Machine Learning 101

Patrick Boily

Department of Mathematics and Statistics, University of Ottawa
Idlewyld Analytics and Consulting Services
Data Action Lab

(in collaboration with Jen Schellinck and Shintaro Hagiwara)

Instructor

Bio

- Professor [’19 – now, Dept. of Math & Stats, uOttawa]
- Lecturer [’99 – now, uOttawa | UQO | Carleton] (~55 courses/workshops)
- Manager [’12 – ‘19, CQADS, Carleton]
- Adjunct Research Professor [’18 – ‘’21, Sprott School of Business, Carleton]
- Public servant [’08 – ’12, CBSA | StatCan | TC | PWGSC]
- Likes symmetry and patterns
- Cheers for the Sens & fancies himself a handyman

Projects

- Global Affairs Canada
- Nuclear Waste Management Organization
- Canadian Air Transport Security Authority
- Various other clients: ~40 projects

Specialties

- Data visualization, data cleaning (unfortunately)
- Applying wide breadth of techniques to all types of data

Contents

Clustering and k-Means

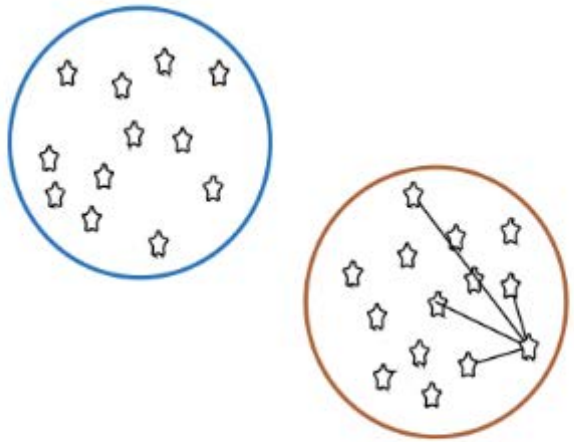
[Scott Adams, *Dilbert*, Jun 13, 1997]

Clustering Overview

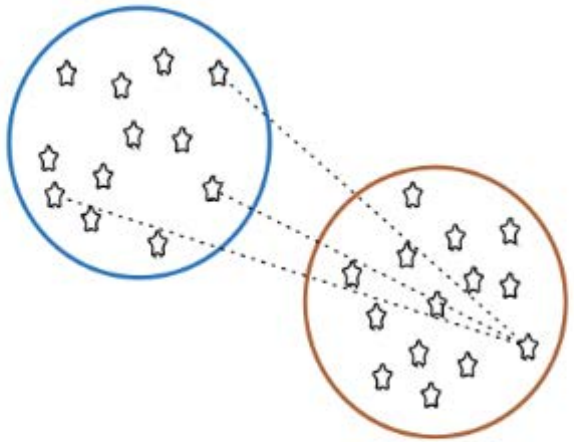
In **clustering**, the data is divided into **naturally occurring groups**. Within each group, the data points are **similar**; from group to group, they are **dissimilar**.

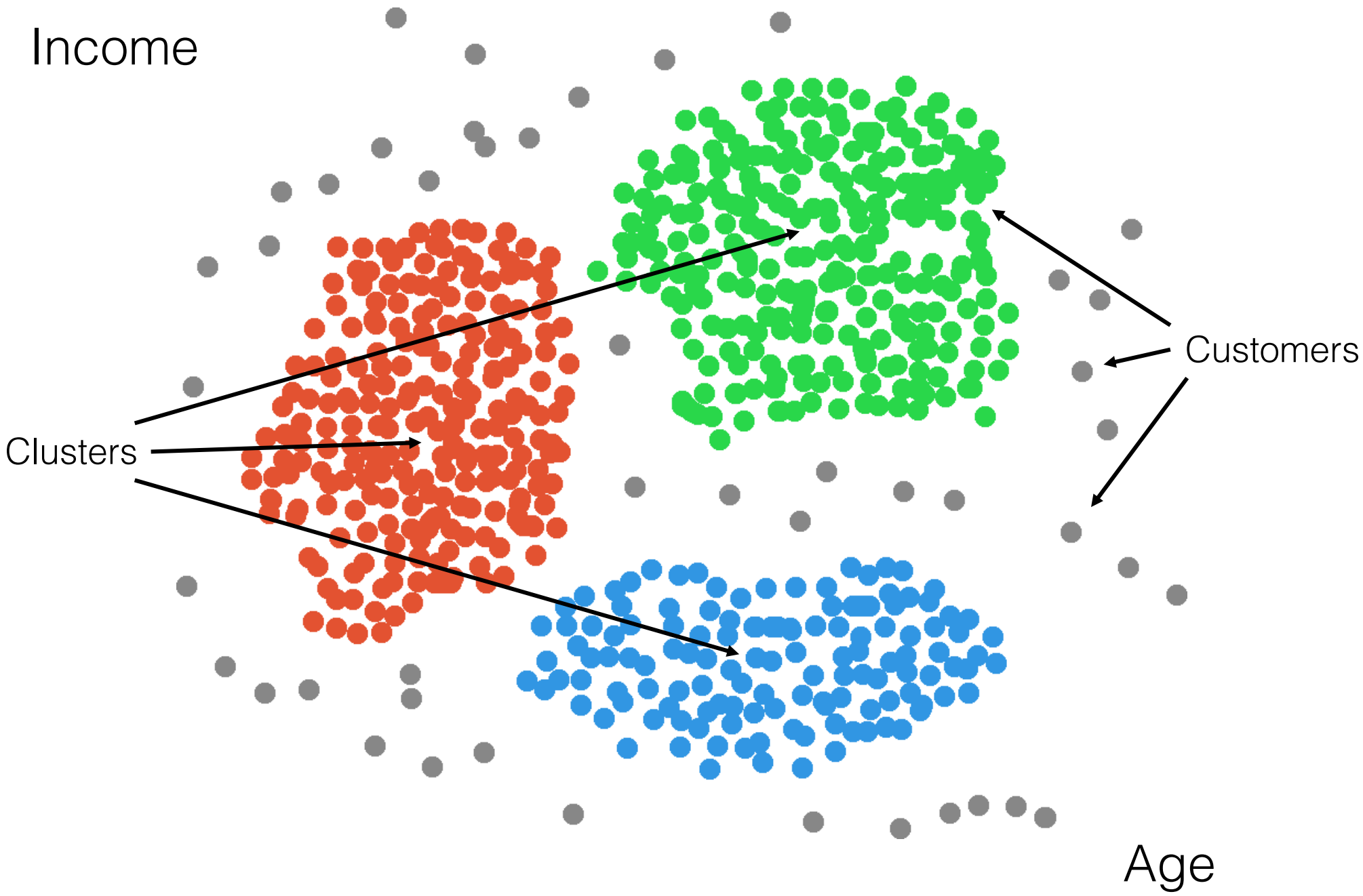
The grouping labels are not determined ahead of time, so clustering is an example of **unsupervised** learning.

average distance to points in own cluster (**low is good**)



average distance to points in neighbouring cluster (**high is good**)




Income
Customers
Clusters
Age

Clustering Overview

Clustering is a relatively **intuitive** concept for human beings as our brains do it unconsciously

- facial recognition
- searching for patterns, etc.

In general, people are very good at **messy** data, but computers and algorithms have a harder time.

Part of the difficulty is that there is **no agreed-upon definition of what constitutes a cluster:**

- “I may not be able to define what it is, but I know one when I see one”

Clustering Overview

Clustering algorithms can be **complex** and **non-intuitive**, based on varying notions of similarities between observations.

- in spite of that, the temptation to explain clusters *a posteriori* is **strong**

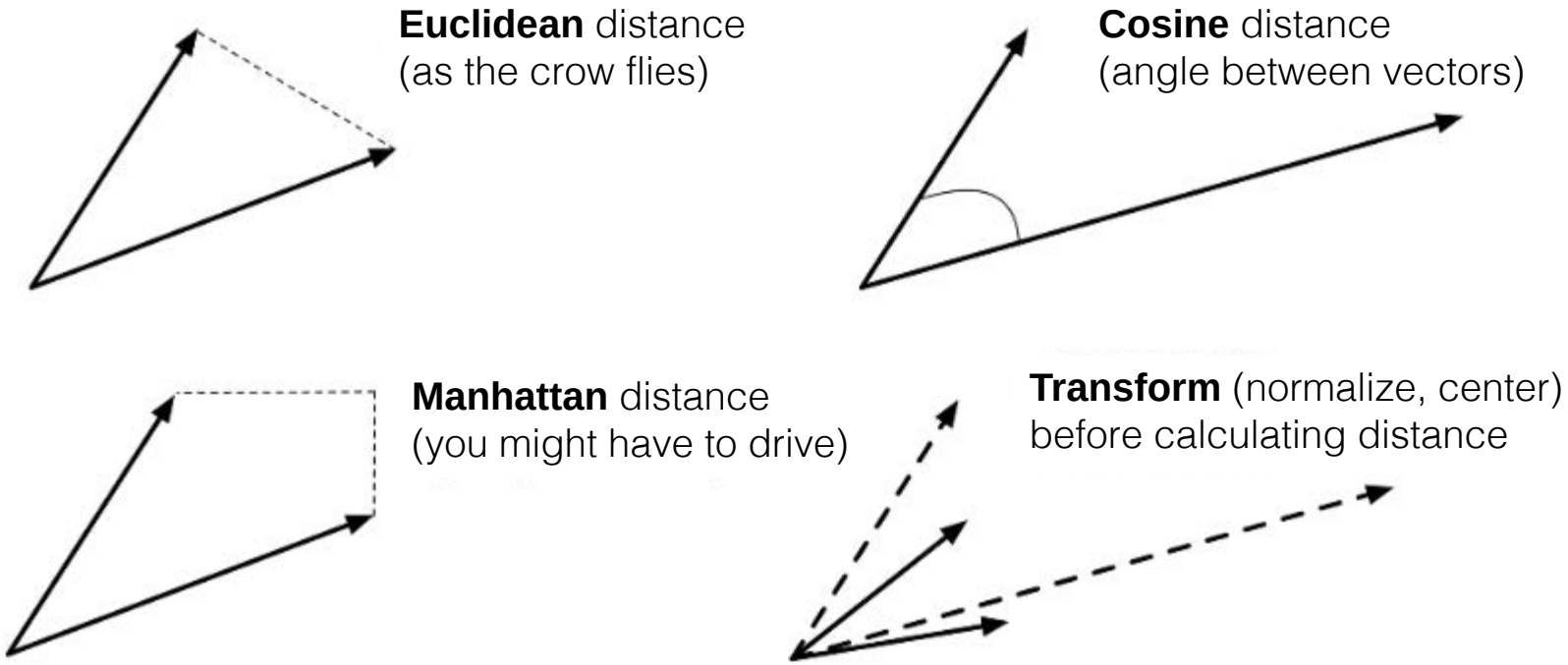
They are also (typically) **non-deterministic**:

- the same algorithm, applied twice (or more) to the same dataset, can discover completely different clusters
- the order in which the data is presented can play a role
- so can starting configurations

Discussion: What does this (potential) non-repeatability imply for validation?

Clustering Requirement

A measure of **similarity** w (or a distance d) between observations.



Typically, $w \to 1$ as $d \to 0$, and $w \to 0$ as $d \to \infty$.

Other metrics also available: Hamming, Jaccard, Pearson, etc.

Applications

Text Documents

- grouping similar documents according to their topics, based on the patterns of common and unusual words

Product Recommendations

- grouping online purchasers based on the products they have viewed, purchased, liked, or disliked
- grouping products based on customer reviews

Marketing and Business

- grouping client profiles based on their demographics and preferences

Applications

Music

- finding similar albums by grouping the customers who own them

Social Network Analysis

- recognizing communities within large groups of people

Medical Imaging

- differentiating between different tissue types in a 3D voxel

Genetic Clustering

- inferring structures in populations

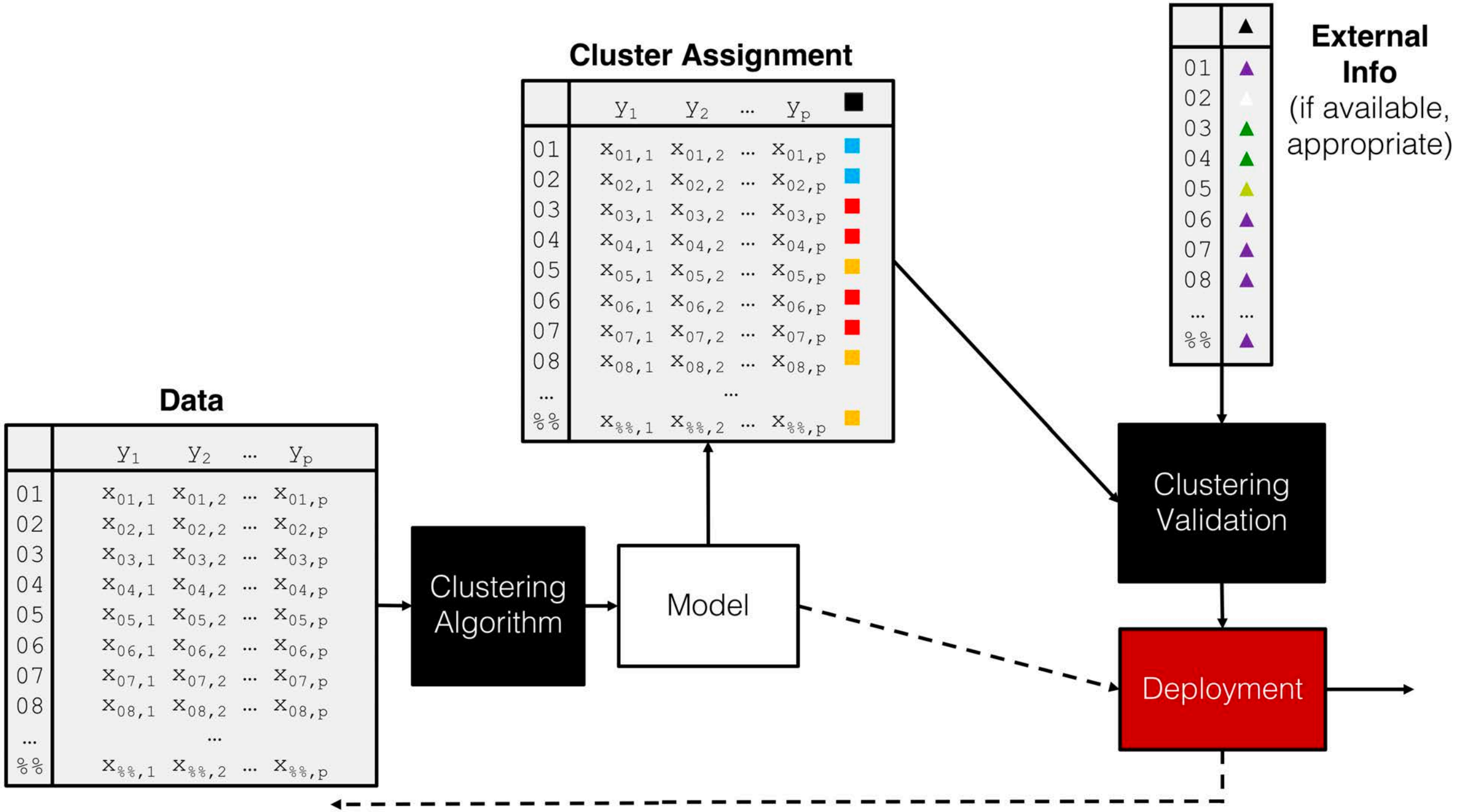
Other Uses

Dividing a larger group (or area, or category) into **smaller** groups, with members of the smaller groups guaranteed to have similarities of some kind.

- tasks may then be solved separately for each of the smaller groups
- this may lead to increased accuracy once the separate results are aggregated

Creating (new) taxonomies **on the fly**, as new items are added to a group of items

- this would allow for easier product navigation on a website like Netflix, for instance.


Cluster Assignment
External Info
(if available, appropriate)
Data
Clustering Algorithm
Model
Clustering Validation
Deployment

Clustering Schemes

k-Means

- classical (and over-used) model
- assumptions made about the shape of clusters

Hierarchical Clustering

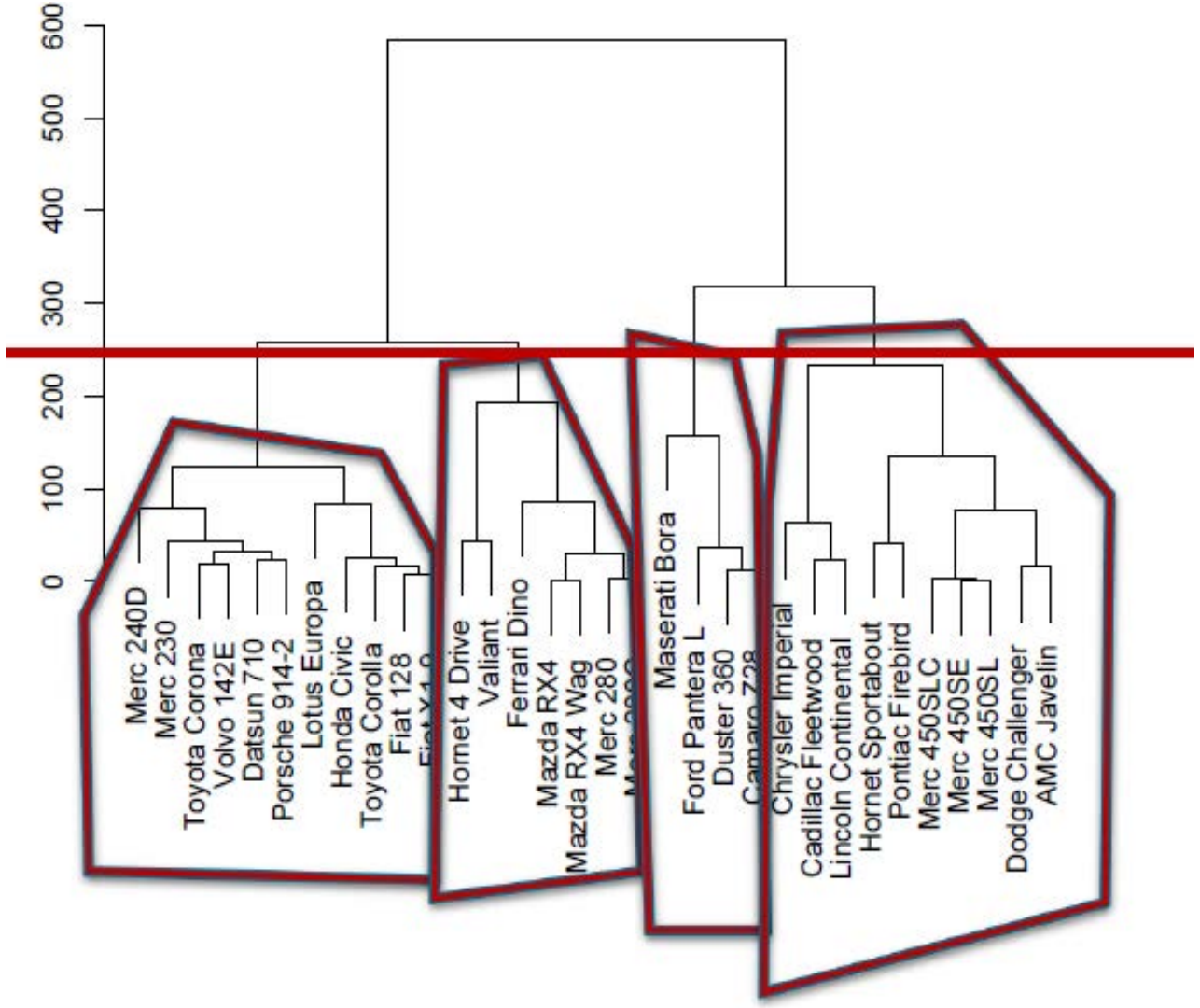
- easy to interpret, deterministic

Latent Dirichlet Allocation

- used for topic modeling

Expectation Maximization

Hierarchical Clustering



Clustering Schemes

Balanced Iterative Reducing and Clustering using Hierarchies

- aka BIRCH

Density-Based Spatial Clustering of Applications with Noise

- graph-based

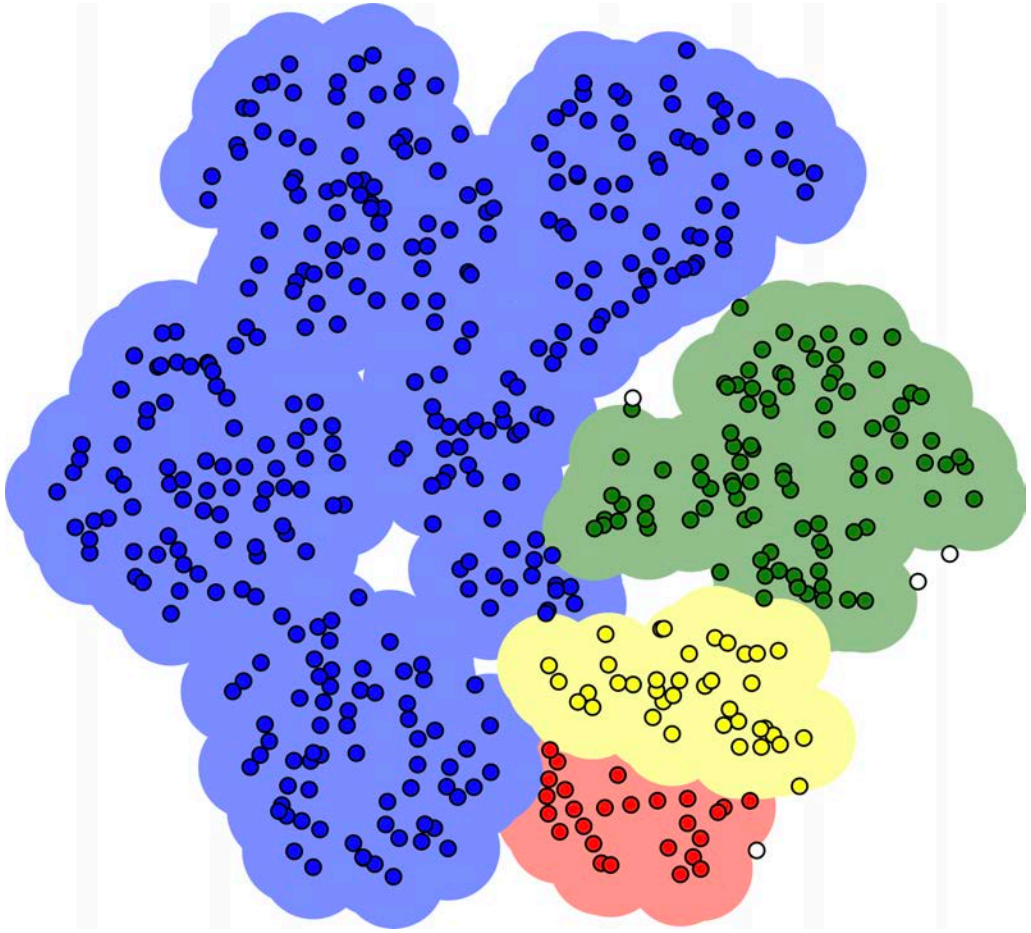
Affinity Propagation

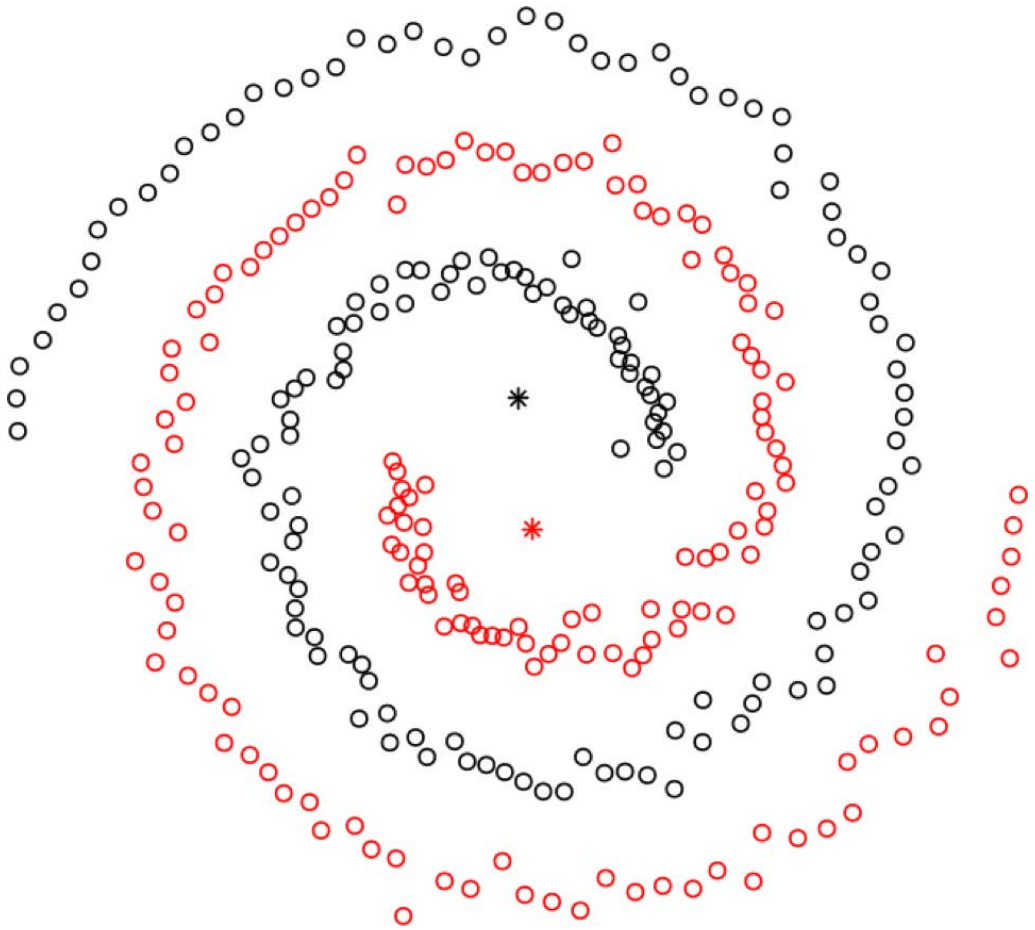
- selects the optimal number of clusters automatically

