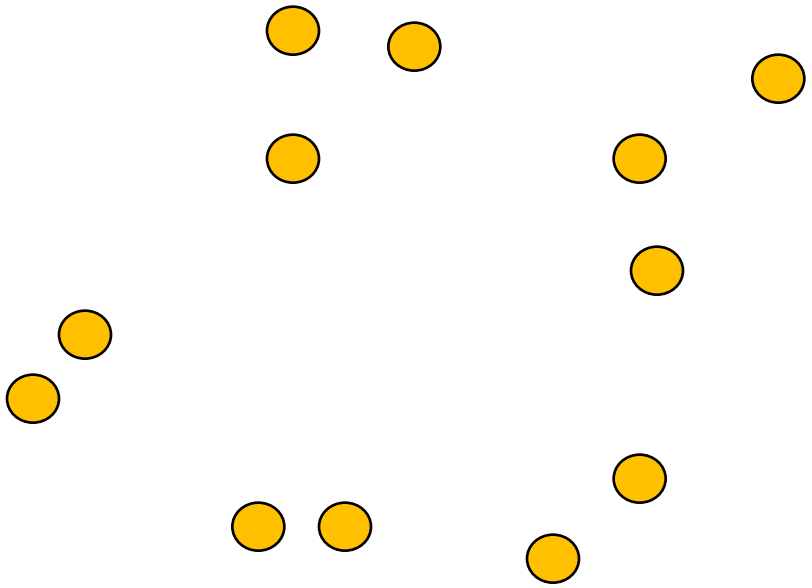
1. Compute the distance matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the distance matrix
6. **Until** only a single cluster remains

Key operation is the computation of distance between two clusters

Different approaches to defining the distance distinguish the different algorithms

Starting Situation

Start with clusters of individual points and a distance/proximity matrix



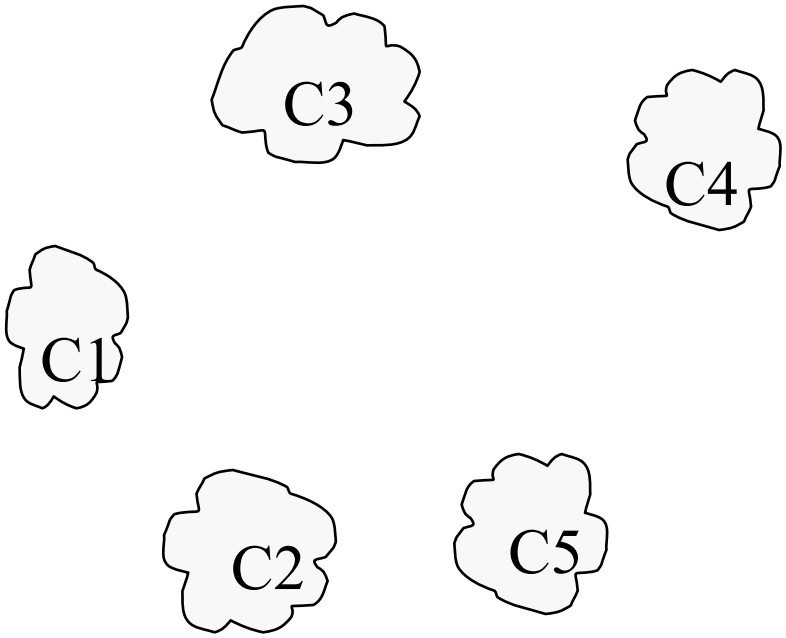
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						

Distance/ Proximity Matrix

Intermediate Situation

After some merging steps,

we have some clusters

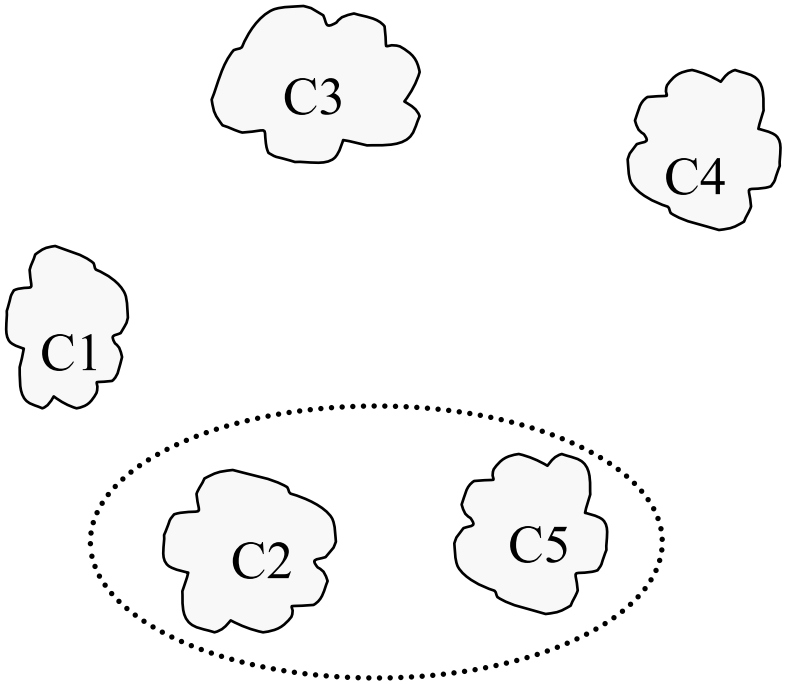


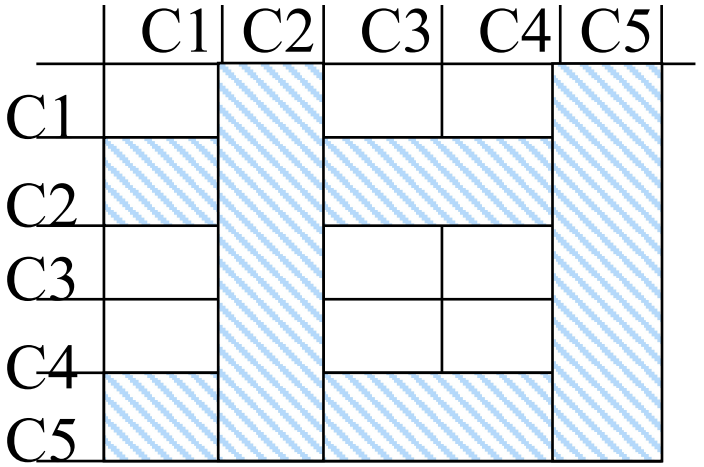
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/ Proximity Matrix

Intermediate Situation

We want to merge the two closest clusters (C2 and C5) and update the distance matrix.