Spectral Clustering

- recognizes non-blob clusters

DBSCAN and Spectral Clustering



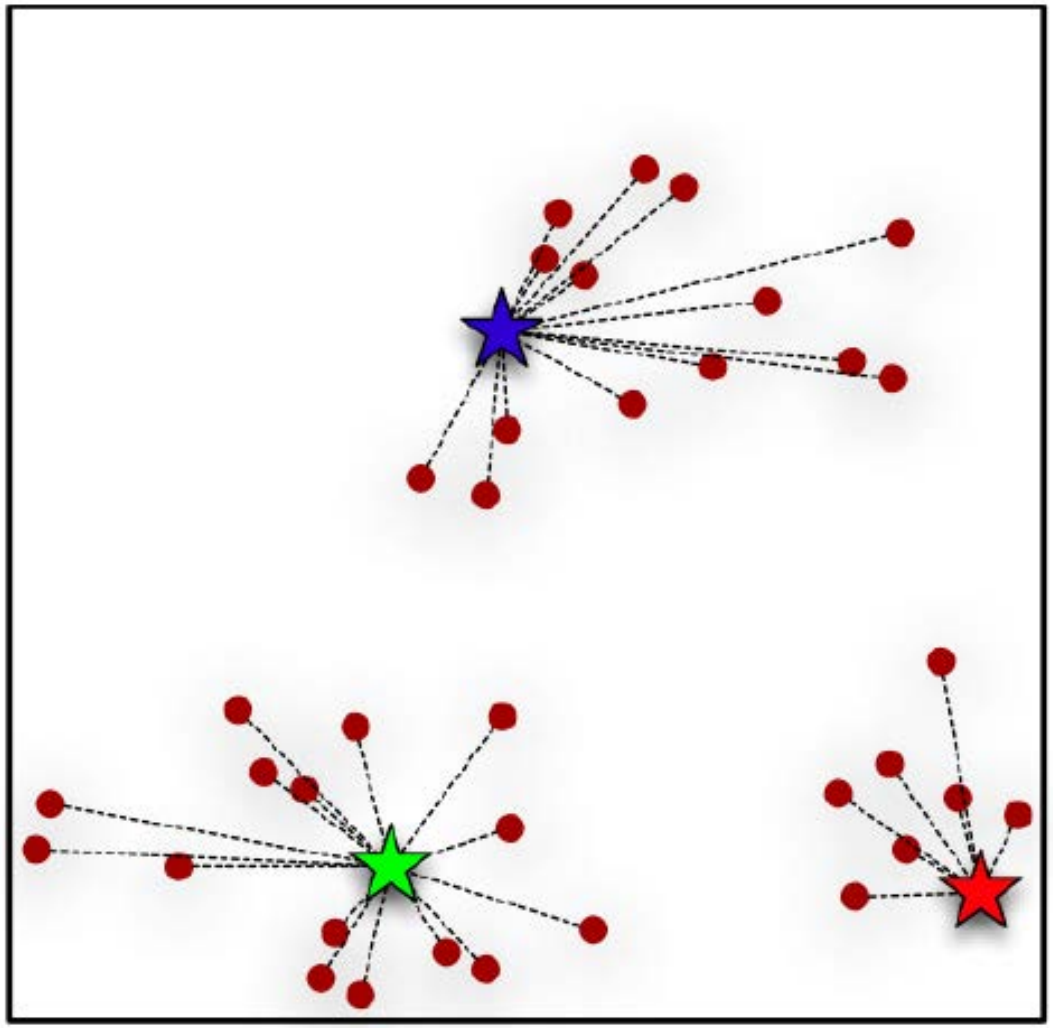


k-means is well-adapted to numerical data (although it can also be used for categorical data), but it has a tendency to force clusters of **roughly equivalent sizes**.

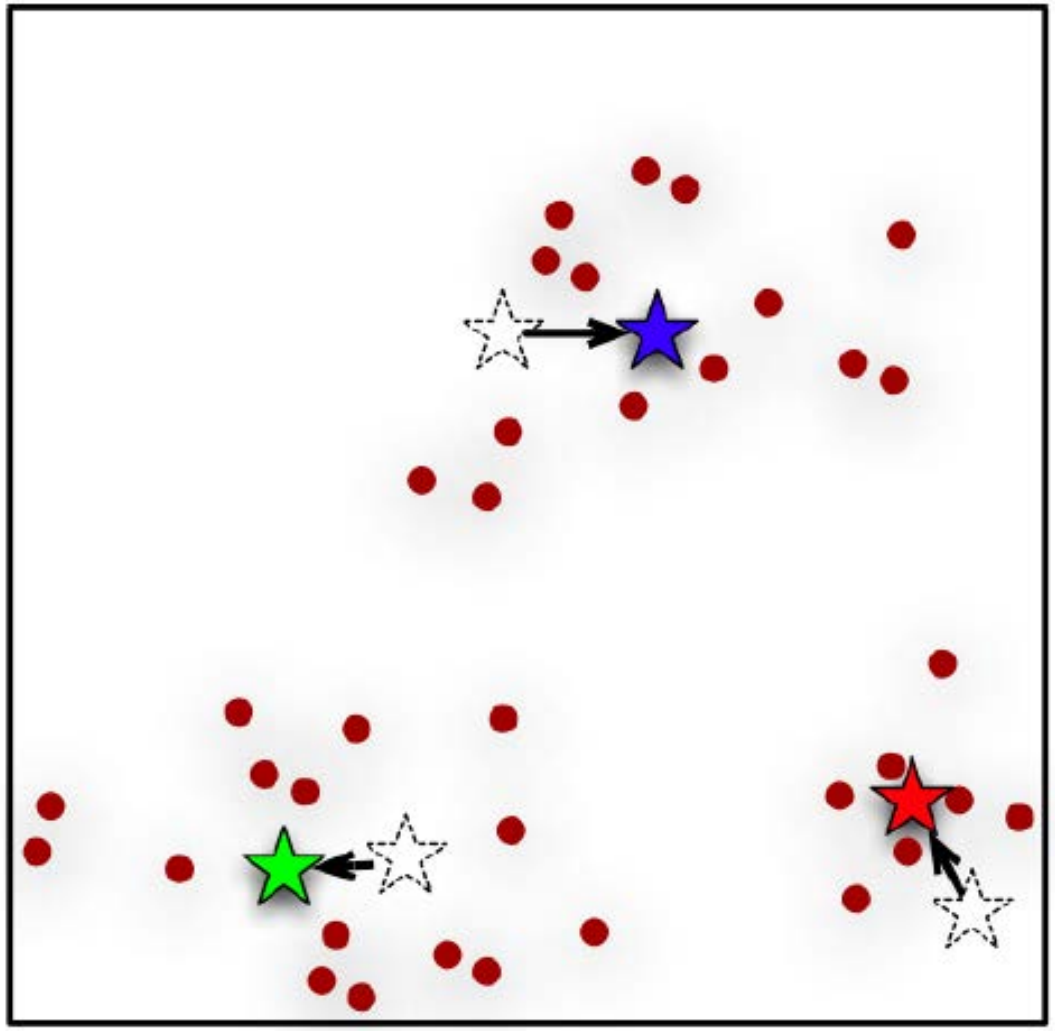
k-Means Algorithm

1. Select the desired **number of clusters**, say k
2. Randomly choose k instances as initial **cluster centres**
3. Calculate the **distance** from each observation to each centre
4. Place each instance in the cluster whose centre it is **nearest** to
5. Compute the **centroid** for each cluster
6. Repeat steps 3 – 5 with the new centroids
7. Repeat step 6 until the clusters are **stable**

[Provost & Fawcett, *Data Science for Business*]

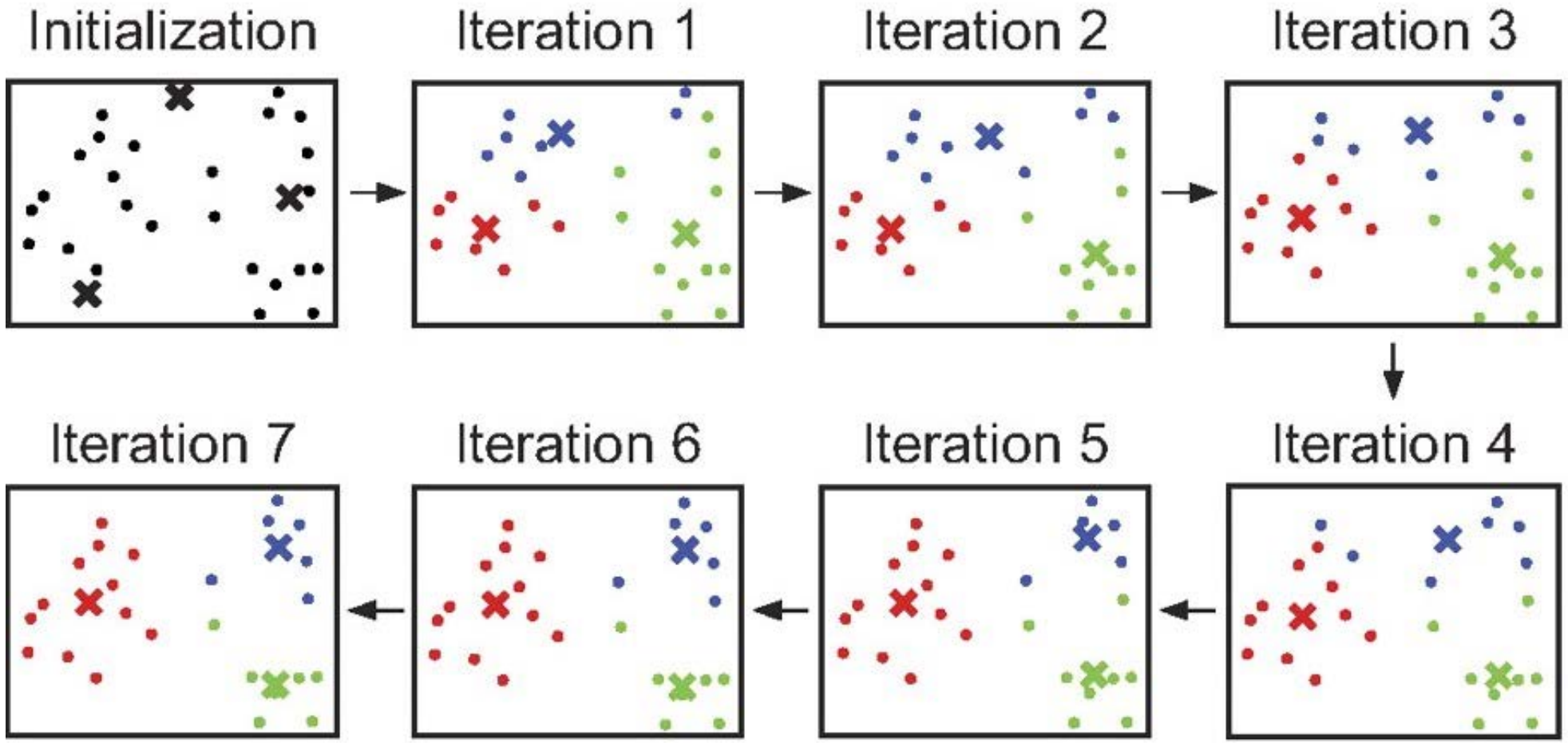


Cluster Allocation



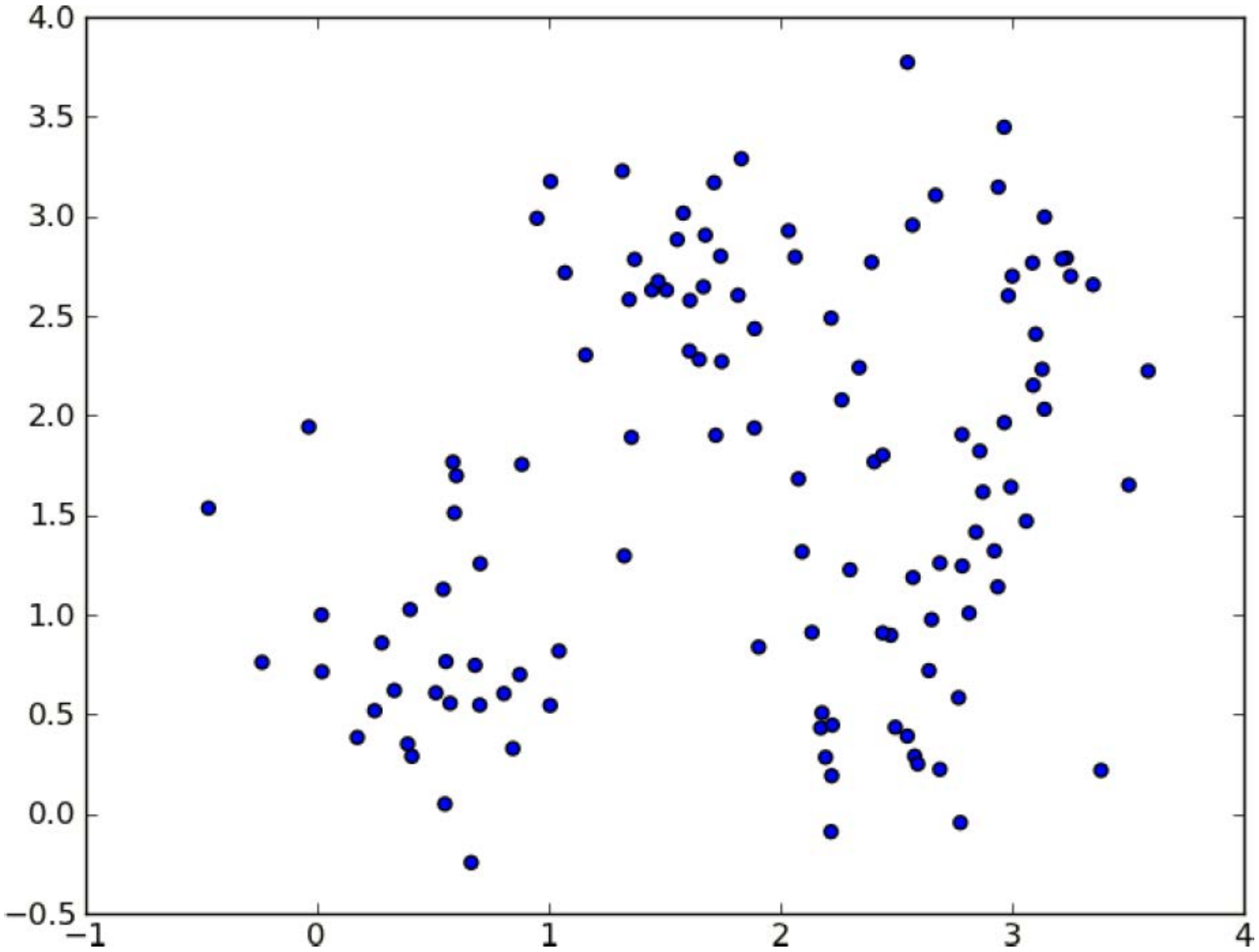
New Cluster Centroids

[Ozden, Lee, Sullivan, Wang, *Identification and Clustering of Event Patterns From In Vivo Multiphoton Optical Recordings of Neuronal Ensembles*]



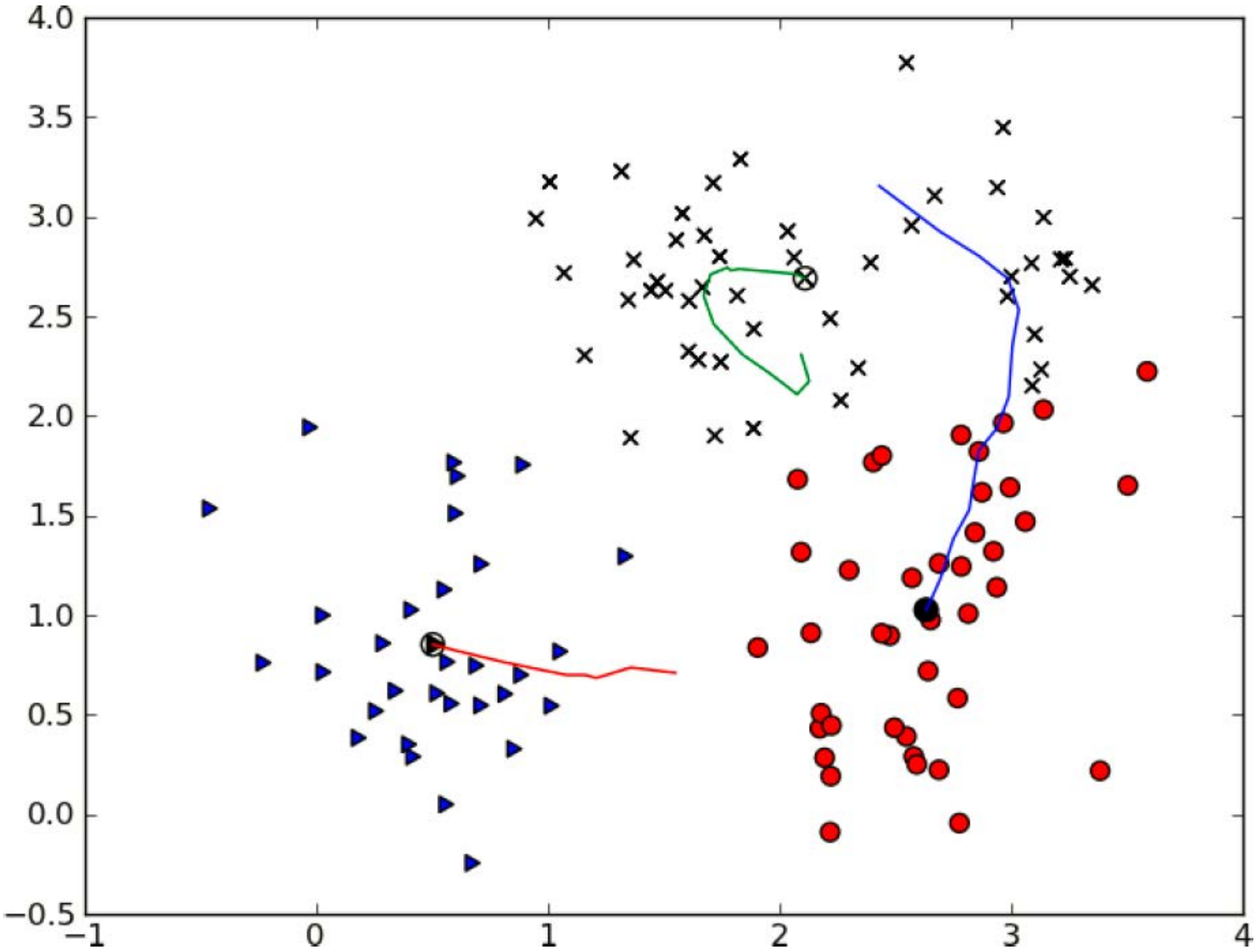
***k*-Means Iterative Process**

[Provost & Fawcett, *Data Science for Business*]



Movement of Centroids

[Provost & Fawcett, *Data Science for Business*]



Movement of Centroids

k-Means Strengths

Easy to implement (without having to actually compute pairwise distances).

- extremely common as a consequence
- elegant and simple

In many contexts, k-means is a **natural** way to look at grouping observations.

Helps provide a **basic understanding of the data structure** in a first pass.

k-Means Limitations

Data points can only be assigned to **one** cluster.

- this can lead to overfitting
- robust solution: consider the **probability** of belonging to each cluster

Underlying clusters are assumed to be **blob-shaped**

- k-means will fail to produce useful clusters if that assumption is not met in practice

Clusters are assumed to be separate (discrete)

- k-means does not allow for **overlapping** or **hierarchical** groupings

Distance Measures (Metrics)

Categorical Variables*

- Hamming distance
- Russel/Rao index
- Jaccard
- Matching coefficient
- Dice's coefficient
- etc.