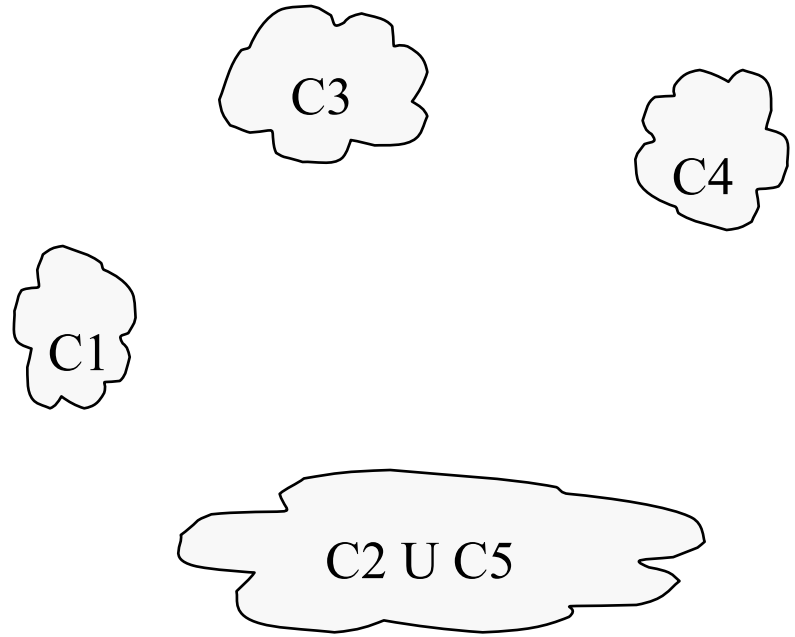


Distance/ Proximity Matrix

After Merging

The question is

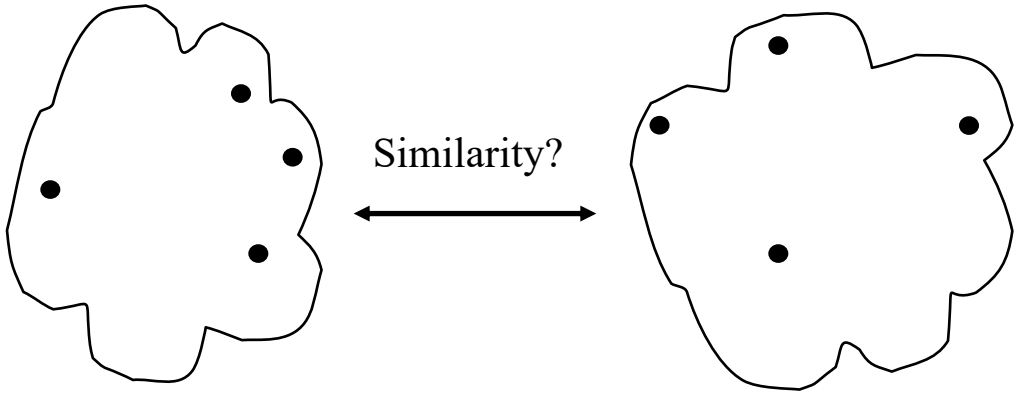
"How do we update the distance/ proximity matrix?"



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Distance/ Proximity Matrix

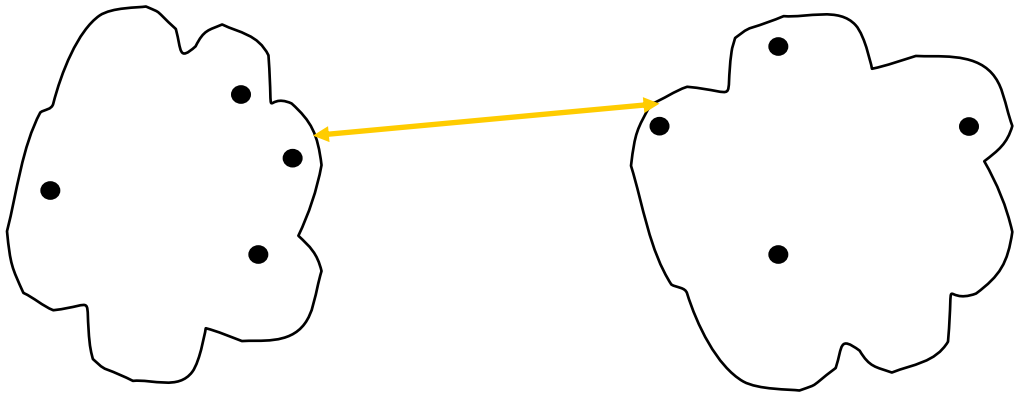
How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average
- Distance Between Centroids

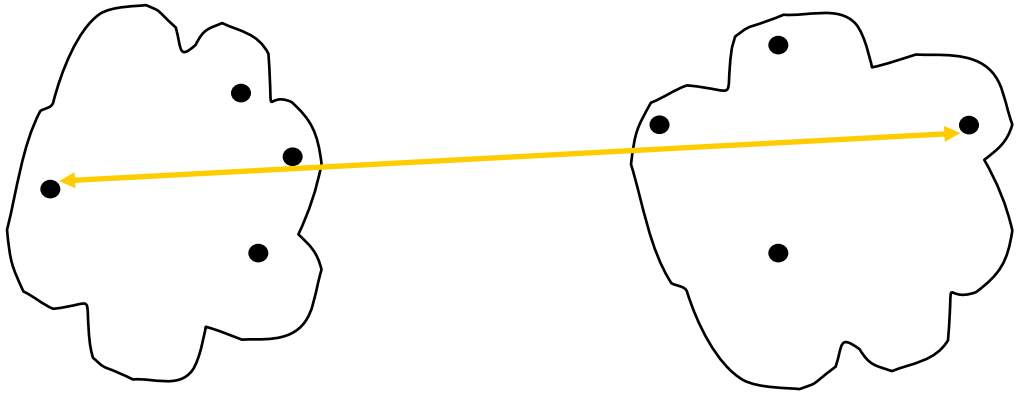
How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average
- Distance Between Centroids