Numerical Variables

- Euclidean
- Manhattan
- Correlation
- Cosine
- Pearson
- etc.

No steadfast rule to determine which distance to use in k-means

Competing schemes are often produced using different metrics.

* may need to be dichotomized

Take-Away: with mixed data,

Hamming $\longleftrightarrow$ Euclidean, Jaccard $\longleftrightarrow$ Manhattan.

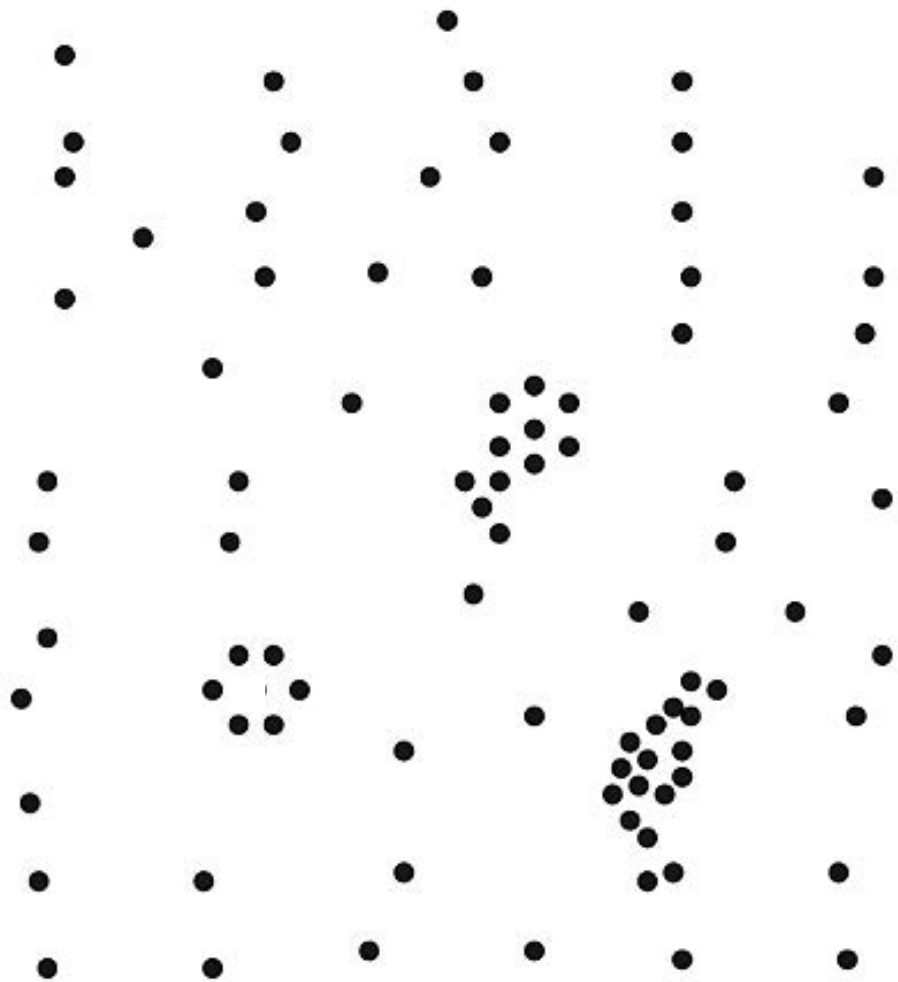
Clustering Challenges

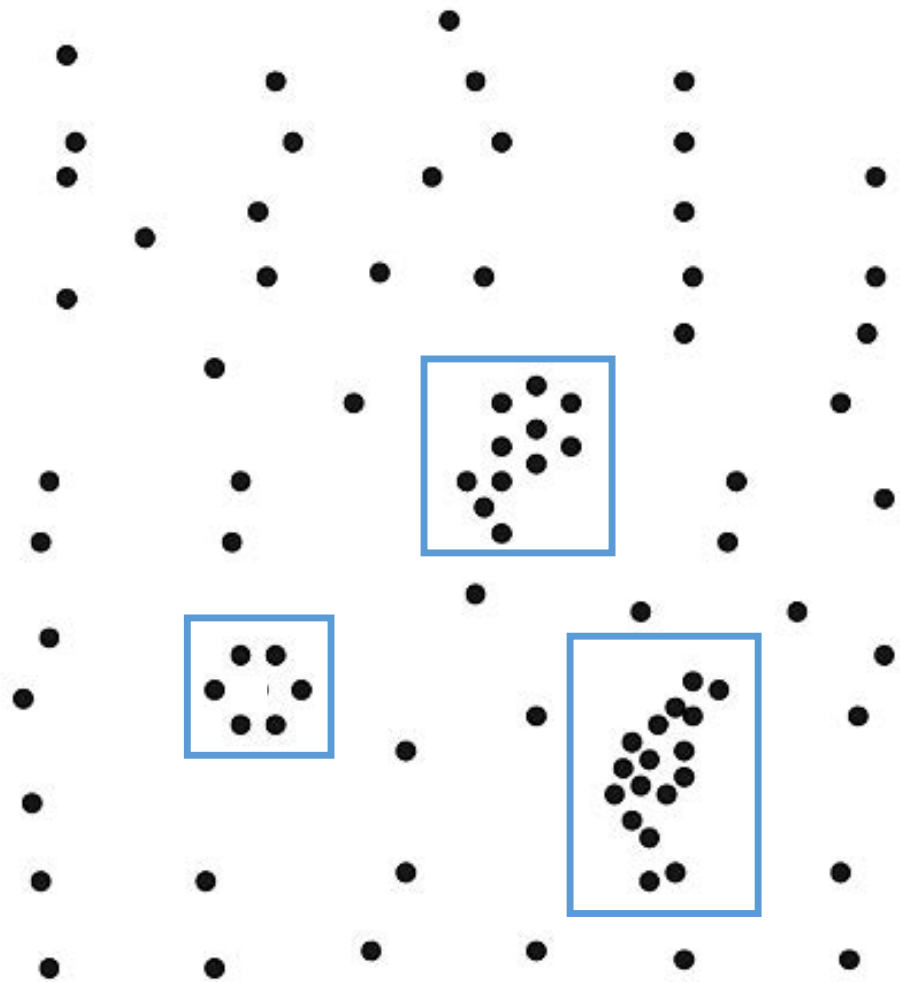
Automation
relatively intuitive for humans, but hard to automate

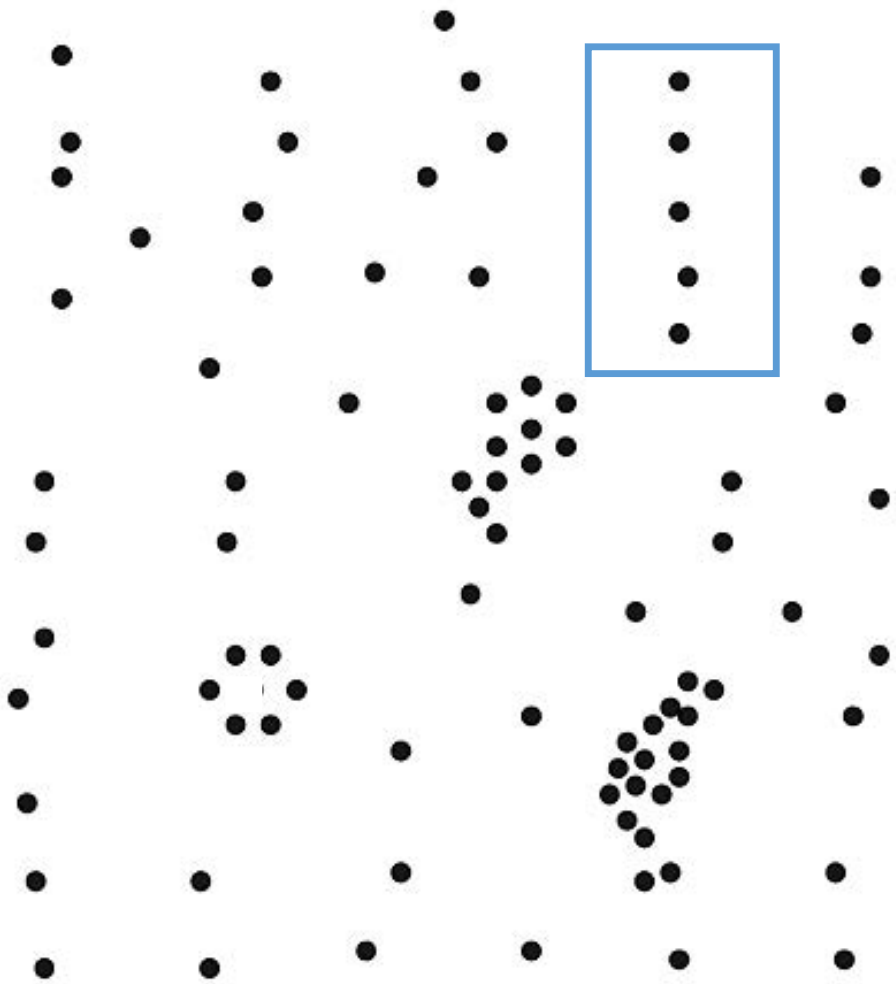
Lack of a clear-cut definition
no universal agreement as to what constitutes a cluster

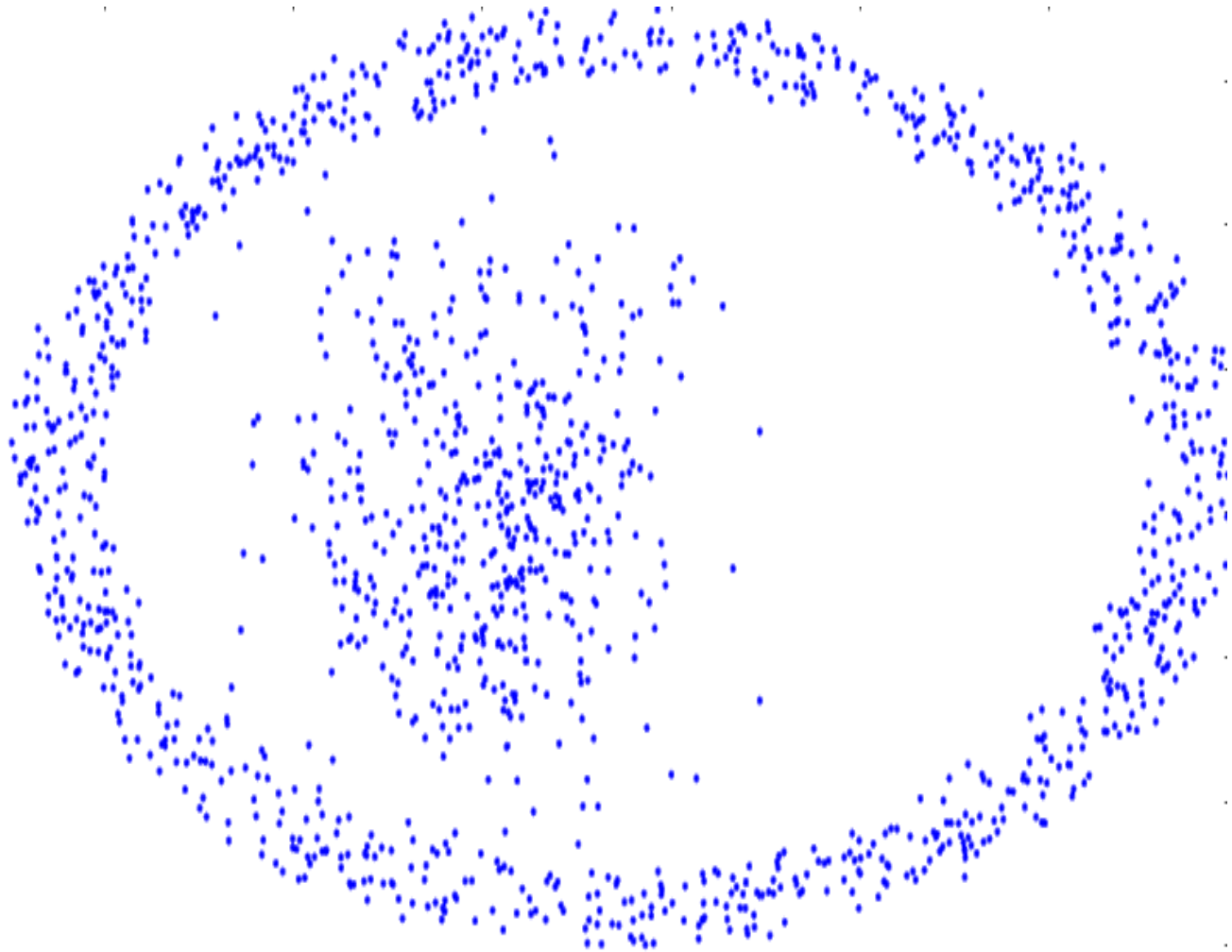
Lack of repeatability
non-deterministic: the same algorithm, applied twice to the same dataset can discover completely different clusters

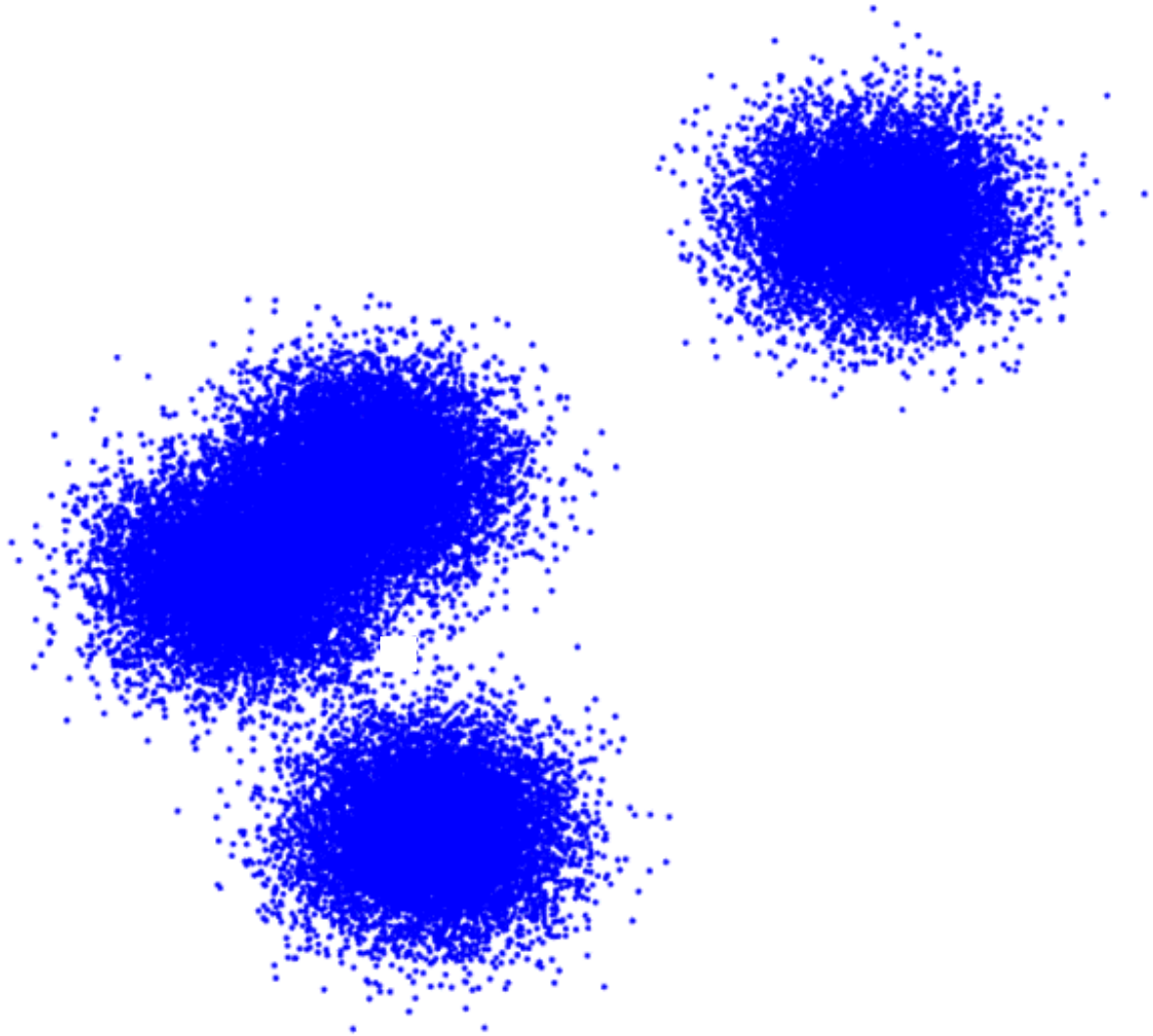
Number of clusters
optimal number of clusters difficult to determine



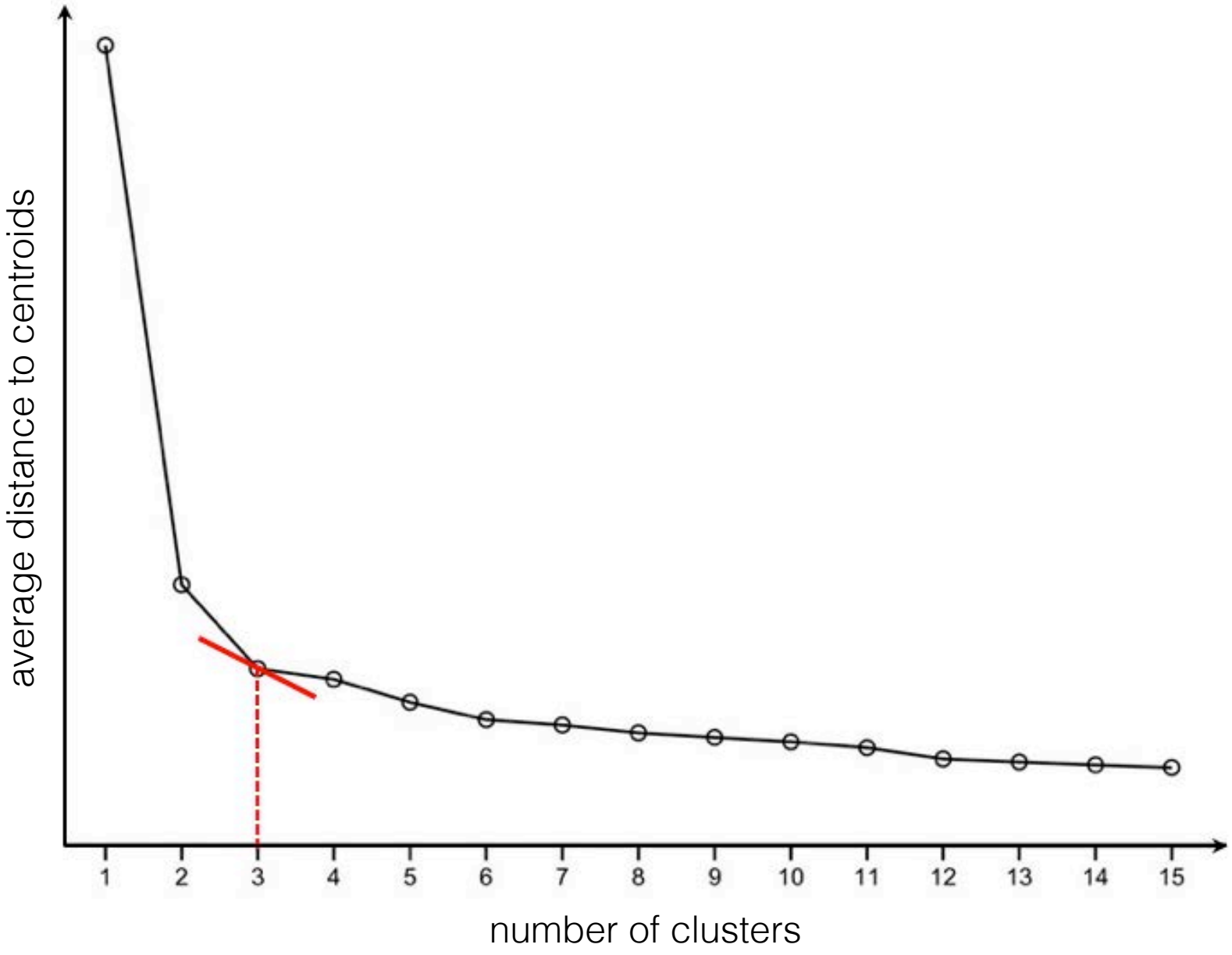








[A. Ng, K. Soo, *Numsense!*]