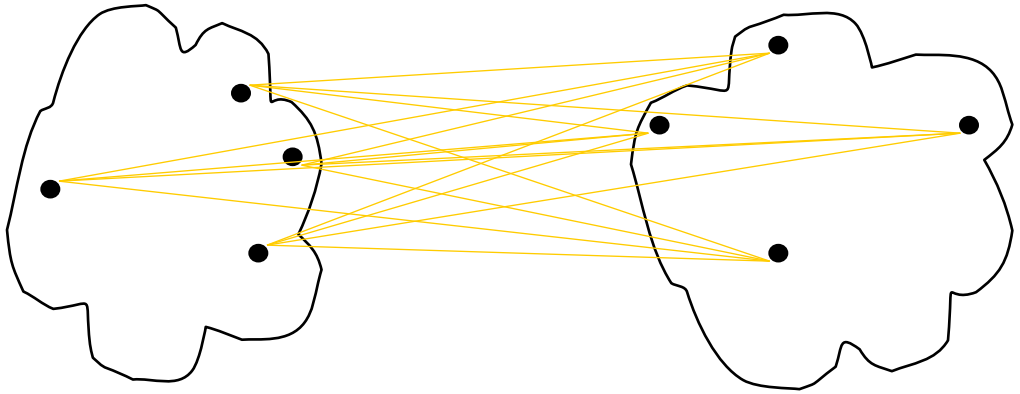
How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average
- Distance Between Centroids

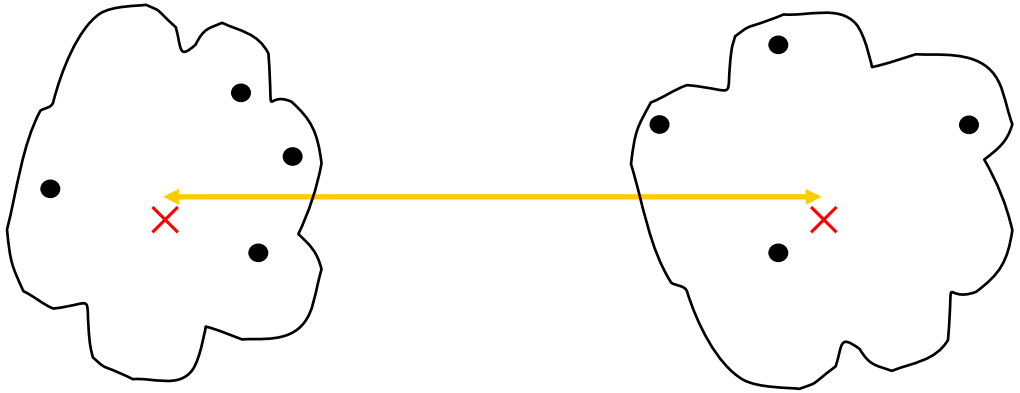
How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

- MIN
- MAX
- Group Average
- Distance Between Centroids

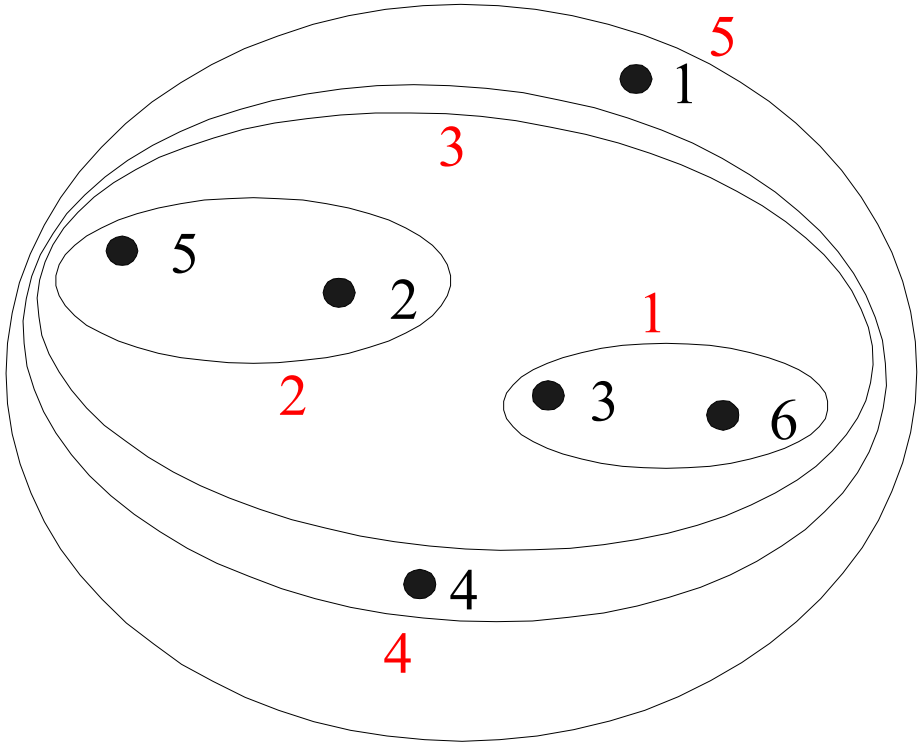
Class Activity

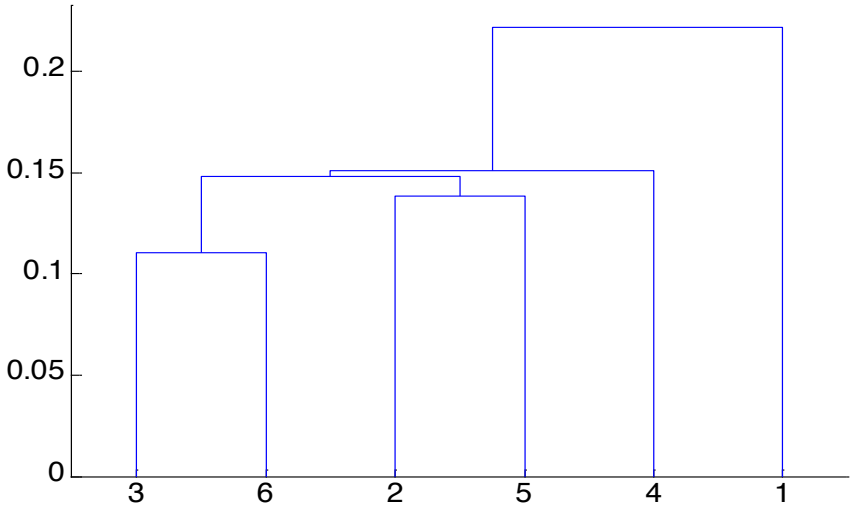
Merging with MIN, let's try first merge...