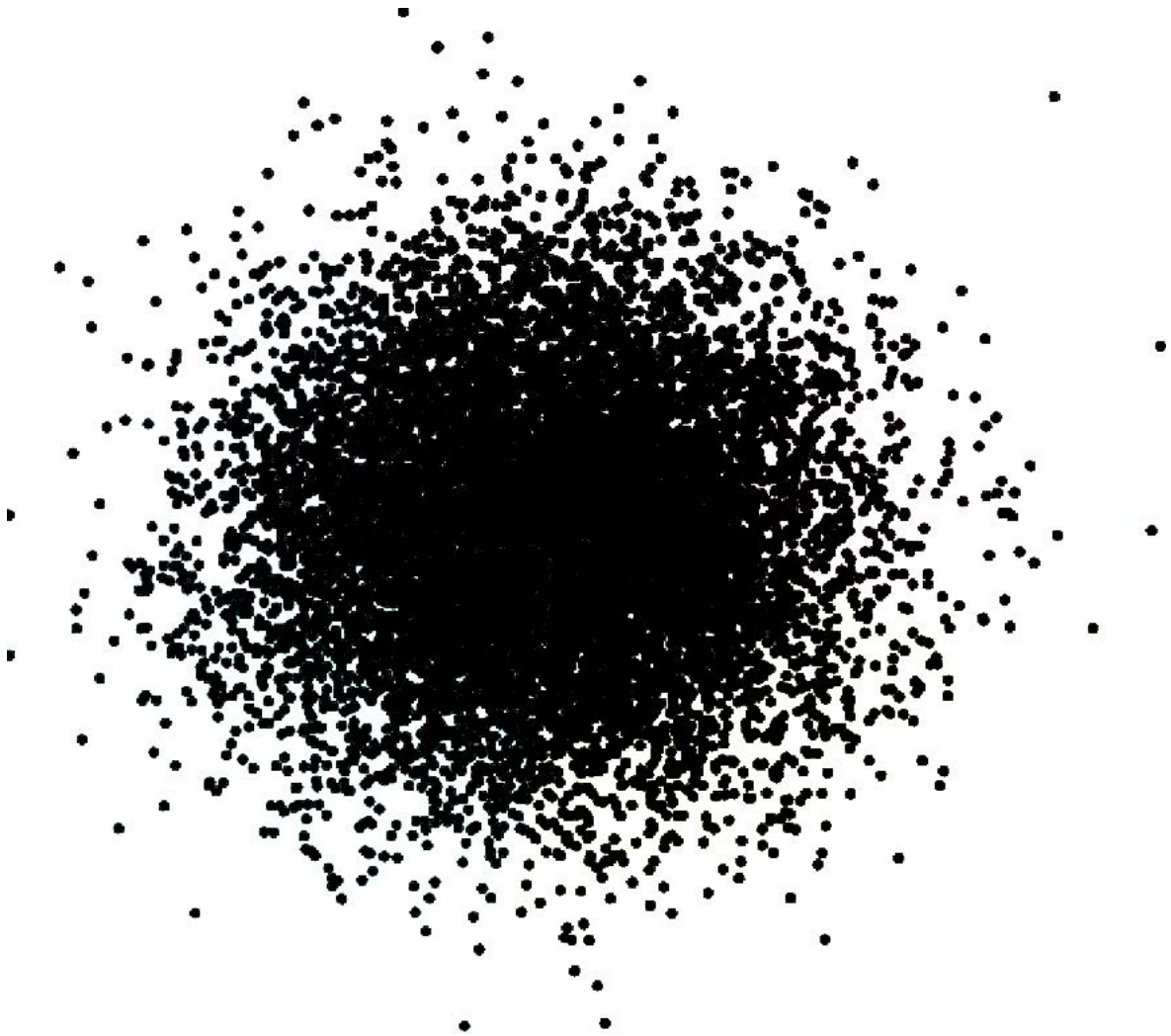
Clustering Challenges

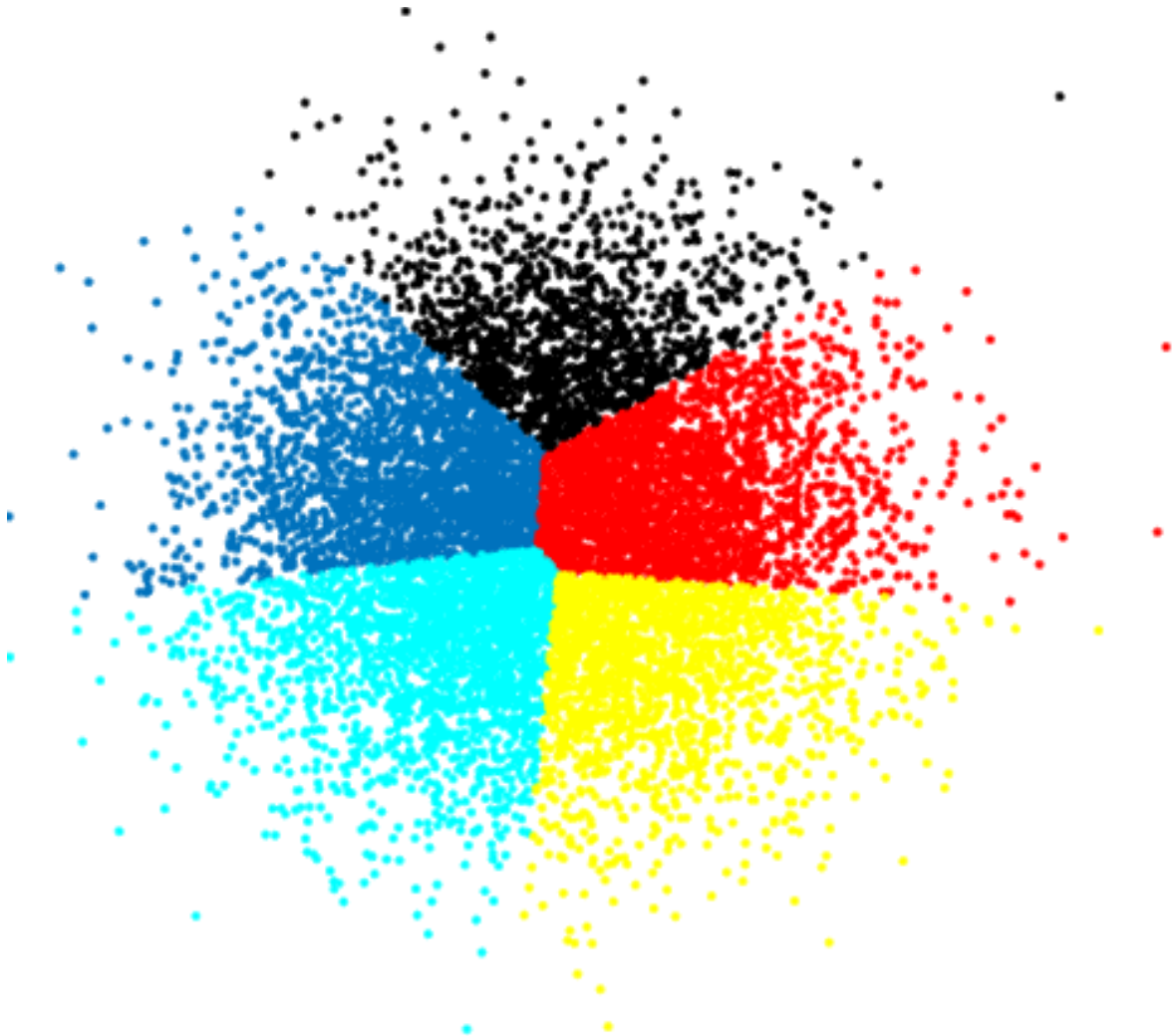
Cluster description
should clusters be described using representative instances or average values?

Model validation
no true clustering information against which to contrast the clustering scheme, so how do we determine if it is appropriate?

Ghost clustering
most methods will find clusters even if there are none in the data

***A postiori* rationalization**
once clusters have been found, it is tempting to try to "explain" them …





Take-Away: clustering looks easy in 2D or 3D spaces… but in high-dimensional spaces, **almost all pairs are equidistant!**

Data science students don’t have to be gardeners, but it helps.

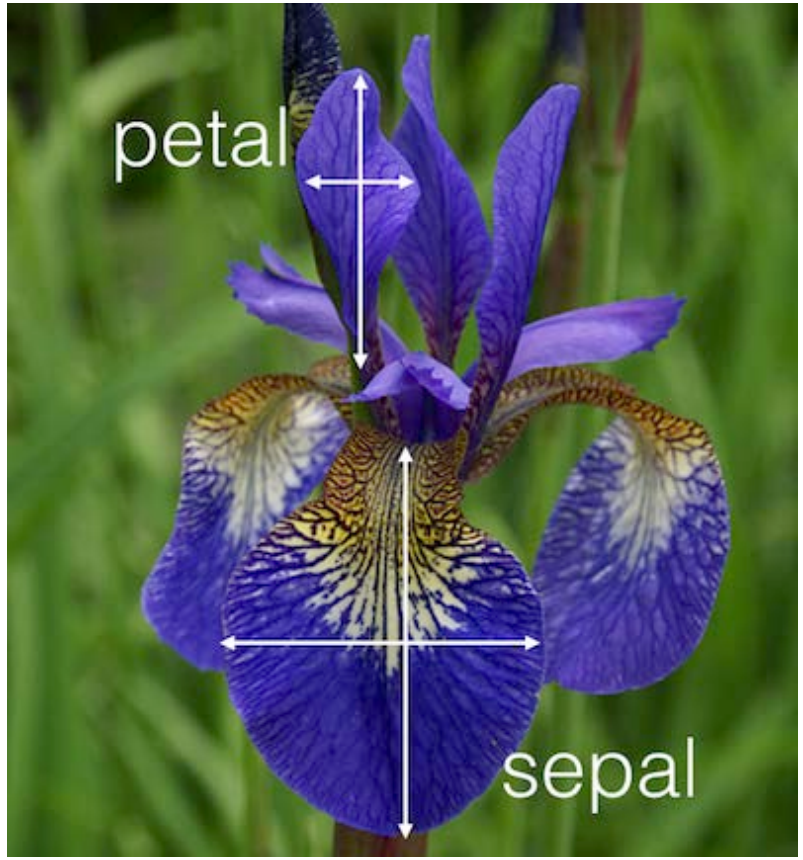
(unknown)

Example – Iris Dataset

Iris is a genus of plants with showy flowers.

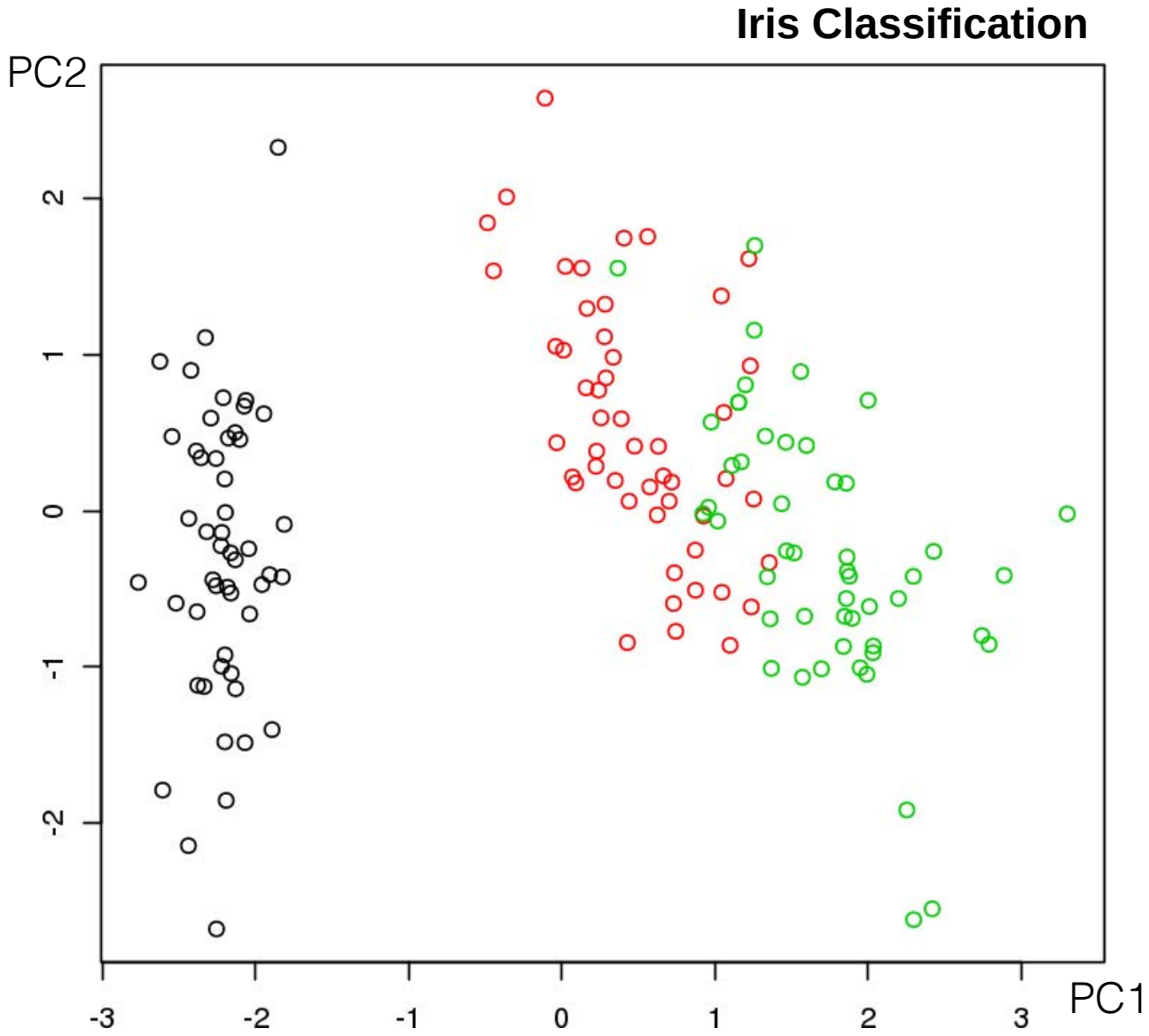
Fisher's iris dataset contains 150 observations of 5 attributes for specimens collected by Anderson, mostly from a Gaspé peninsula's pasture in the 1930s:

- **petal width**
- **petal length**
- **sepal width**
- **sepal length**
- **species**

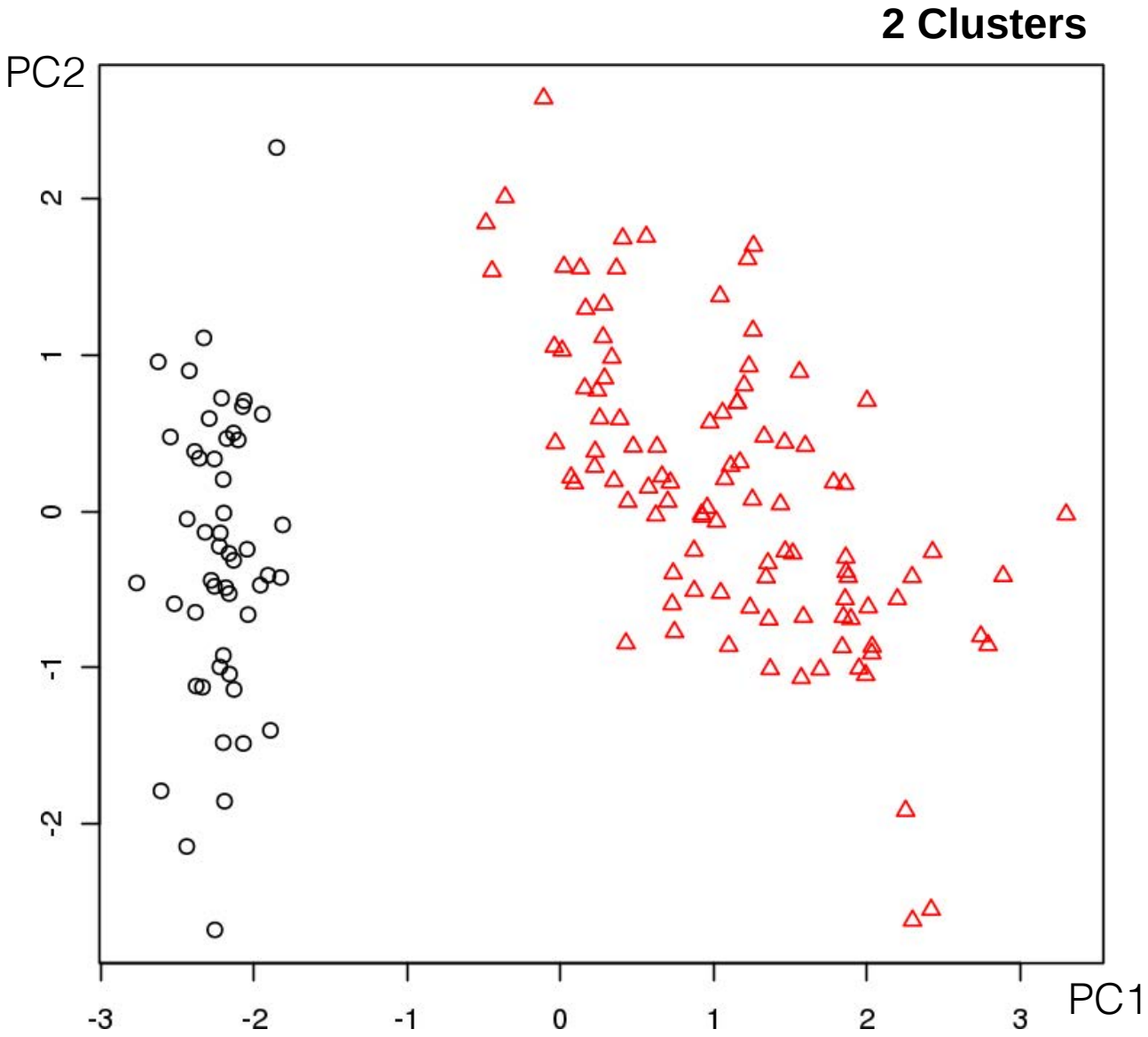


[http://blog.kaggle.com/wp-content/uploads/2015/04/iris_petal_sepal.png]

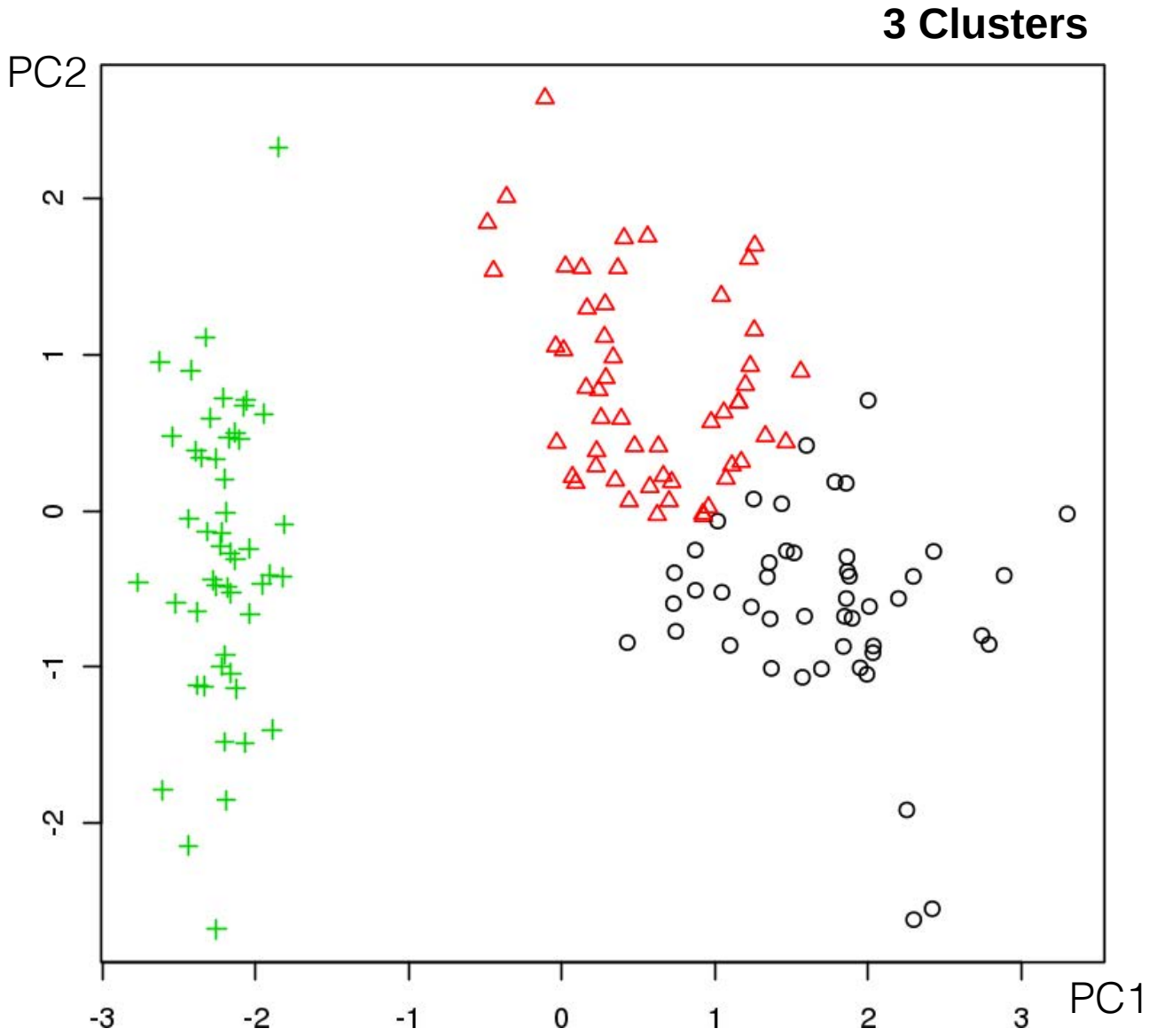
Example – Iris Dataset



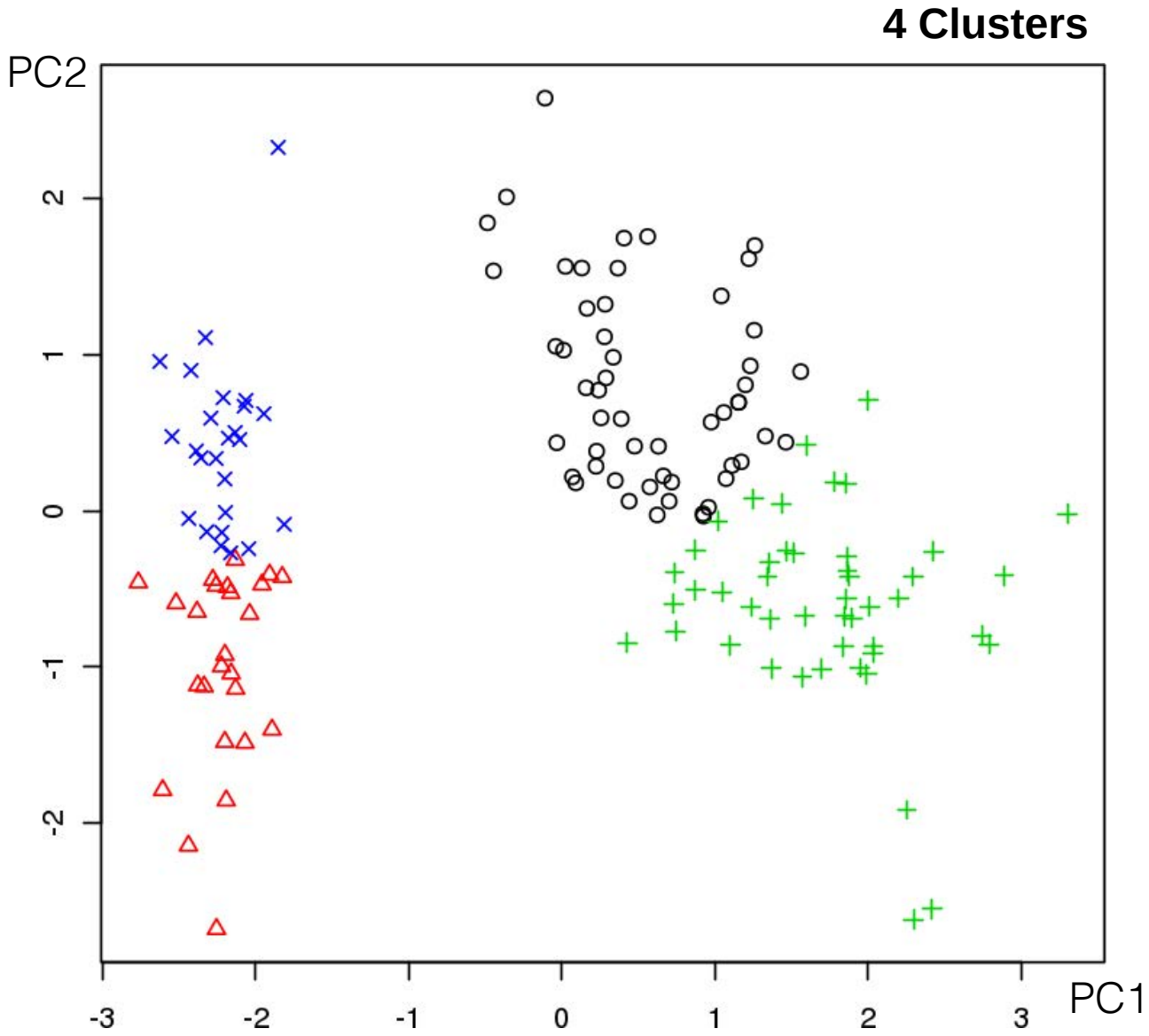
Example – Iris Dataset



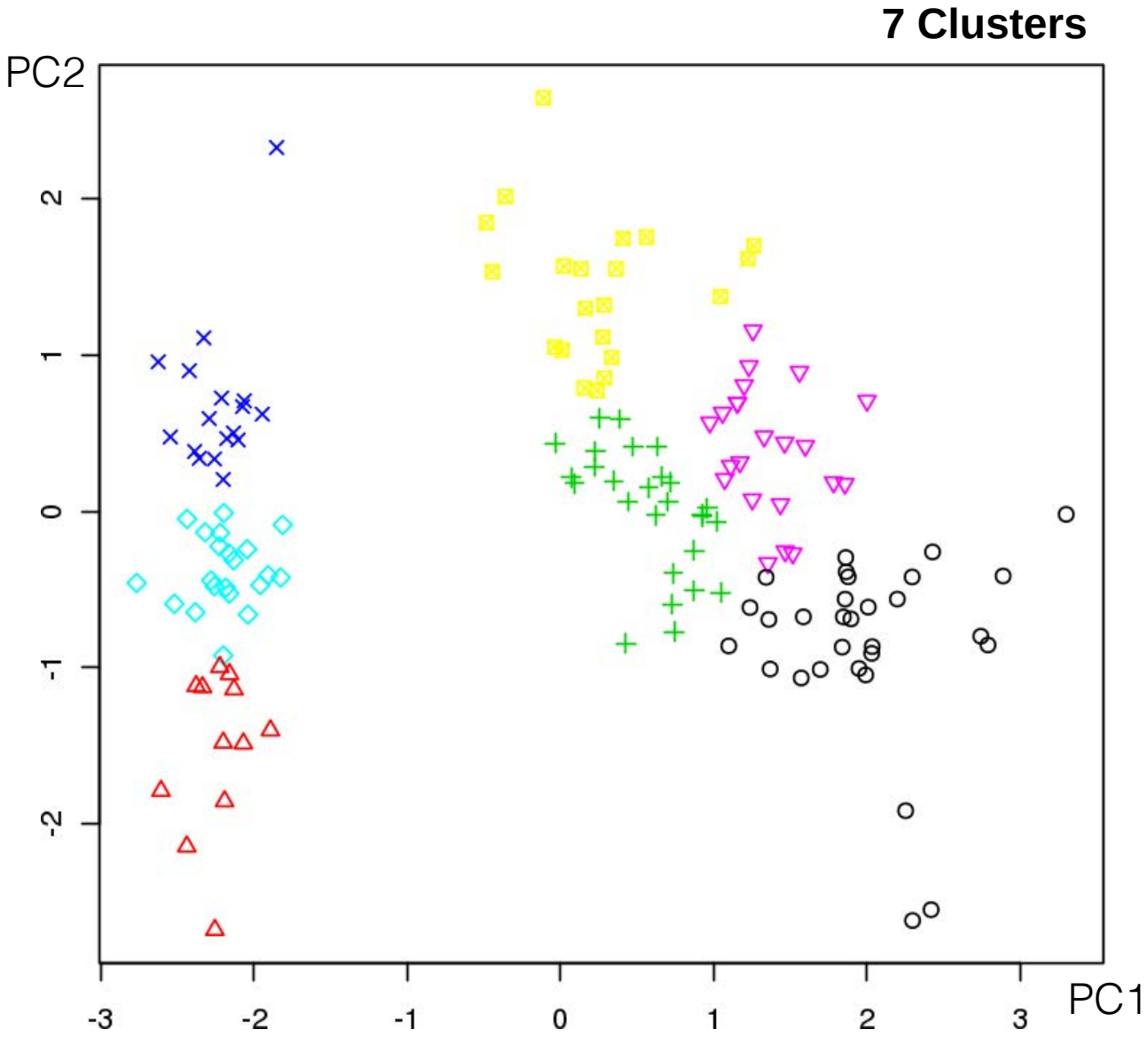
Example – Iris Dataset



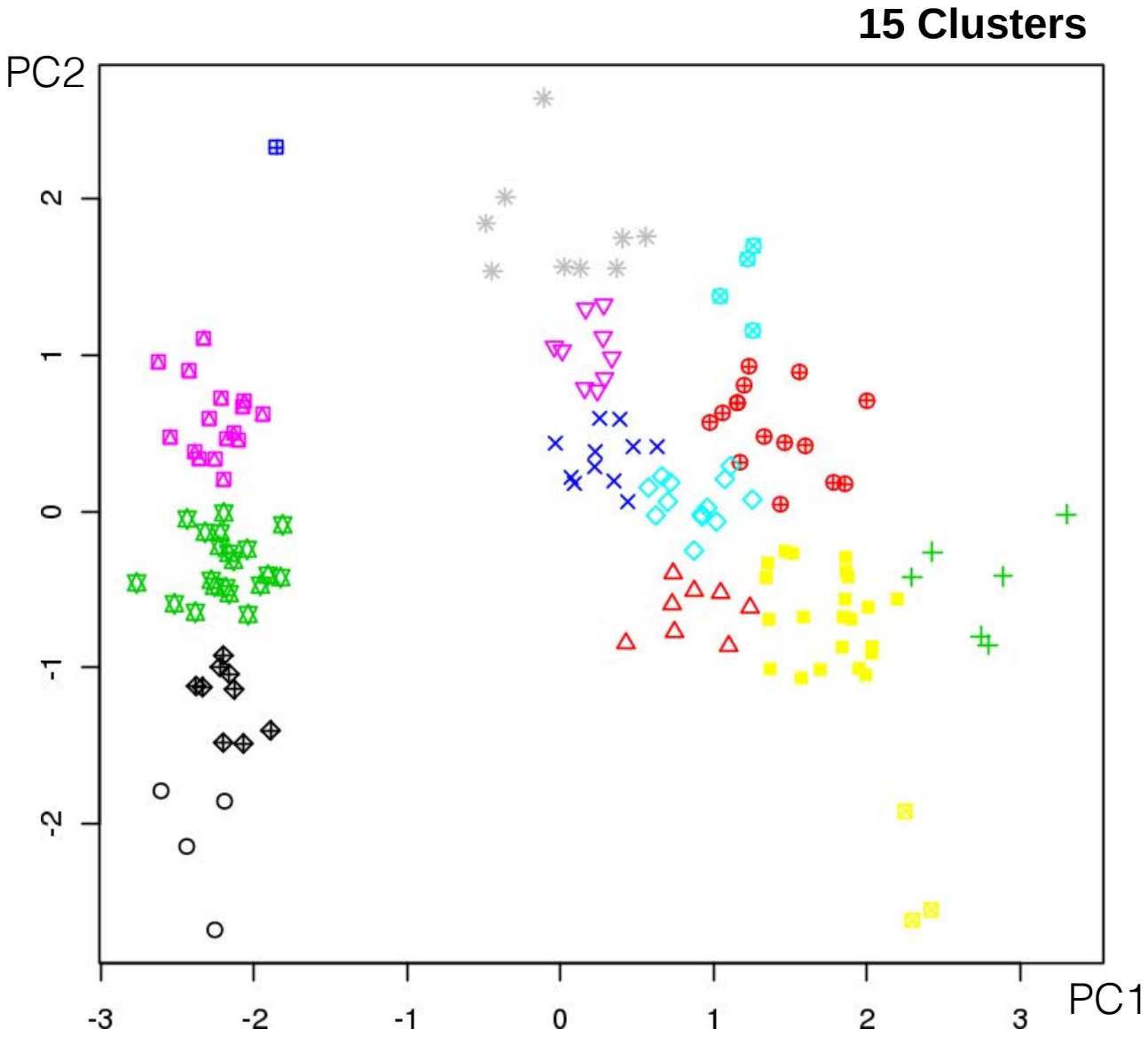
Example – Iris Dataset



Example – Iris Dataset



Example – Iris Dataset



Clustering Validation

What does it mean for a clustering scheme to be **better** than another?

What does it mean for a clustering scheme to be **valid**?

What does it mean for a single cluster to be good?

How many clusters are there in the data, really?

Main challenge: what are we comparing the clustering scheme **against**? (versions of this problem plague unsupervised tasks)

Take-Away: right vs. wrong/good vs. bad is meaningless.

Optimal vs. sub-optimal is the way to go.

Clustering Validation

Optimal clustering scheme:

- maximal separation between clusters
- maximal similarity within groups
- agrees with human eye test
- useful at achieving its goals

Validation types

- external (uses additional information)
- internal (uses only the clustering results)
- relative (compares across clustering attempts)

Internal Clustering Validation

Davies-Bouldin Index can be used to determine the number of clusters in k-means

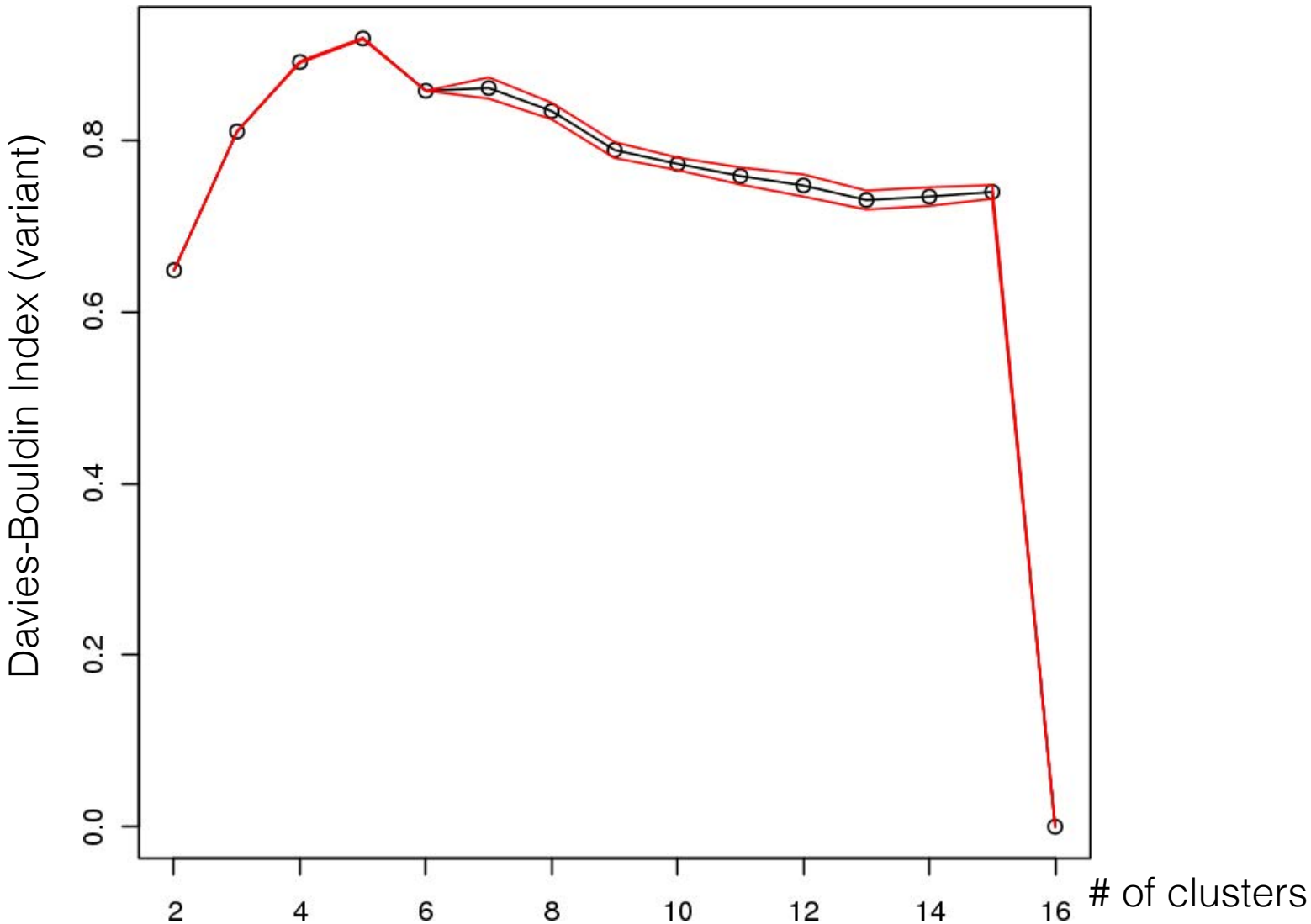
$$DB = \frac{1}{N}\sum_{i=1}^{N} \max_{j \neq i} \frac{s_i + s_j}{d(c_i, c_j)},$$

where N is the number of clusters, c_m is the centroid of the mth cluster, and s_m is the average distance of the points in the mth cluster to c_m

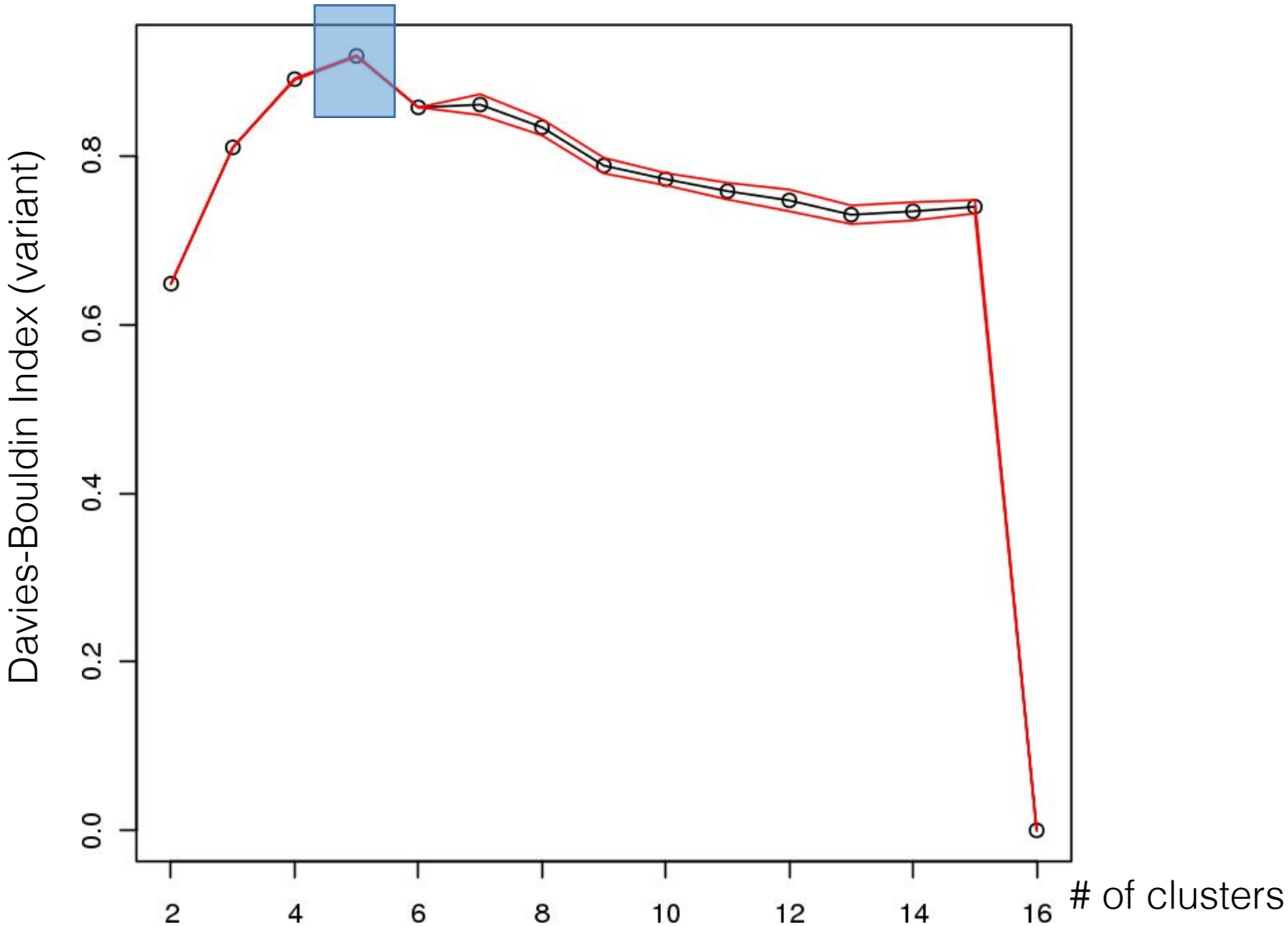
Other Methods

- Sum of Squared Errors
- Dunn's Index
- Silhouette Metric
- etc.

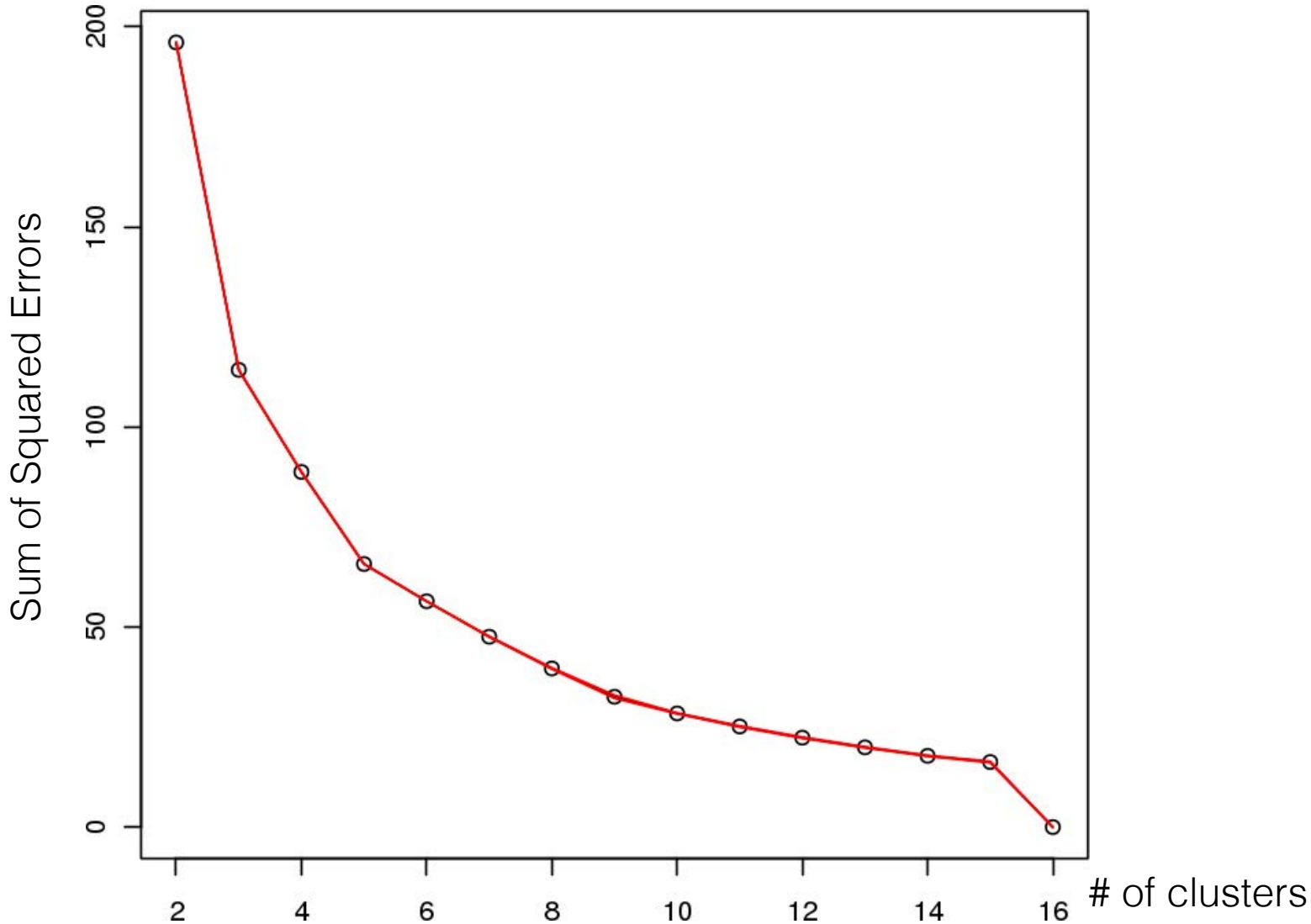
Example – Iris Dataset (revisited)



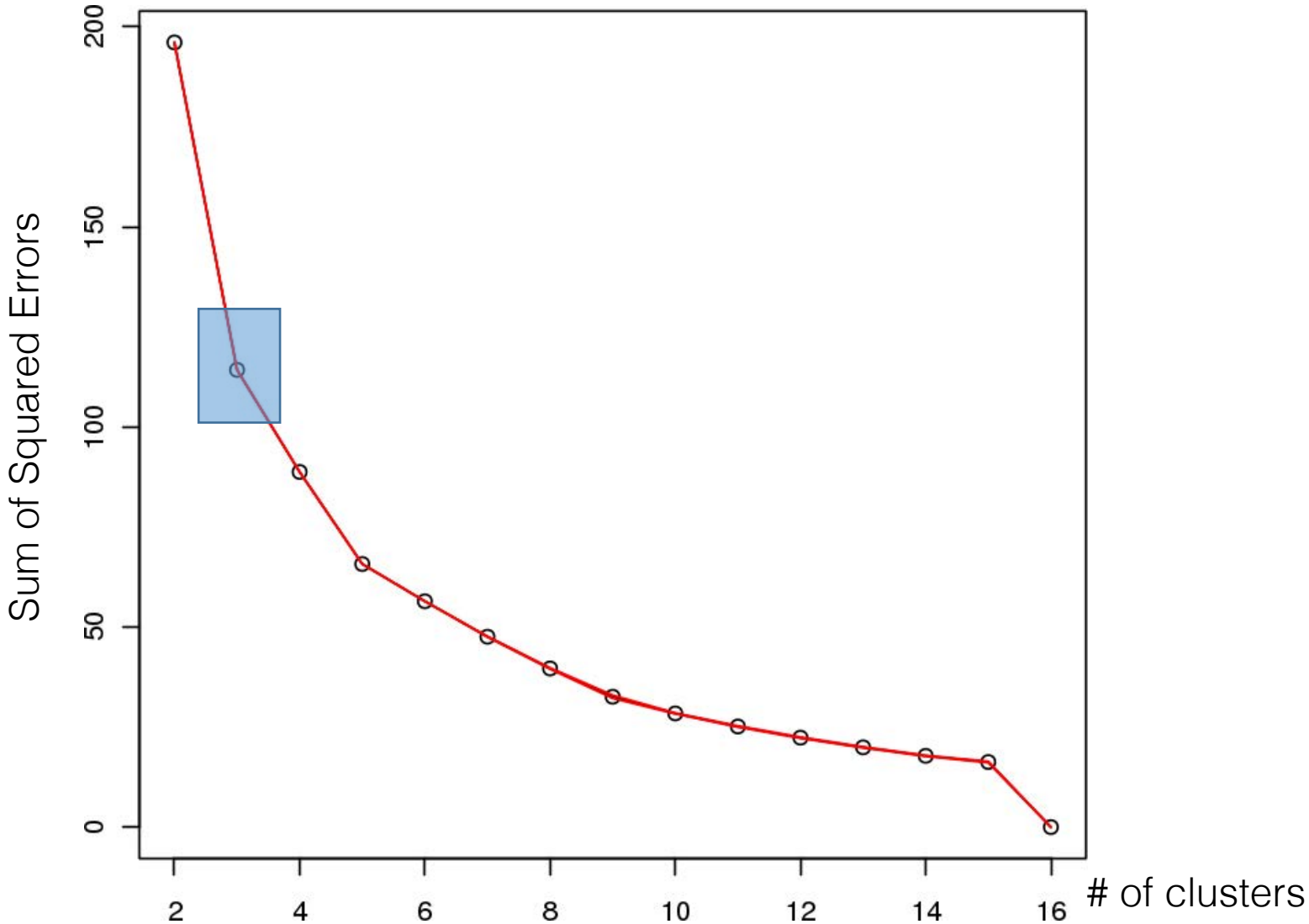
Example – Iris Dataset (revisited)