	P1	P2	P3	P4	P5	P6
P1	0	0.24	0.22	0.37	0.34	0.23
P2	0.24	0	0.15	0.2	0.14	0.25
P3	0.22	0.15	0	0.15	0.28	0.11
P4	0.37	0.14	0.15	0	0.29	0.22
P5	0.23	0.25	0.28	0.29	0	0.39
P6	0.23	0.25	0.11	0.22	0.39	0

First Join					
	P1	P2	P36	P4	P5
P1		0.24	0.22	0.37	0.34
P2			0.15	…	
P36					
P4					
P5					

Hierarchical Clustering: MIN

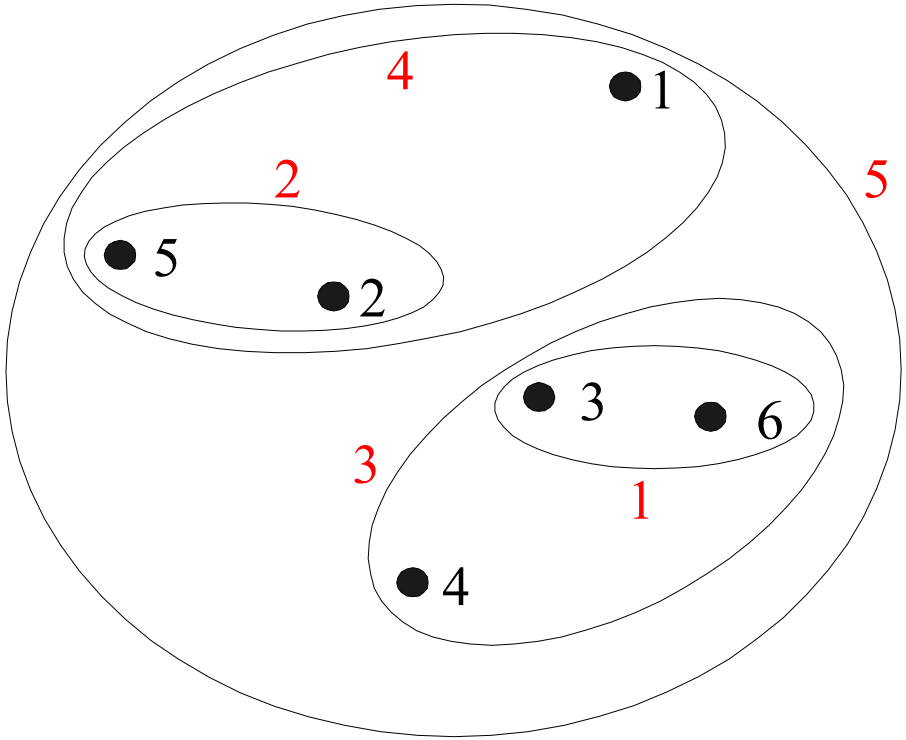




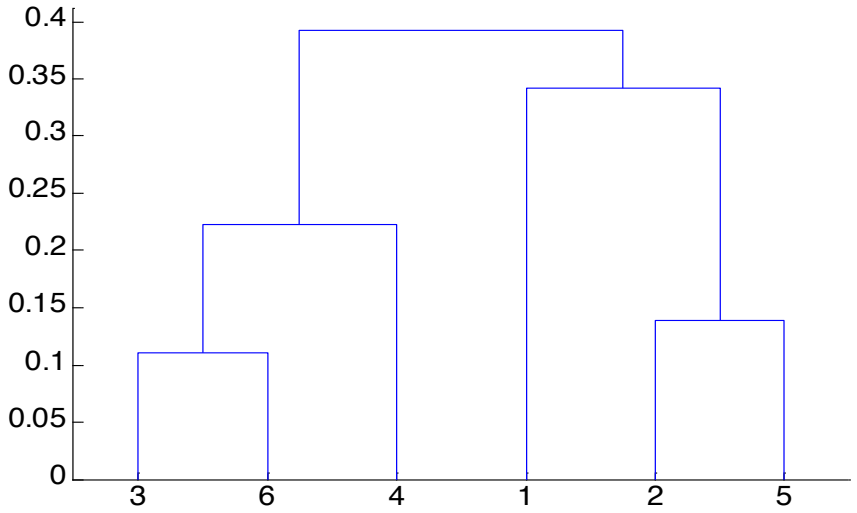
Nested Clusters

Dendrogram

Hierarchical Clustering: MAX

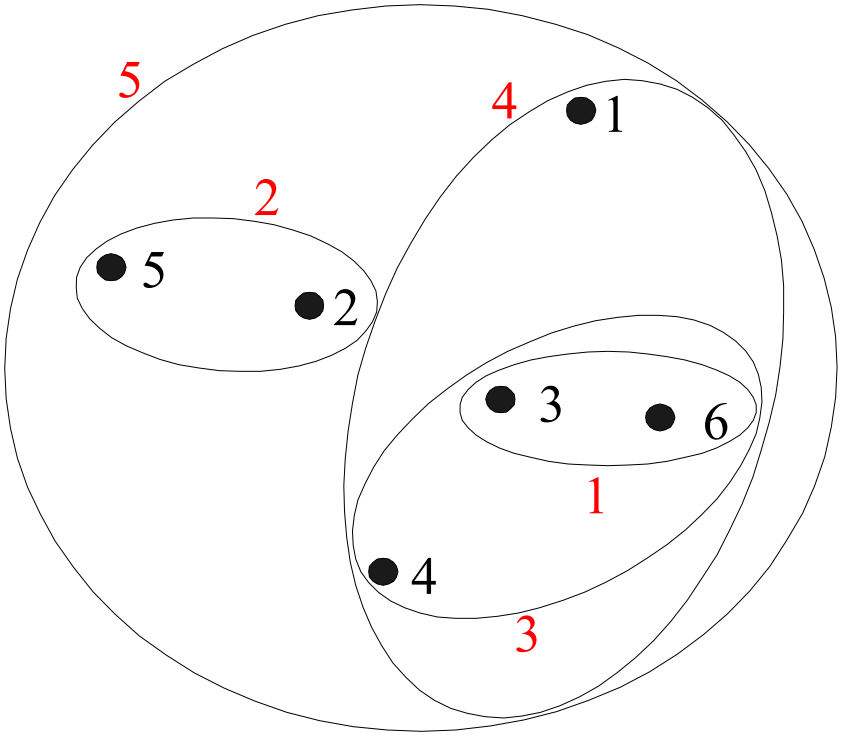


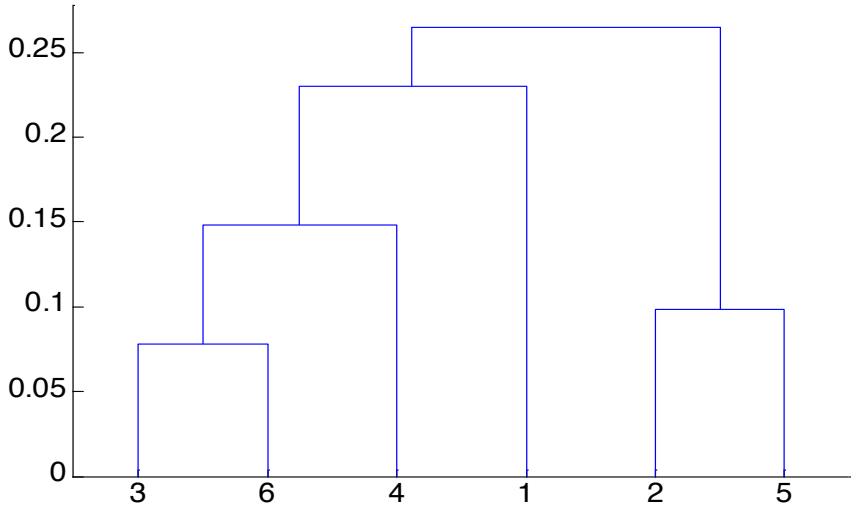
Nested Clusters