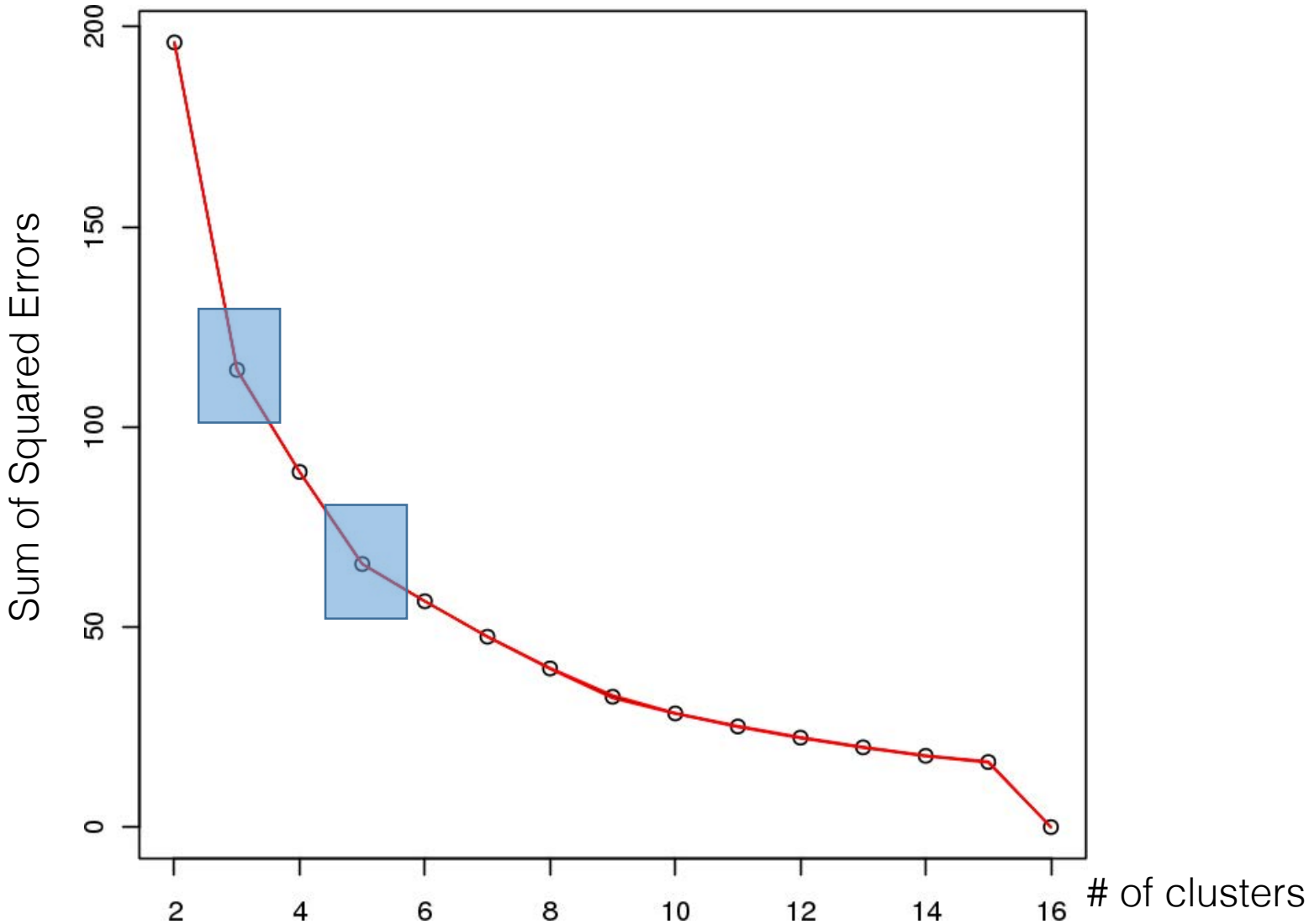
Example – Iris Dataset (revisited)



Example – Iris Dataset (revisited)

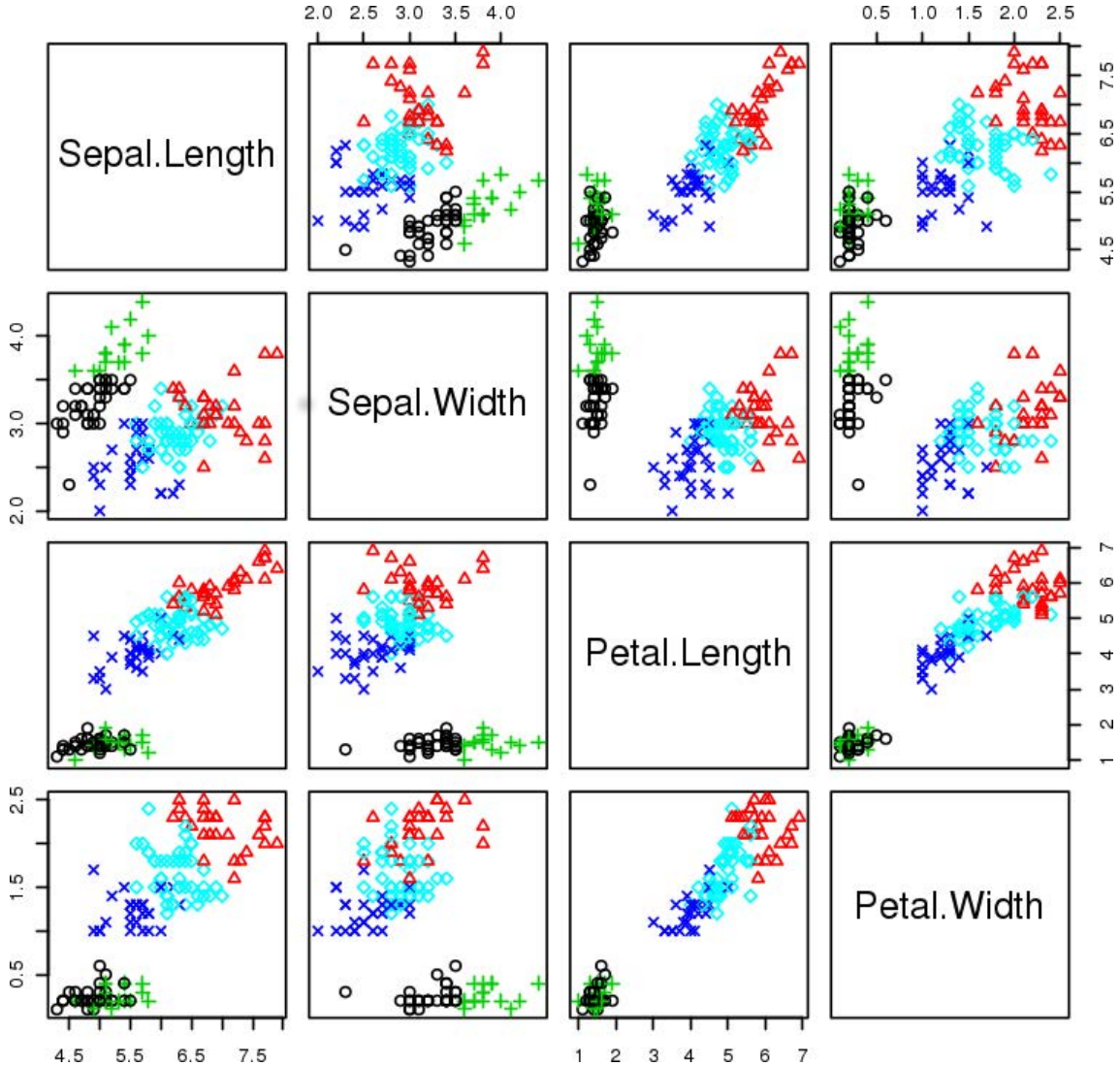


Example – Iris Dataset (revisited)



Example – Iris Dataset (revisited)

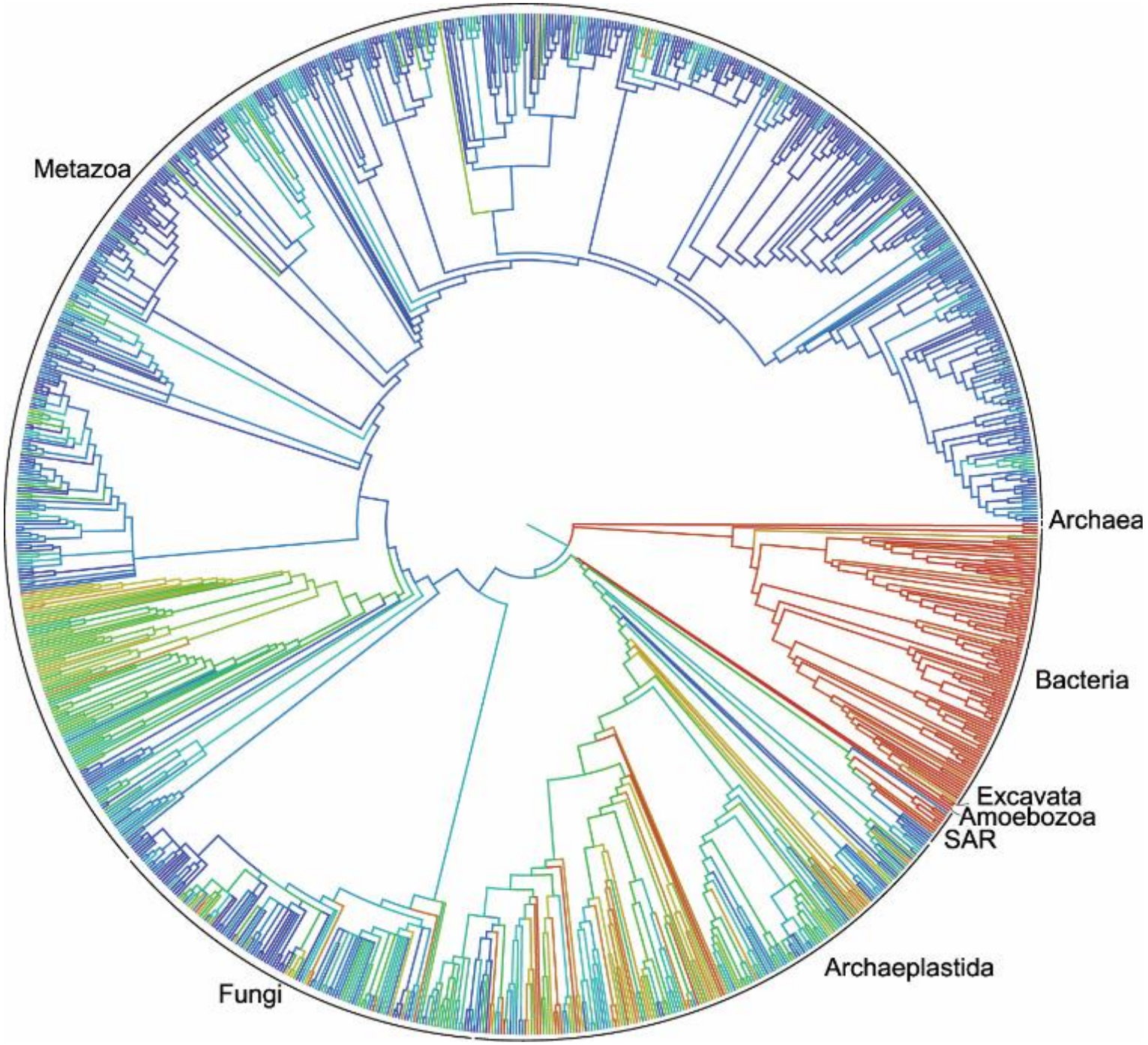
5 clusters (1 replicate)



Take-Away: validating clusters is just as complicated as defining clusters.

We'll have more to say on the topic.

Hierarchical Clustering



[https://phys.org/news/2015-09-tree-life-million-species.html]

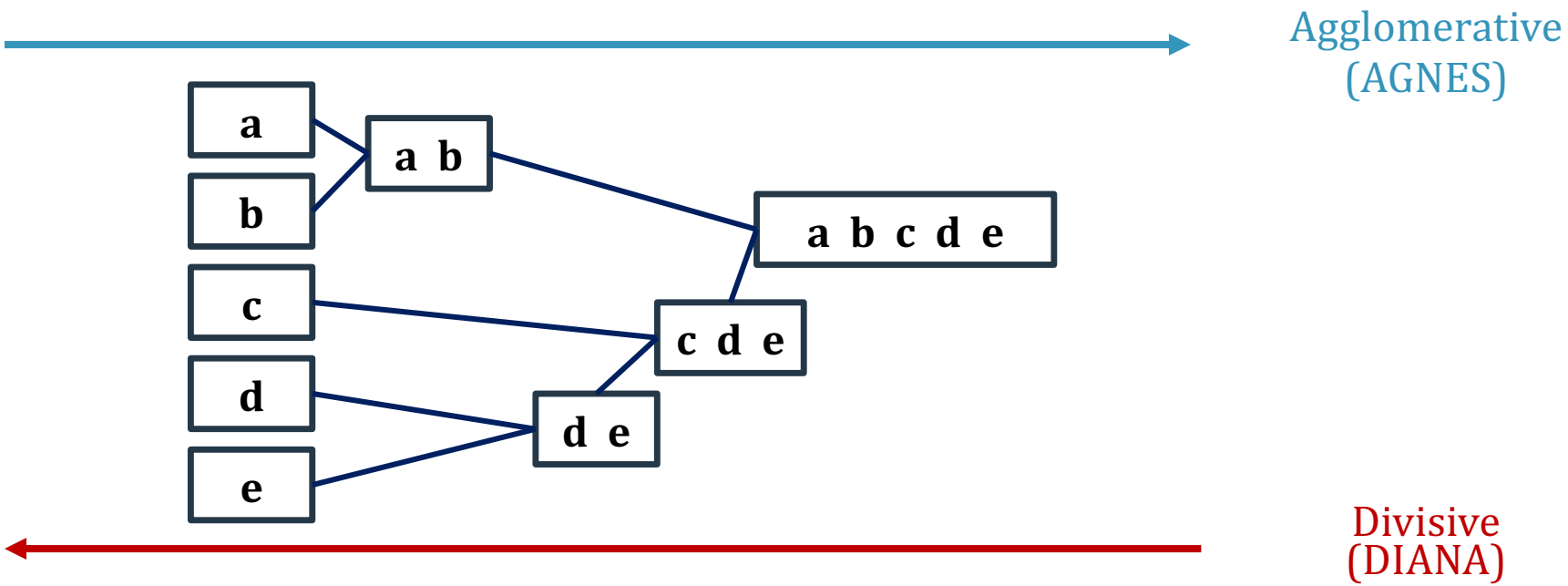
Hierarchical Clustering Overview

Hierarchical clustering (HC) clusters a dataset into a **hierarchy** of clusters (order relation is set containment).

There are two main strategies:

- **Bottom-up** (agglomerative)
  initially, each observation starts in its own **separate** cluster
  clusters are **merged** as the hierarchy is climbed
  after the last merge, all observations are in the **same** cluster
- **Top-down** (divisive)
  initially, each observation starts in the **same** cluster
  clusters are **split** as the hierarchy is descended down
  after the last split, each observation ends in its own **separate** cluster

Bottom-up HC is significantly faster than Top-down HC (poly. vs. exp.)


Agglomerative
(AGNES)
a
b
c
d
e
a b
d e
c d e
a b c d e
Divisive
(DIANA)

Hierarchical Clustering Overview

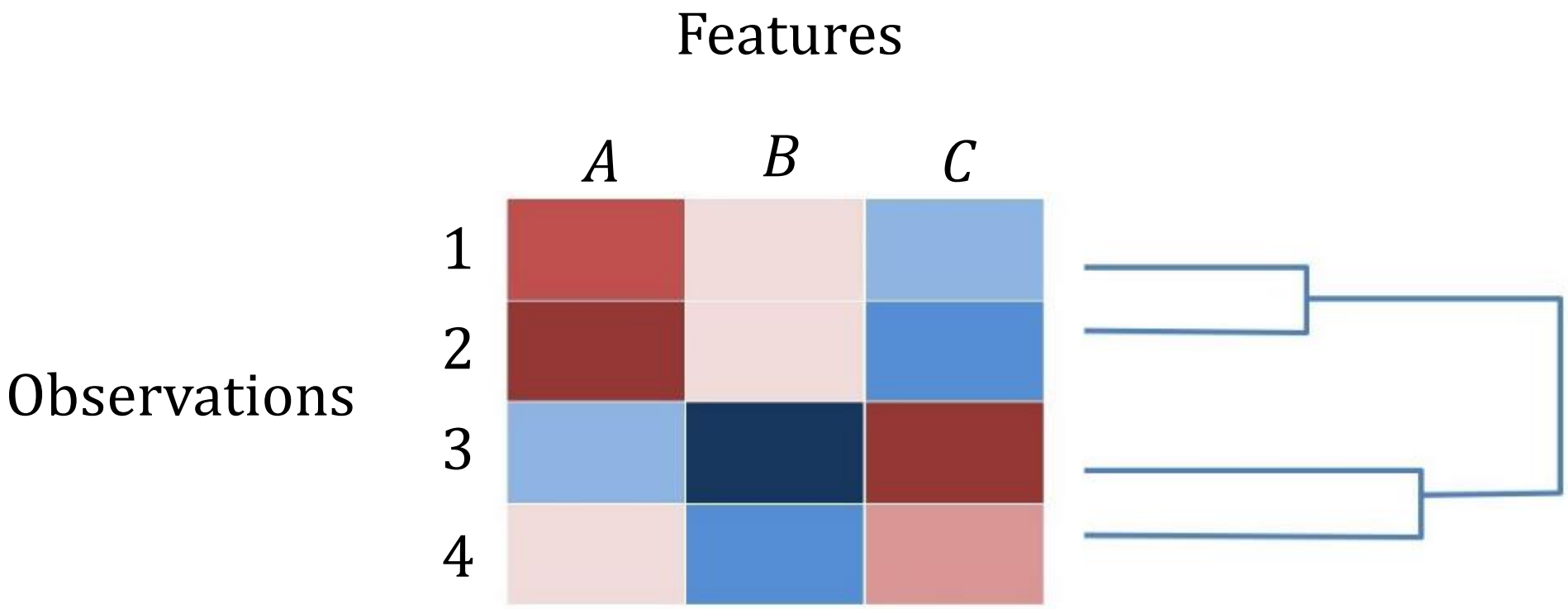
The main question: how to split, or how to merge, clusters?

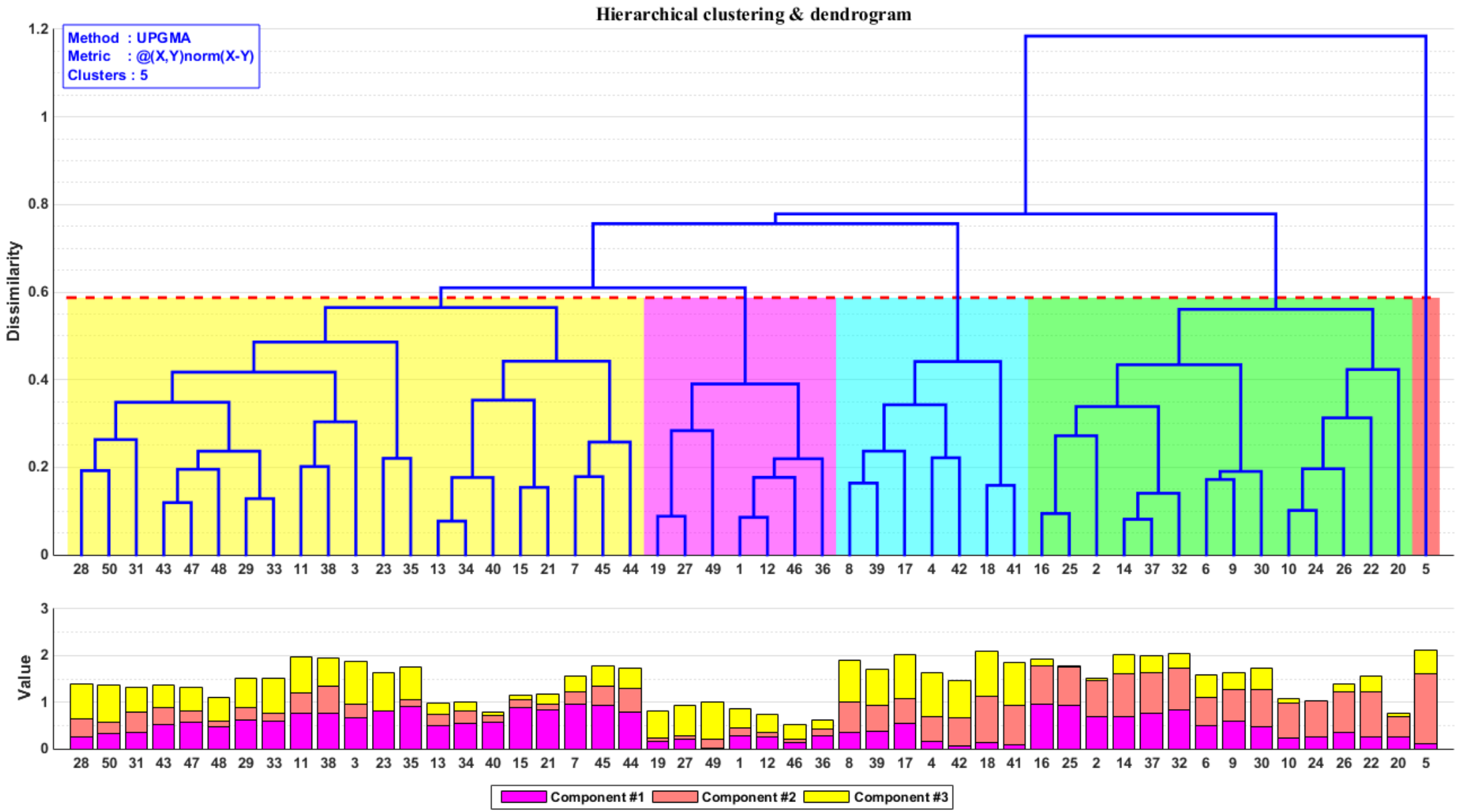
This requires the notion of a distance between clusters (**linkage**).

- in Bottom-up HC, **nearest pairs** of clusters are merged up the hierarchy (requires only computing distances between pairs)
- in Top-down HC, a cluster must be **optimally split** into sub-clusters down the hierarchy (much harder, computationally)

Another issue: at what level do we **report** the clustering scheme? When do we stop climbing or descending the hierarchy?

Latent class analysis might be a better approach, in general.


Features
A
B
C
Observations
1
2
3
4



[author unknown]

Back to Knowledge Discovery

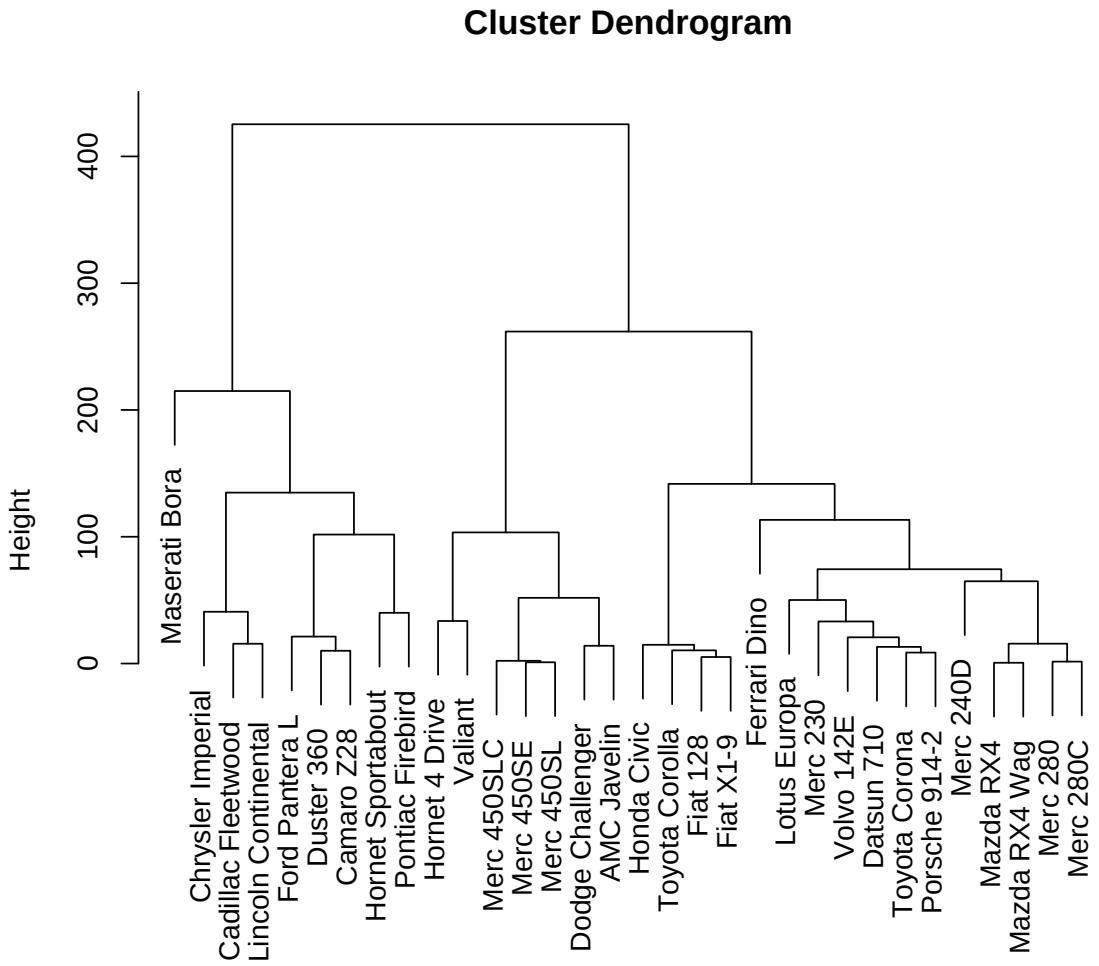
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1

More unsupervised learning: what underlying structures can we discover in this data?

[mtcars dataset]

Back to Knowledge Discovery

What do you notice in this diagram, structure-wise?

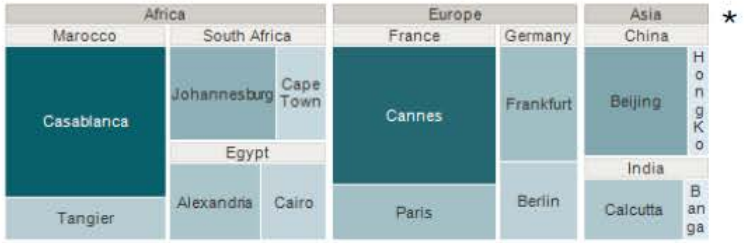


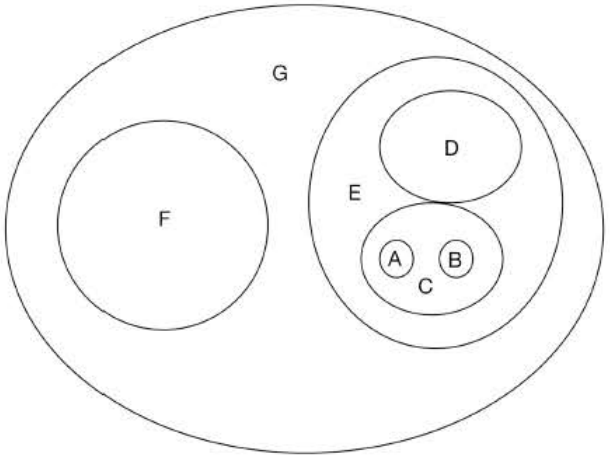
In a nutshell, **hierarchical systems** are ordered sets where elements and/or subsets are organized in a given relationship to one another, both among themselves and within the whole.

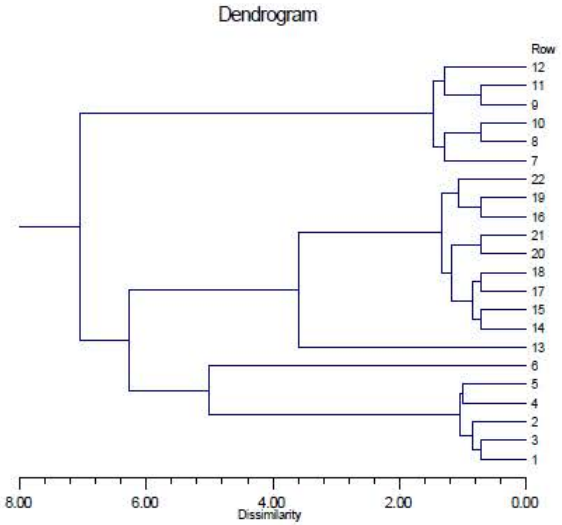
Relationships vary according to the field domain and type of system, but in general, we can describe them by the properties of elements and the laws that govern them (e.g., how they are shared and/or related).

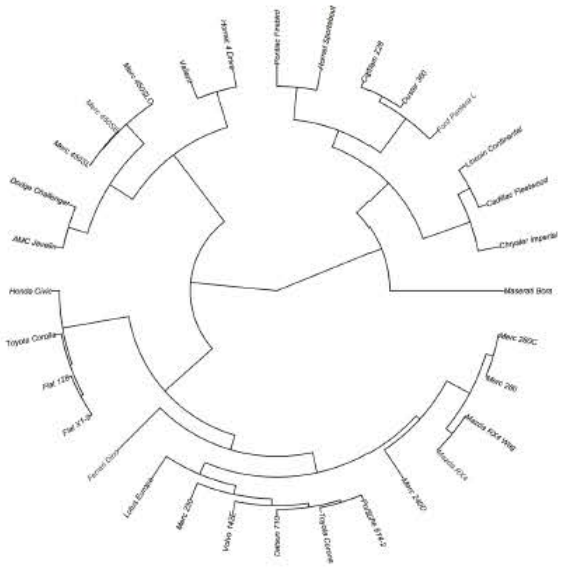
– I. Meireilles, *Design for Information*

Visualizing Hierarchy





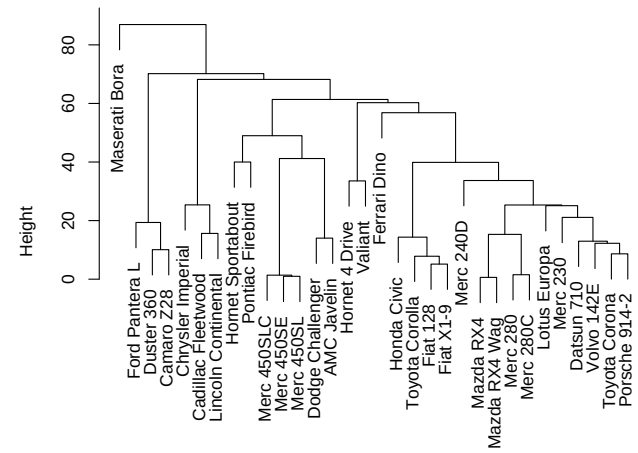


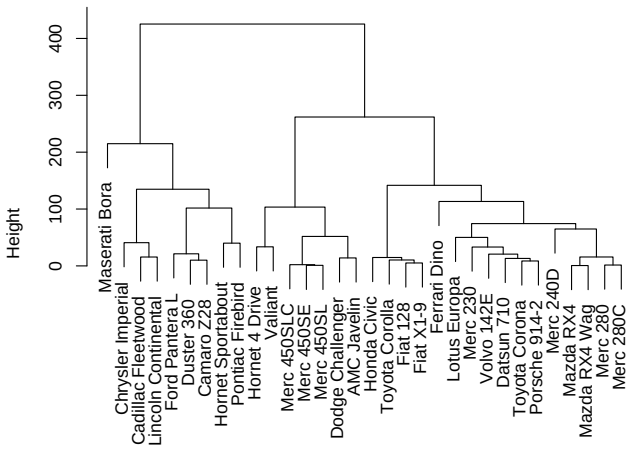


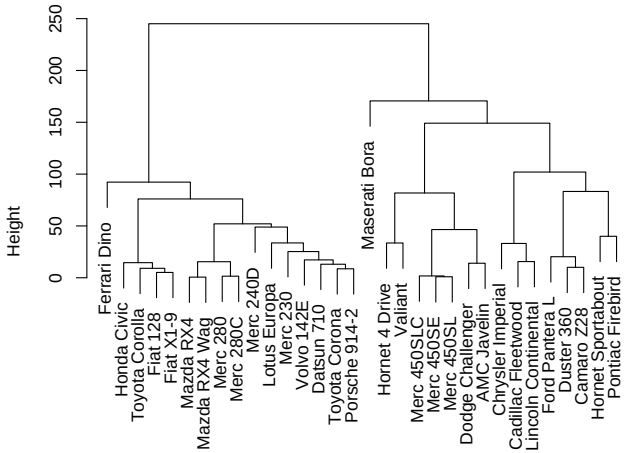
(mammals(primates (apes(orangutan,human)), (monkeys(pygmy marmoset)),(lemurs(ruffed lemur))),(cetacea (whales(long-finned pilot whale,southern right whale)),(dolphins(striped dolphin, bottle-nose dolphin))))

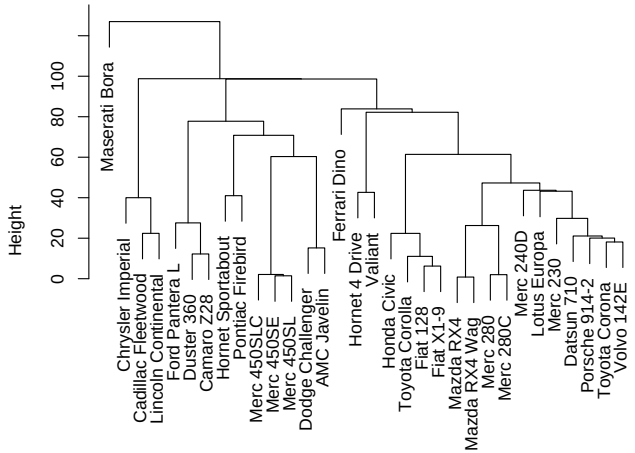
(*treemap from: TIBCO Spotfire Documentation: What is a Treemap?)

Button Press Hierarchical Clustering





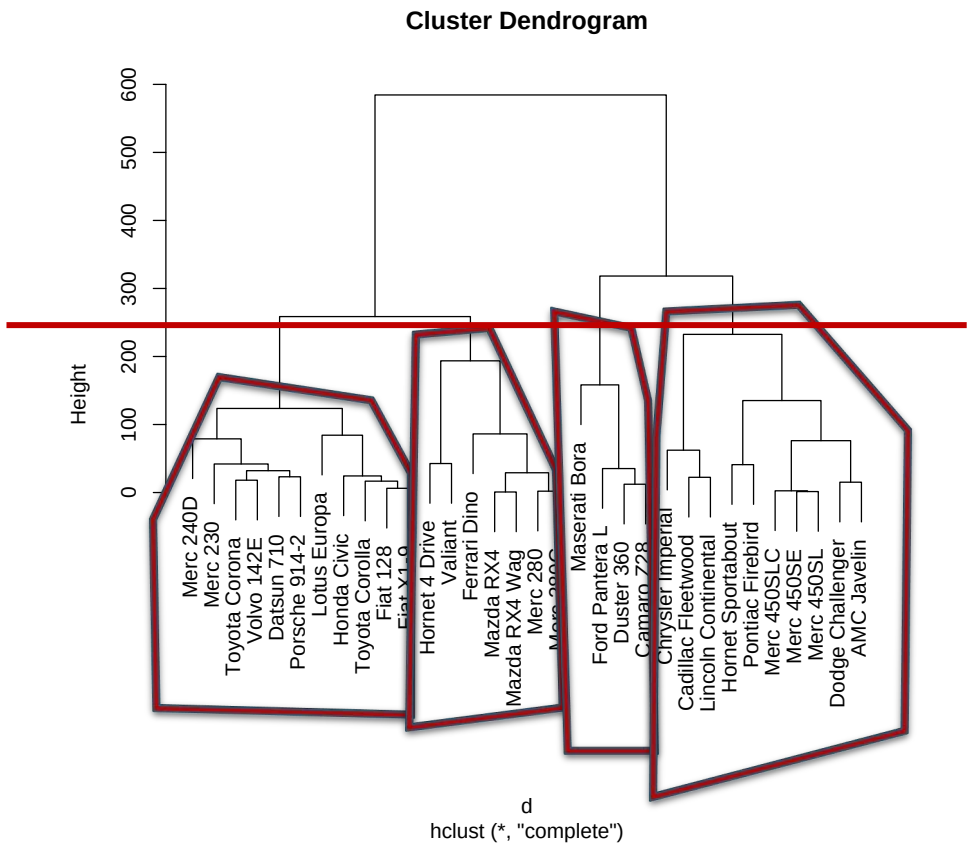


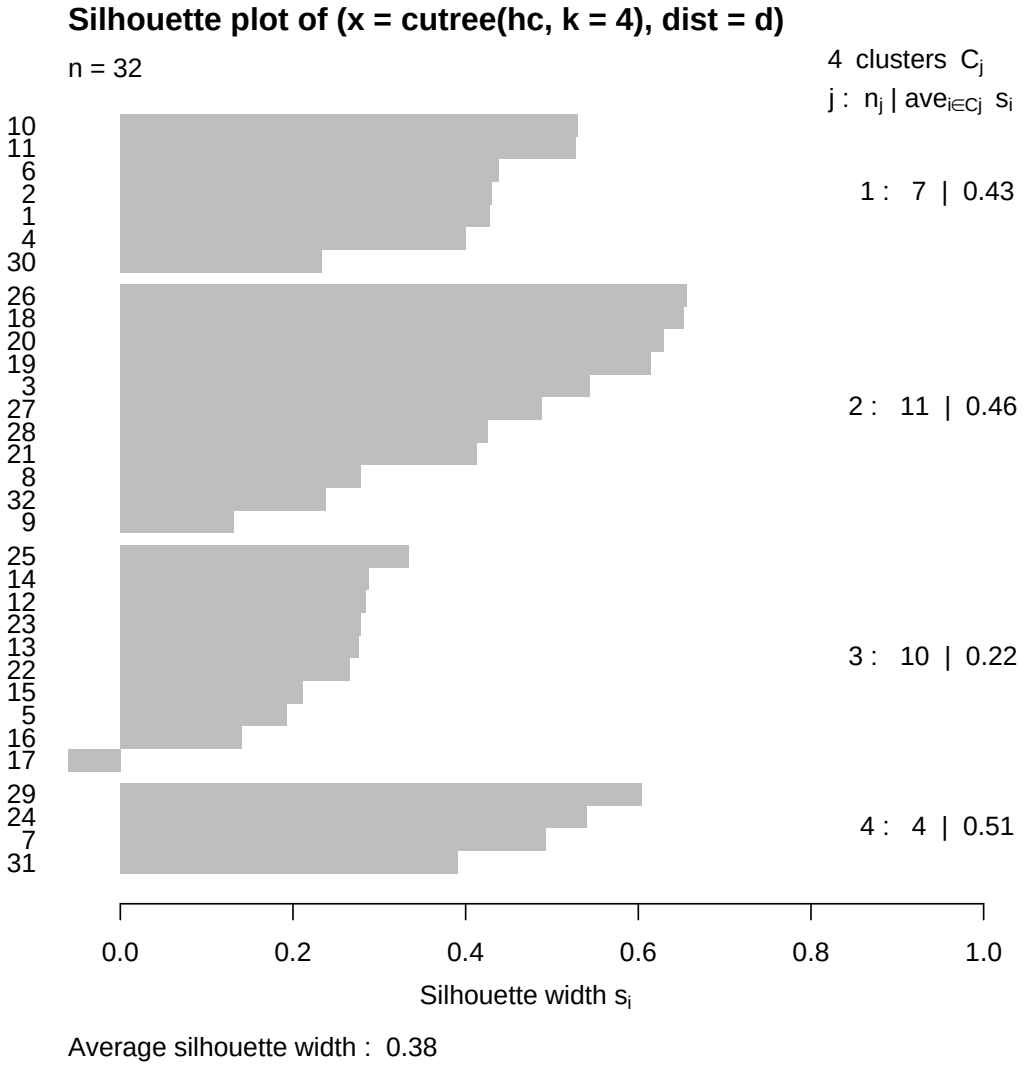


What can we say about the data structure if presented with a **cluster dendrogram**?

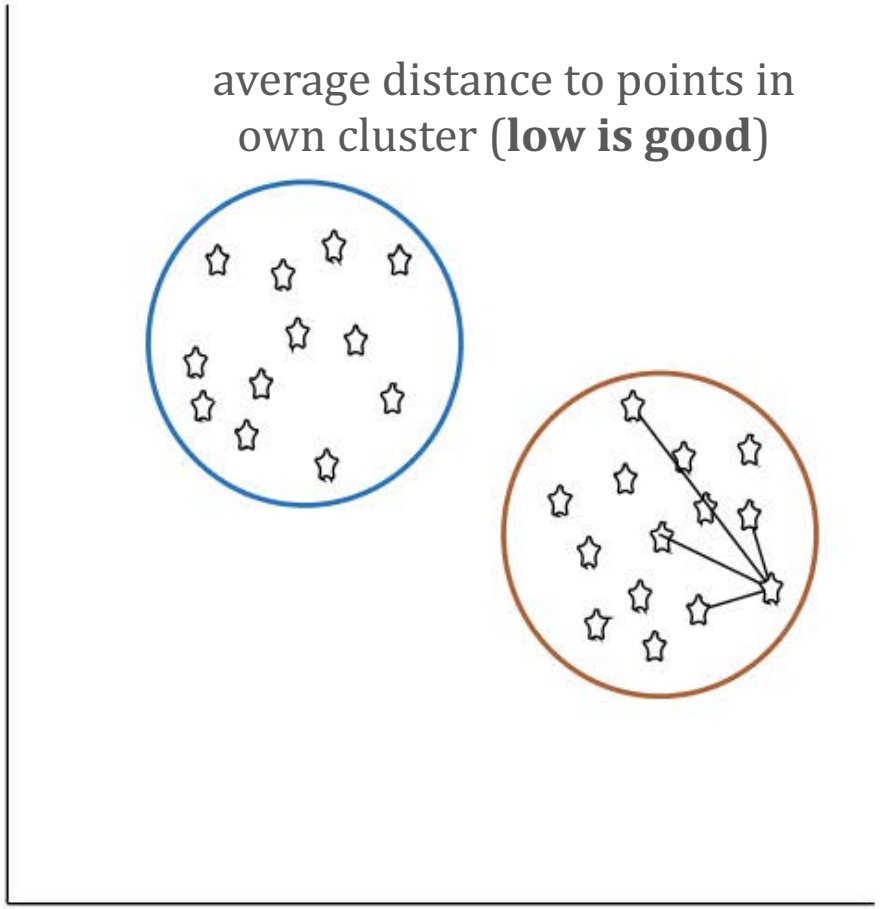
Same data clustered using different parameter settings: which one is **optimal**?