Dendrogram

Hierarchical Clustering: Group Average





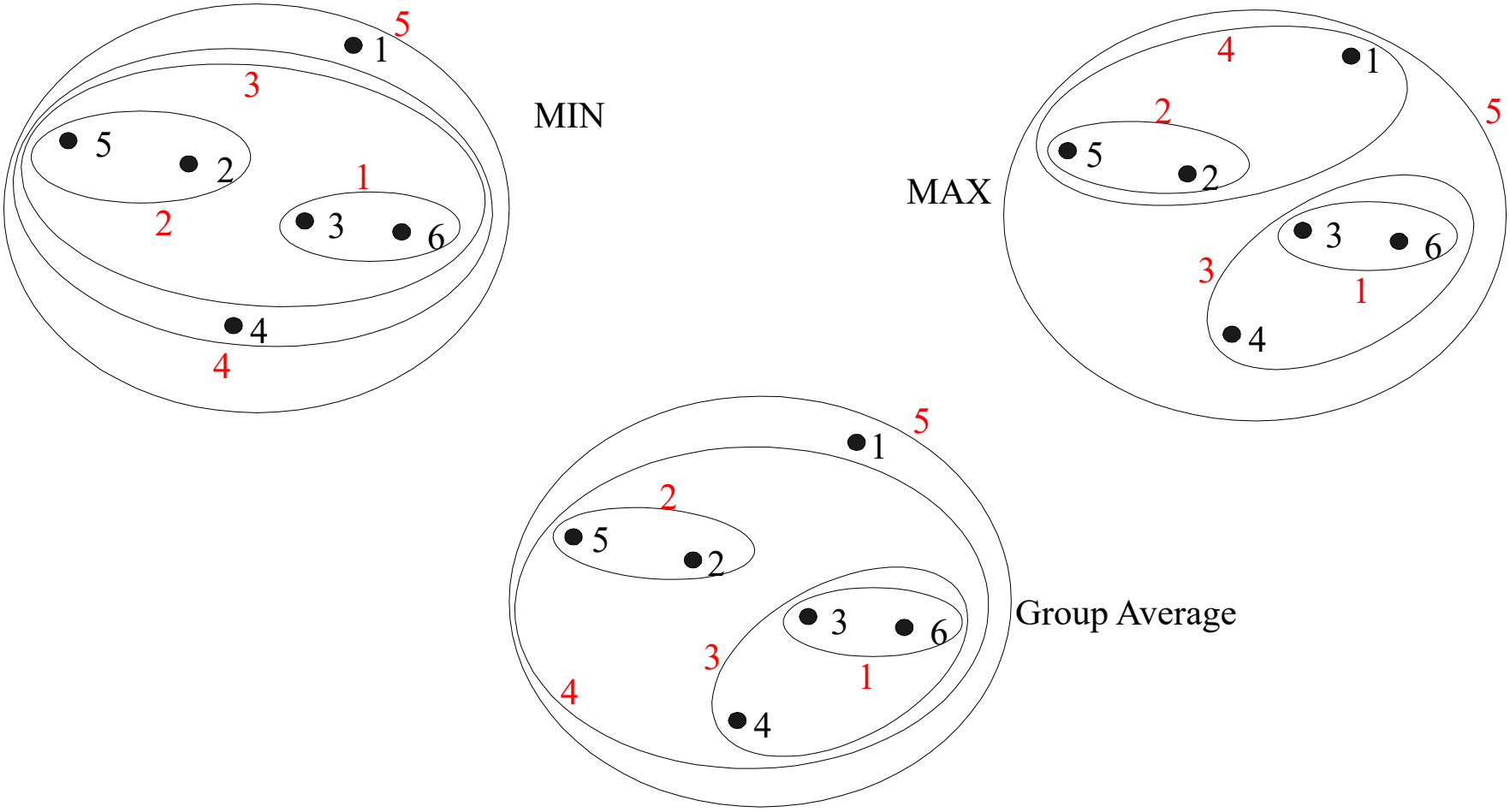
Nested Clusters

Dendrogram

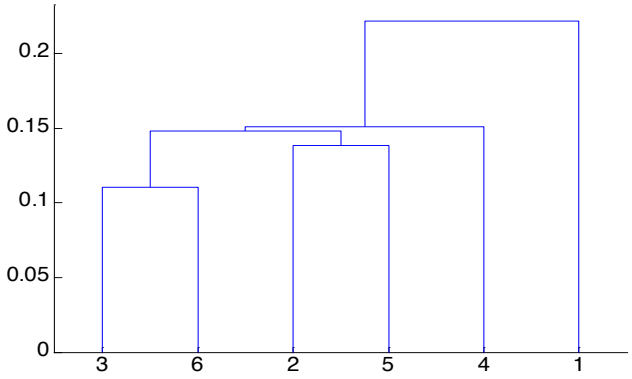
Similarity measures: pros and cons

- MIN
  - > Can handle non-elliptical shapes
  - > Sensitive to noise and outliers
- MAX
  - > Less susceptible to noise and outliers
  - > Tends to break large clusters, biased towards elliptical shapes
- Group Average
  - > Compromise between Single and Complete Link
  - > Less susceptible to noise and outliers
  - > Biased towards globular clusters

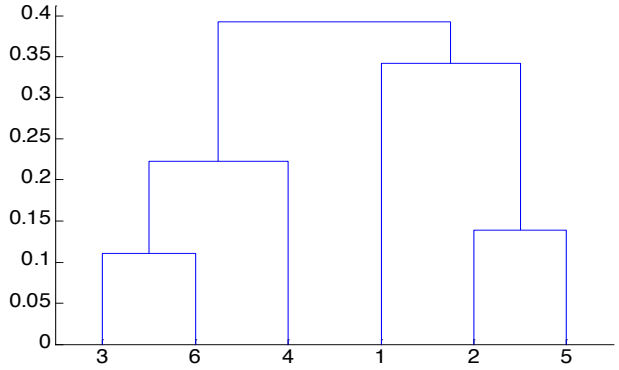
Hierarchical Clustering: Comparison



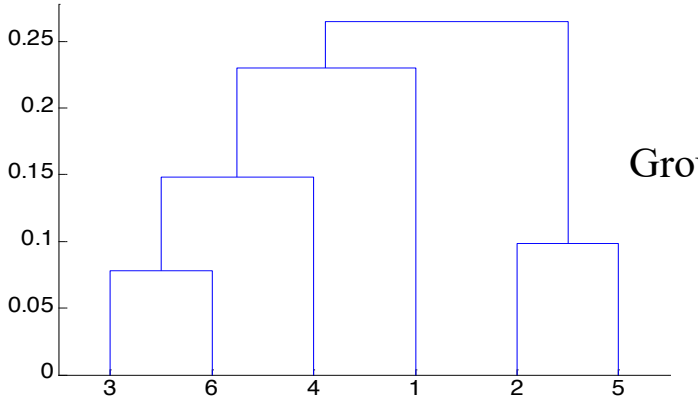
Hierarchical Clustering: Comparison



MIN



MAX



Group Average

Hierarchical clustering: considerations

- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly minimized, unlike k-Means.
- Different schemes have problems with one or more of the following:
  - > Sensitivity to noise and outliers.
  - > Difficulty handling different sized clusters and convex shapes.
  - > Breaking large clusters.

Hierarchical clustering in R

Hierarchical clustering of the Iris data using the function hclust (also part of the Stats package):

```
> set.seed(9999)
> data("iris")niris = iris
> #scale numerical data this gives poorer result for iris
> #niris[,1:4] = scale(niris[,1:4])
> ihfit = hclust(dist(niris[,1:4]), "ave")
> plot(ihfit, hang = -1)
```

Hierarchical clustering in R

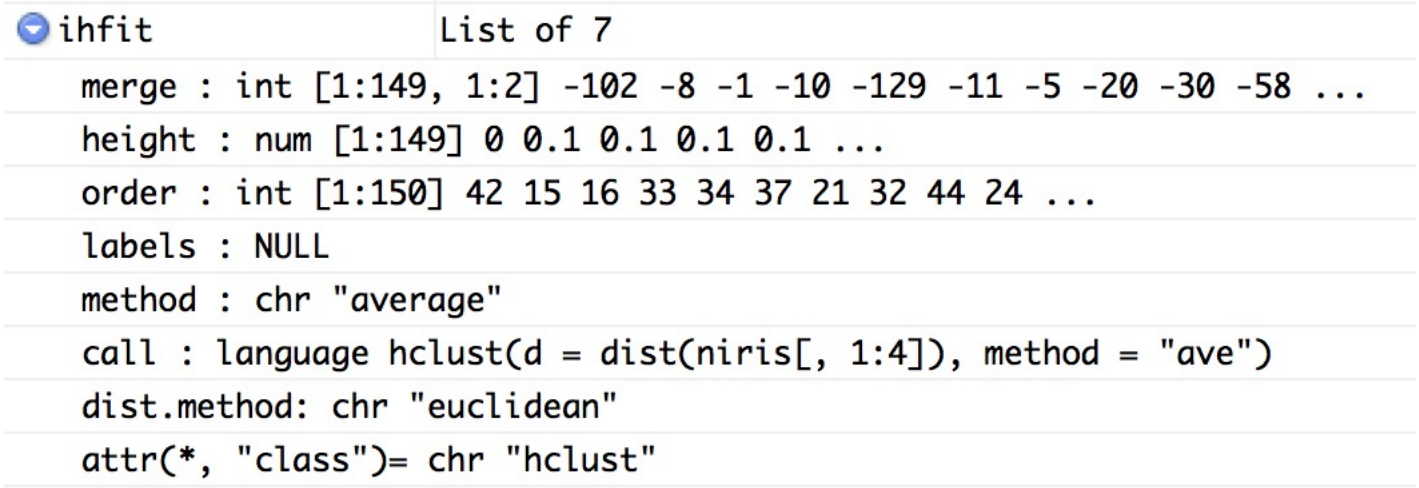
The fitted object:

```
> ihfit
Call:
hclust(d = dist(niris[, 1:4]), method = "ave")

Cluster method   : average Distance         :
euclidean
Number of objects: 150
```

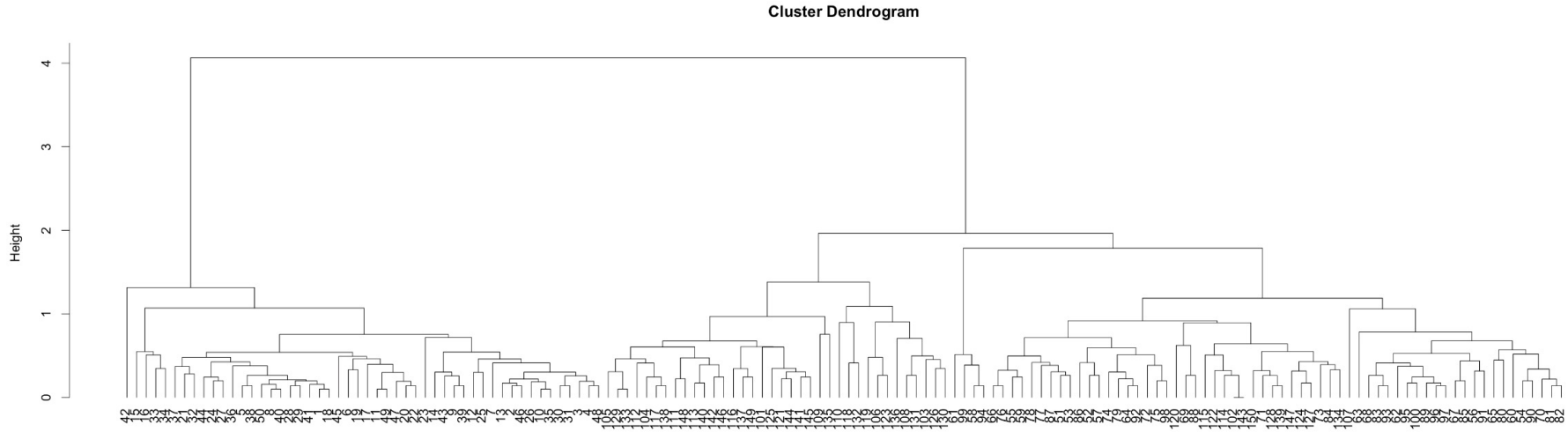
Hierarchical clustering in R

Viewing in the environment browser:



Hierarchical clustering in R

Dendrogram:



Where are the clusters?

How many do you want?

Hierarchical clustering in R

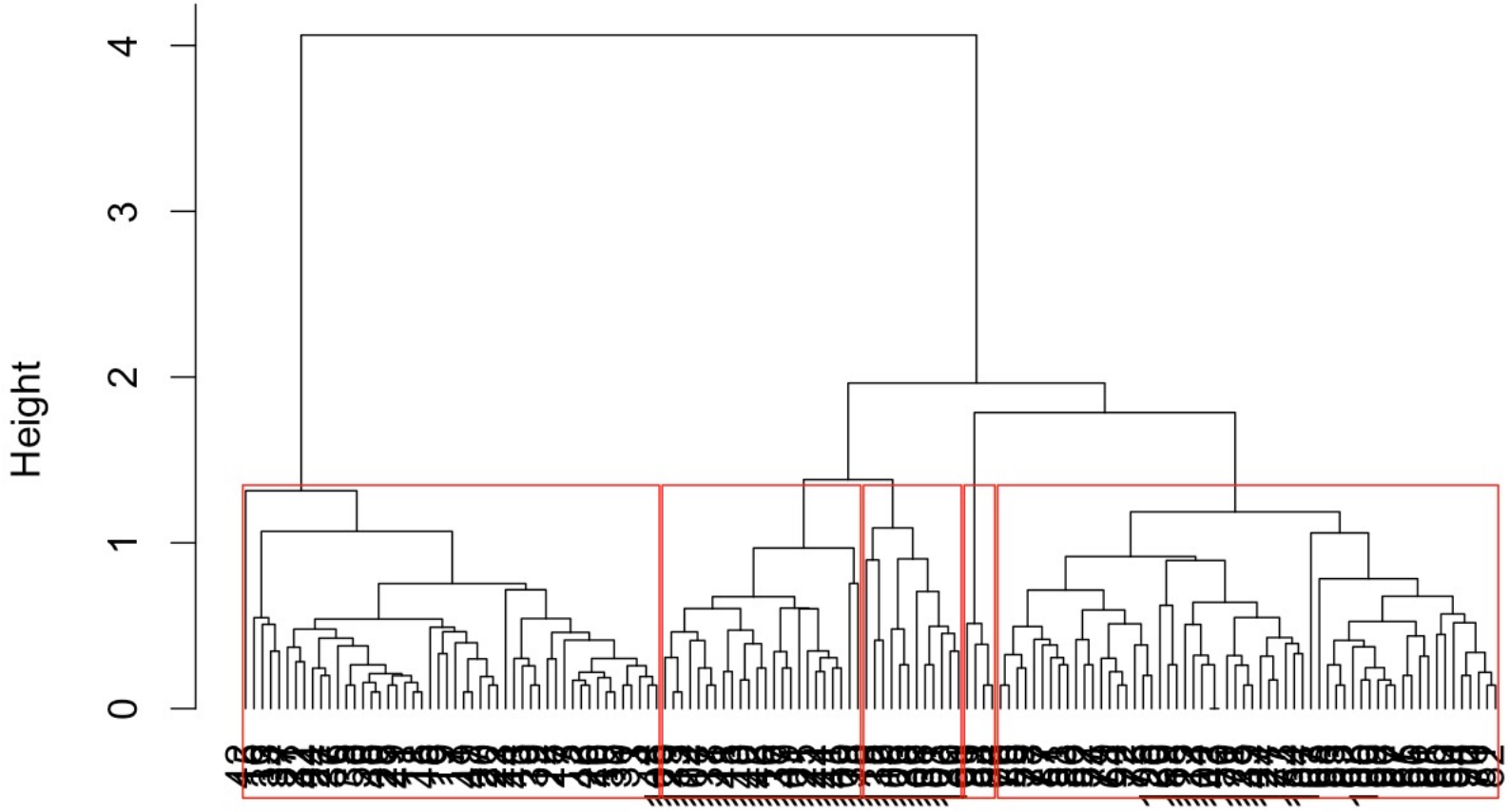
Setting a particular number of clusters:

```
> # pruning the tree into 5 clusters
> cutihfit = cutree(ihfit, k = 5)
> rect.hclust(ihfit, k = 5, border = "red")
> table(actual = niris$Species, fitted = cutihfit)
```

```
            fitted
actual        1  2  3  4  5
  setosa     50  0  0  0  0
  versicolor  0 46  4  0  0
  virginica   0 14  0 24 12
```

Hierarchical clustering in R

Dendrogram showing 5 clusters:



? hclust

- Description

  Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

- Usage

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

d           dissimilarity structure
method      agglomeration method (many to choose)
...
```

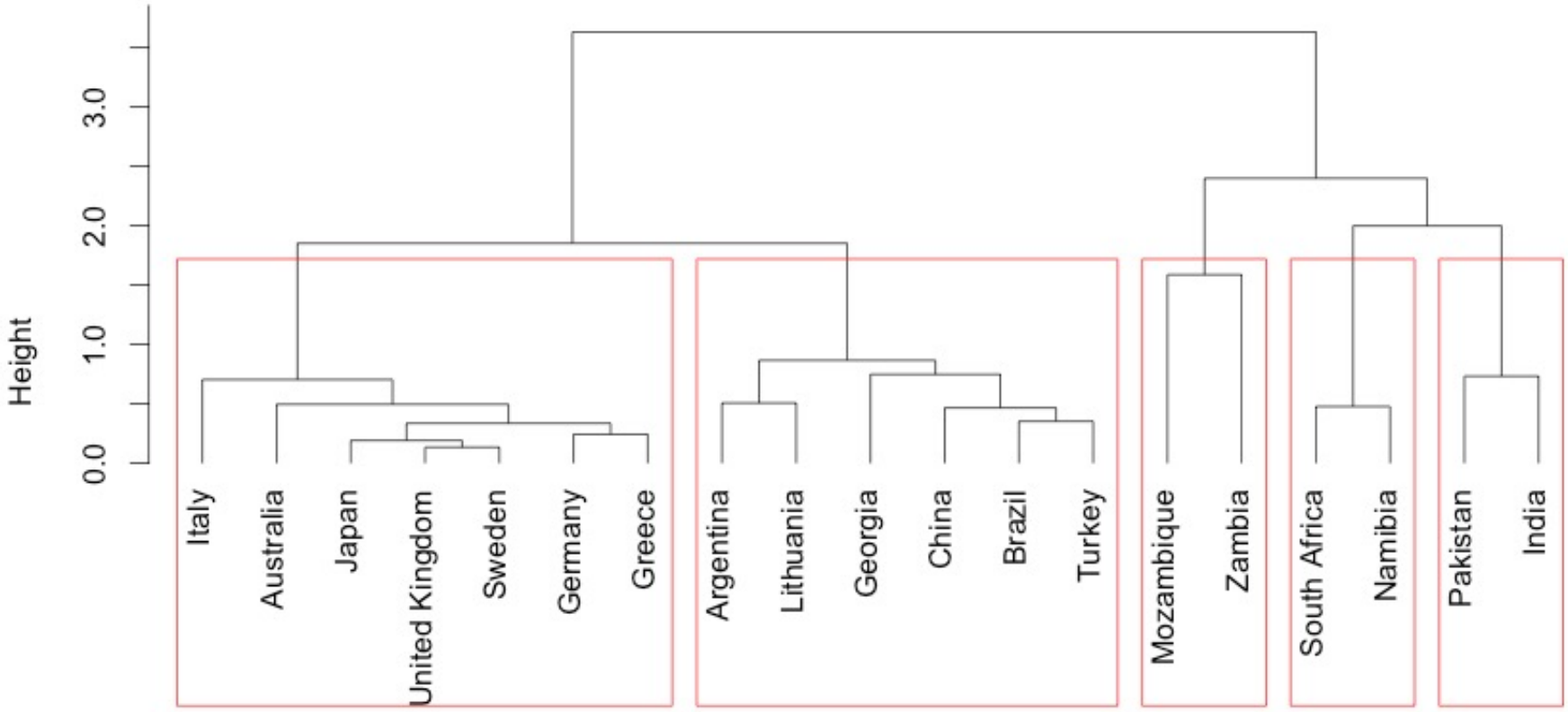
Countries data (scaled)

Reading data, scaling, setting row names to country names (to appear in dendrogram)

```
> CD <- read.csv("CountriesData.csv")
> CD[,2:5] = scale(CD[,2:5])
> rownames(CD) = CD$Country
> hfit = hclust(dist(CD[,2:5]), "average")
> plot(CDhfit)plot(CDhfit, hang = -1)
> cutCDhfit = cutree(CDhfit, k = 5) #Pruning
> rect.hclust(CDhfit, k = 5, border = "red")
```

Countries data

Dendrogram



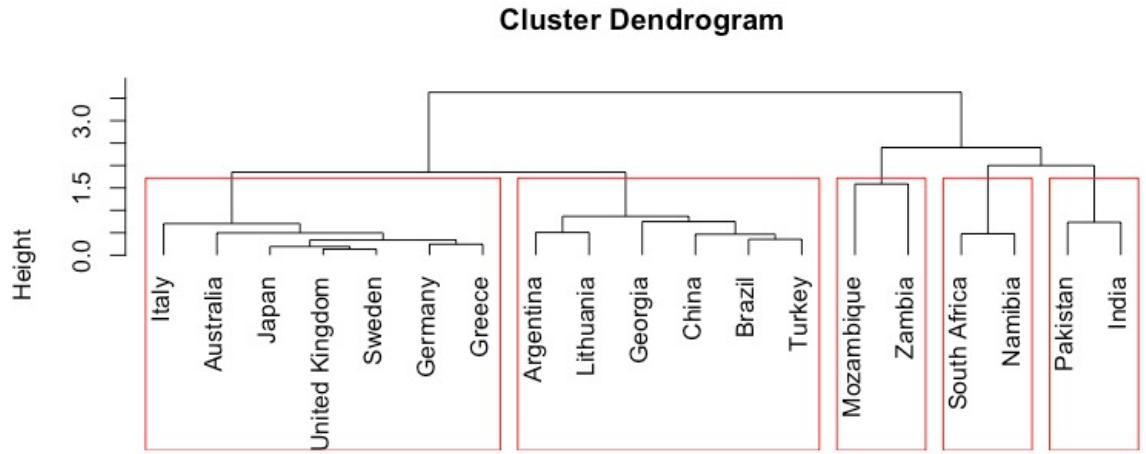
Countries data (normalised)

Normalised input gives similar tree to scaled data:

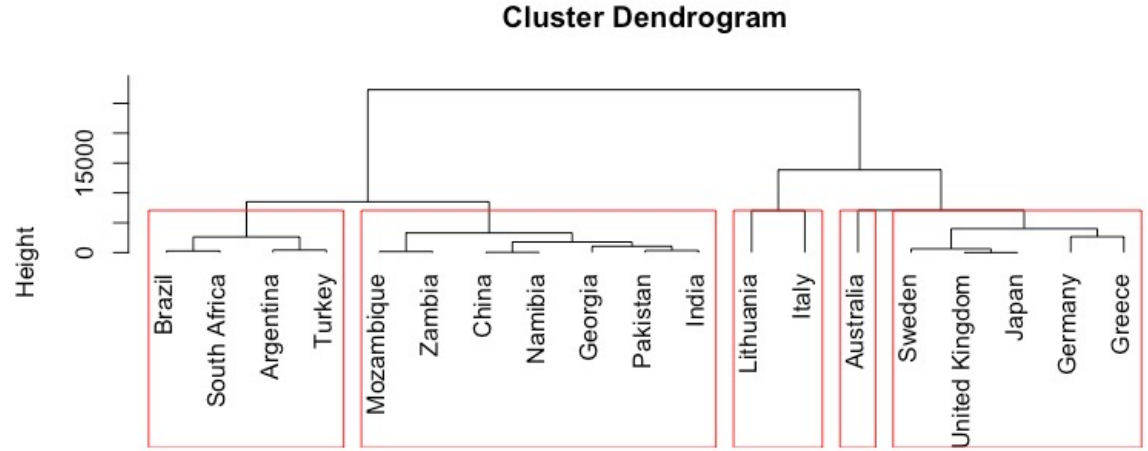
```
> CD <- read.csv("CountriesData.csv")
> # for loop to normalise cols 2 - 5
> for (i in 2:5){
> CD[,i] = (CD[,i]-min(CD[,i]))/(max(CD[,i])-min(CD[,i]))
> }
> rownames(CD) = CD$CountryCD
> hfit = hclust(dist(CD[,2:5]), "average")
> ...
```

Countries data: effect of scaling

Scaled



Not-scaled



Closing remarks

Clustering:

- An important unsupervised learning tool for grouping data.
- Enables data reduction (i.e. to identify representative subsets of the data).

Many R packages for cluster analysis:

- Cluster – is one of these which gives more control over clustering algorithm and additional analysis tools.

Solutions to review questions

1. C
2. B
3. C
4. C

References to this lecture

- James et al., An Introduction to Statistical Learning with Applications in R, 2nd Ed. Springer, 2021. Section 12.4.
- Giordani, Ferraro and Martella, An Introduction to Clustering with R. Springer, 2020.

Notes on the presentation

This presentation contains slides created to accompany: *Introduction to Data Mining*, Tan, Steinbach, Kumar. Pearson Education Inc., 2006.

Presentation originally created by Dr. Sue Bedingfield, with additions by Rui Jie Chow & Dr. Parthan Kasarapu.