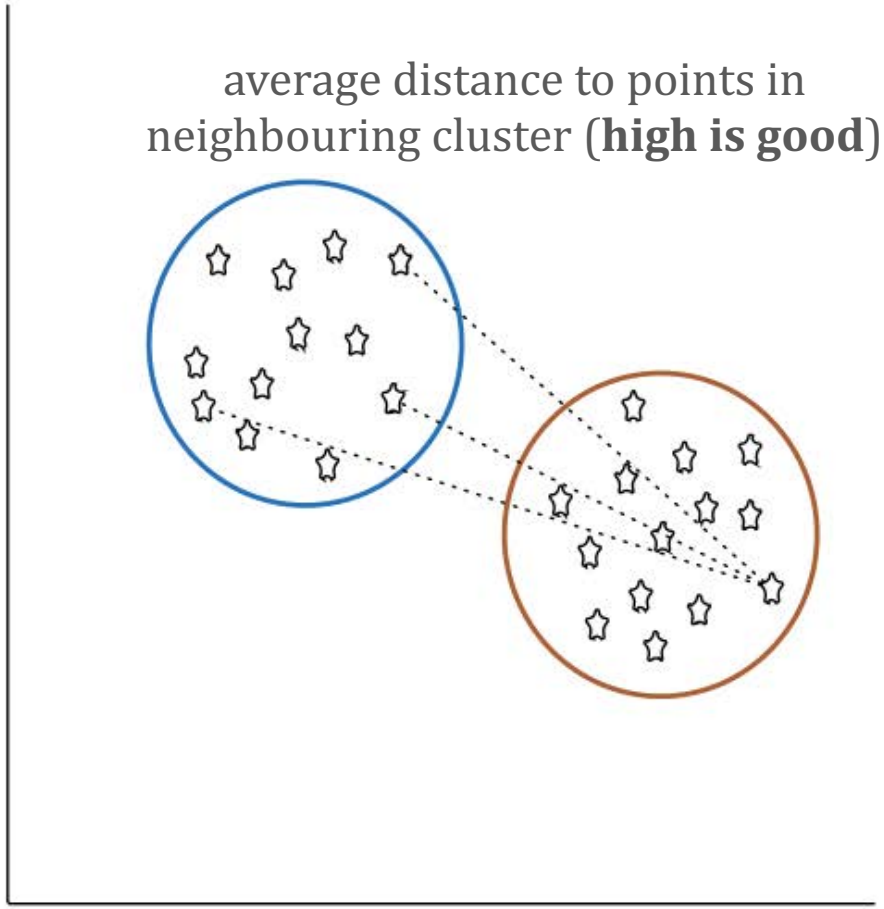
Evaluating the Results





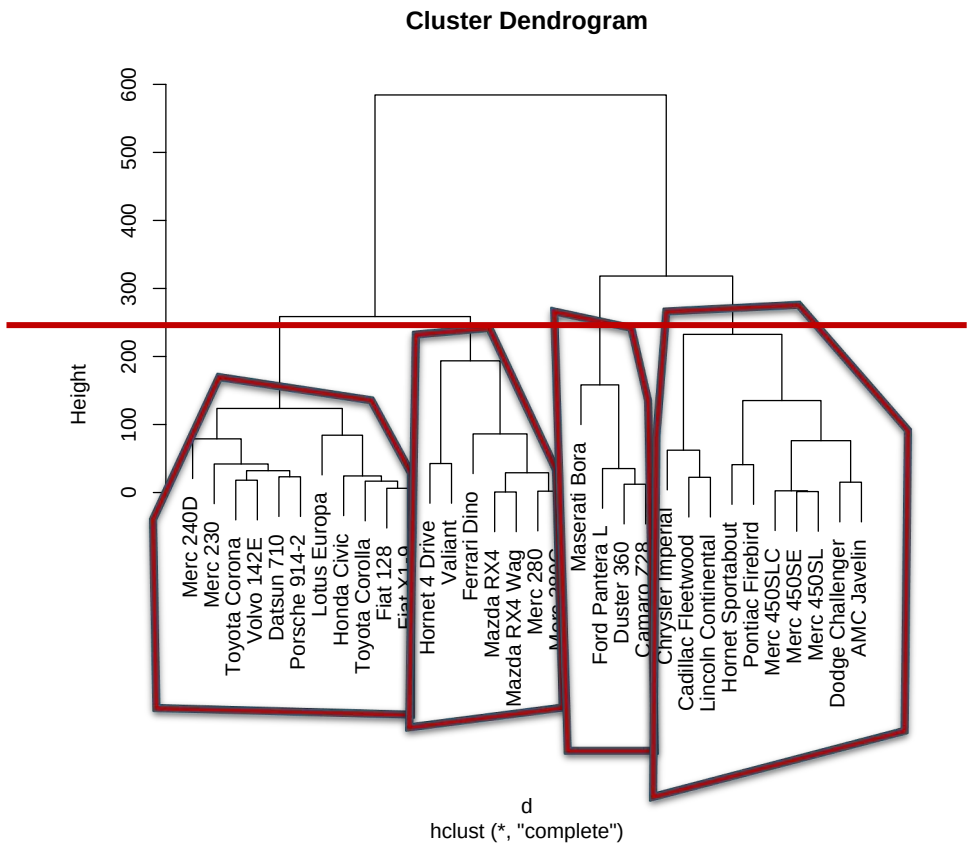
The Silhouette Metric

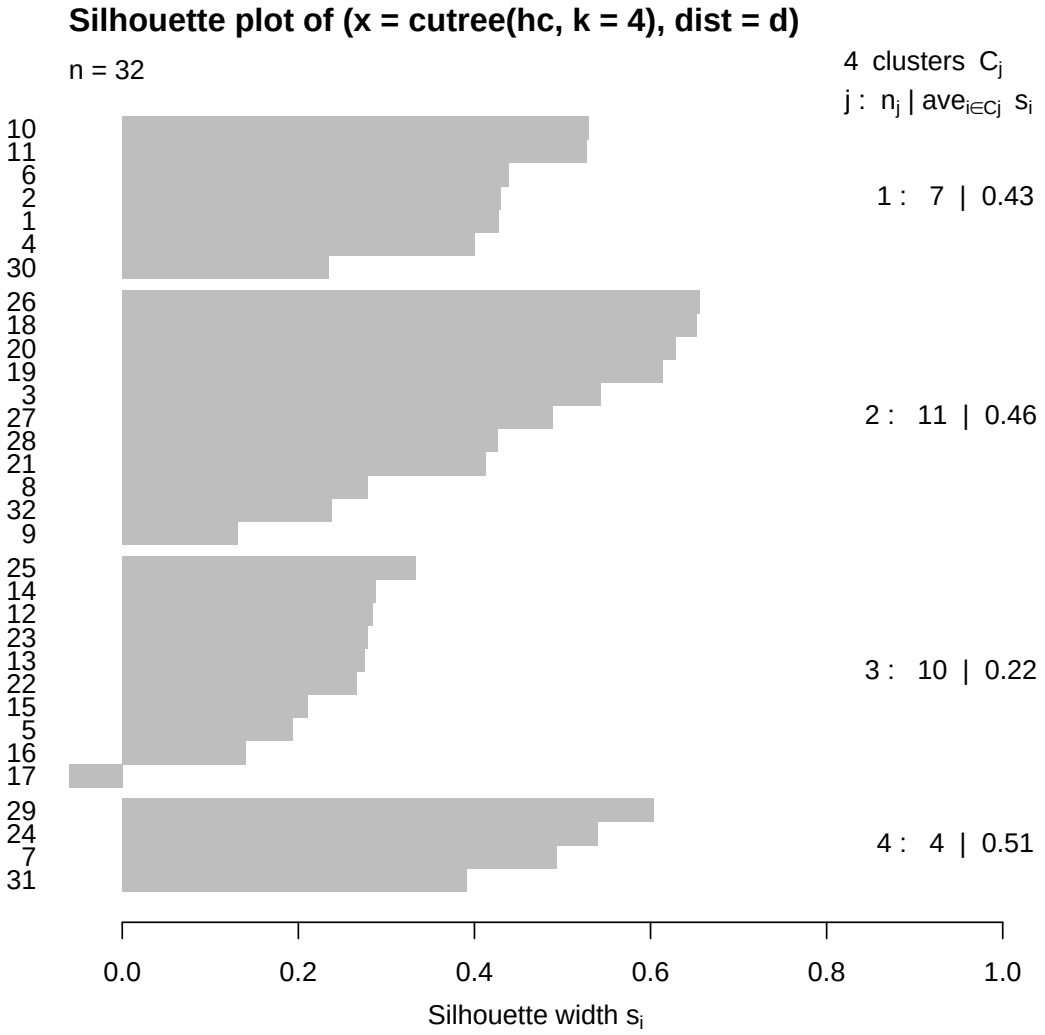




$$\text{silhouette metric for a point} = \frac{\text{average dissimilarity with neighbouring cluster} - \text{average dissimilaity with own cluster}}{\text{maximum dissimilarity value (own or neighbour)}}$$

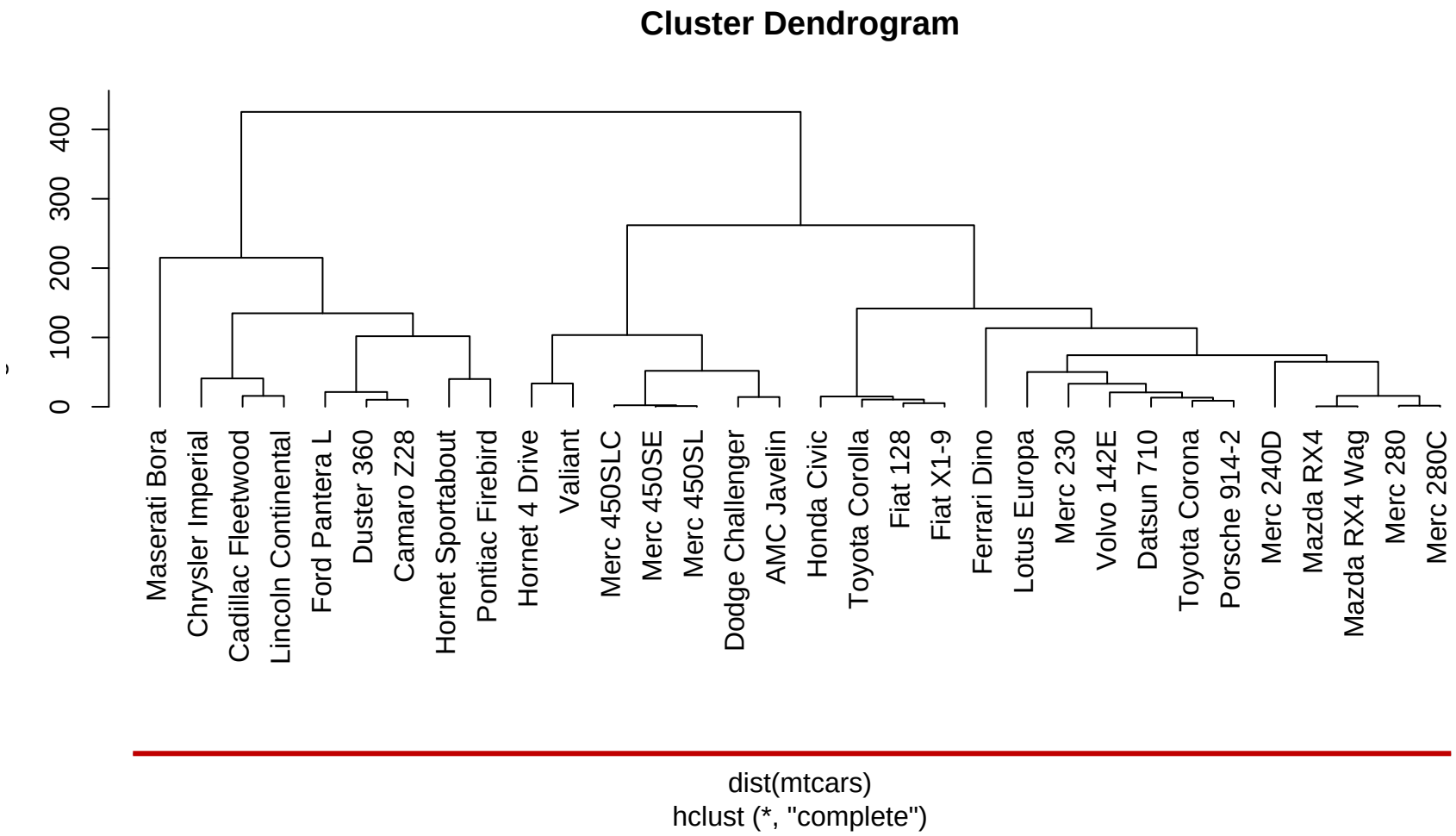
Evaluating the Results





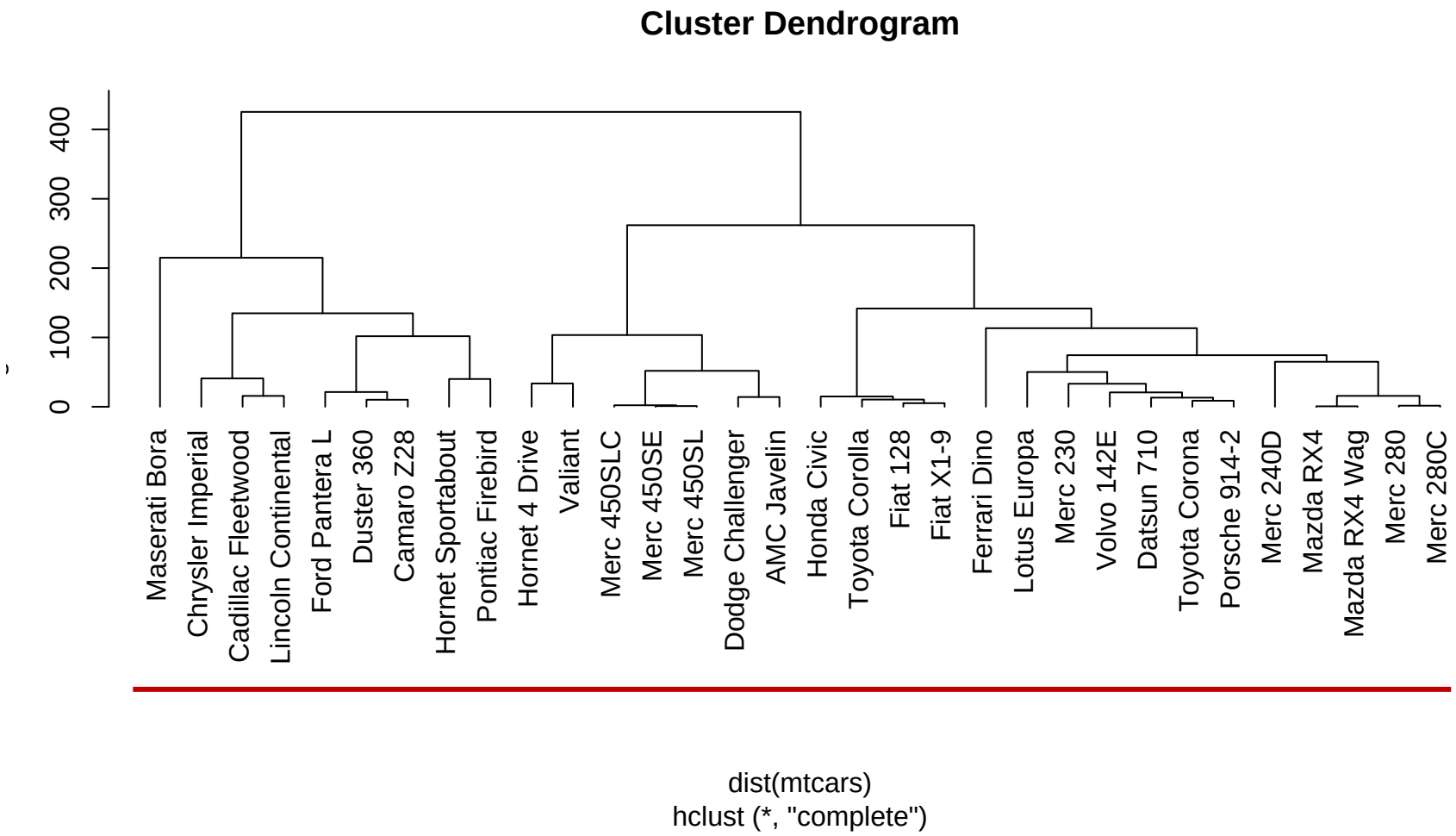
Hierarchical Clustering Algorithm

An Illustration with mtcars



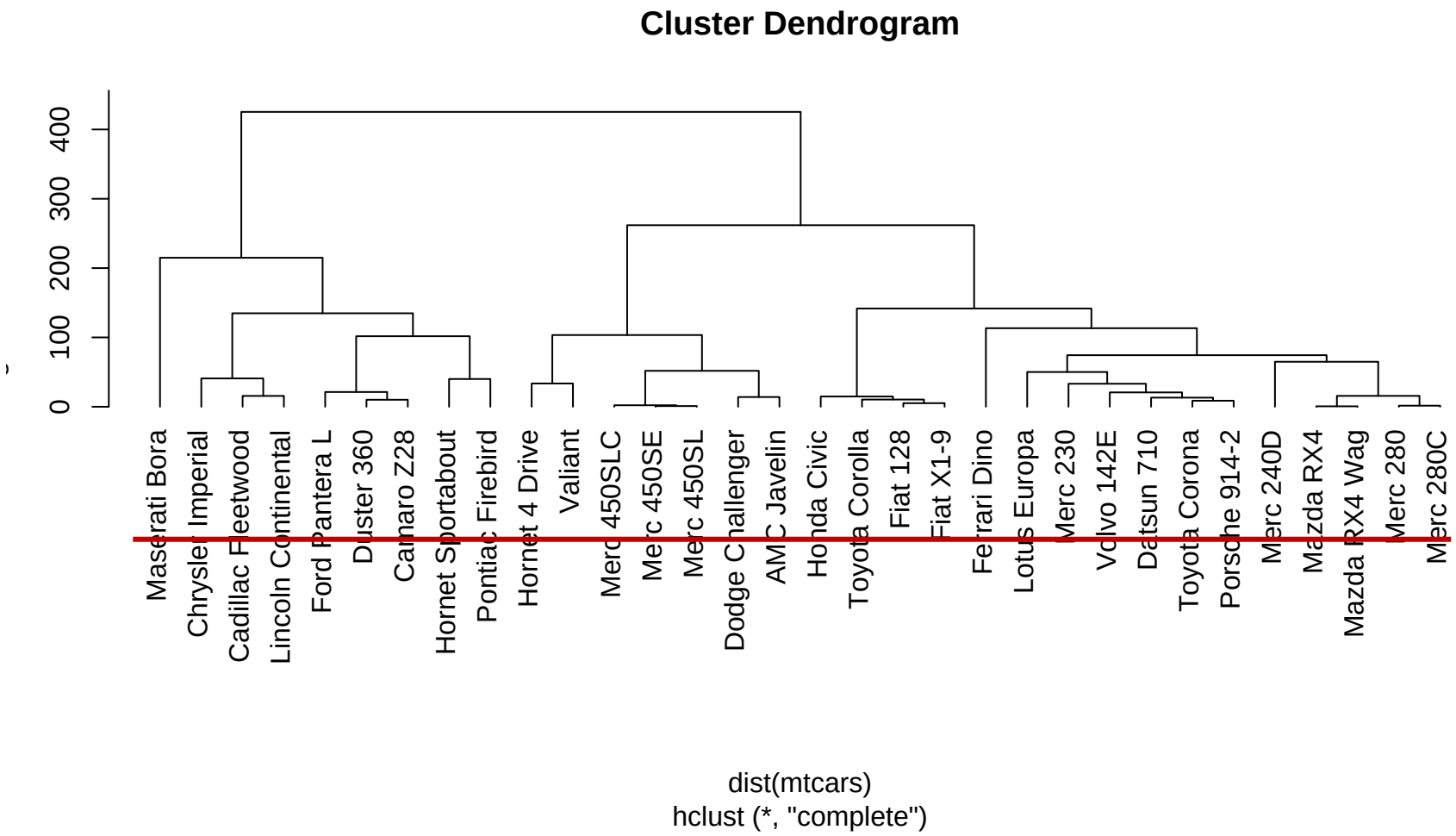
Hierarchical Clustering Algorithm

An Illustration with mtcars



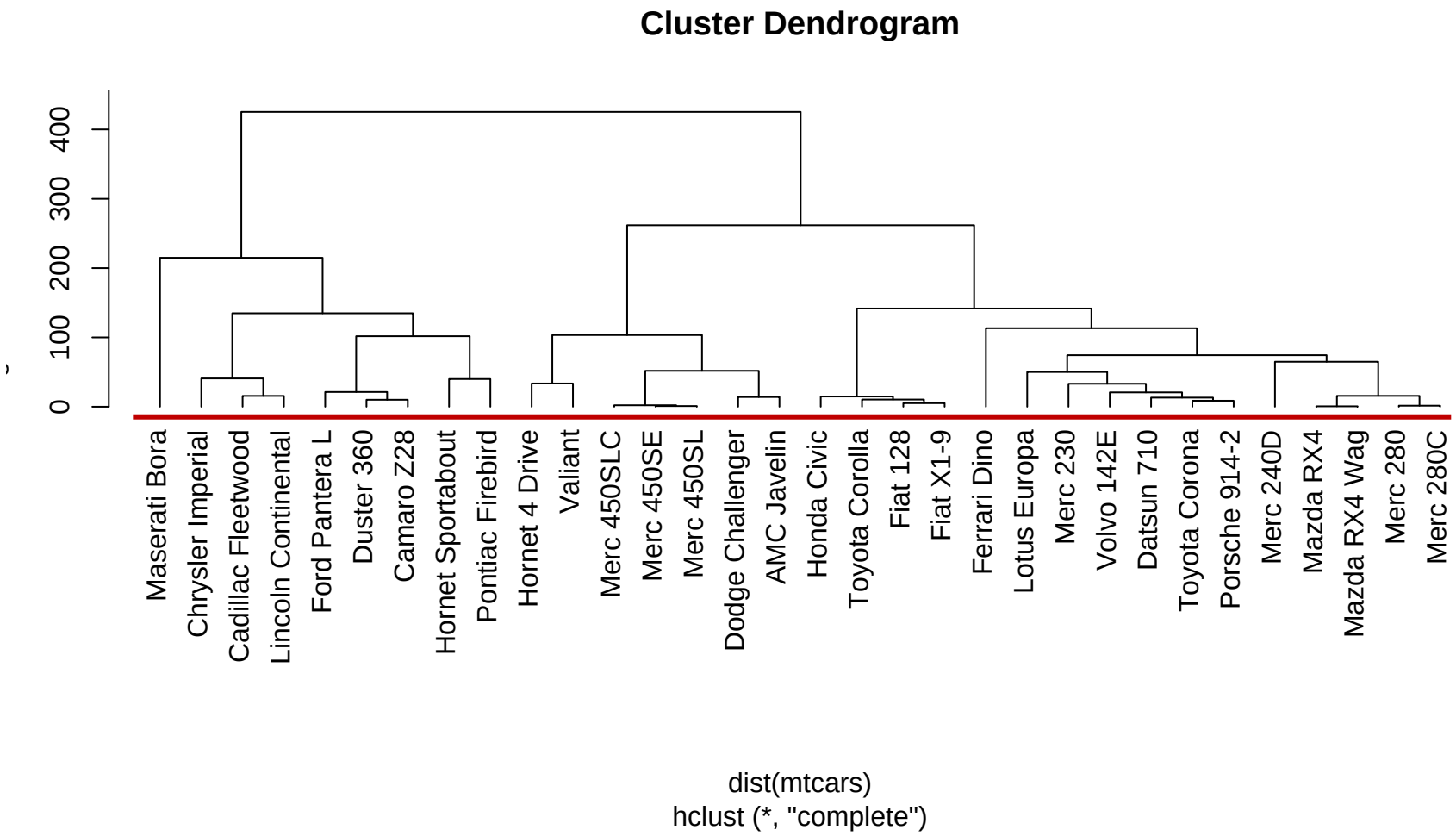
Hierarchical Clustering Algorithm

An Illustration with mtcars



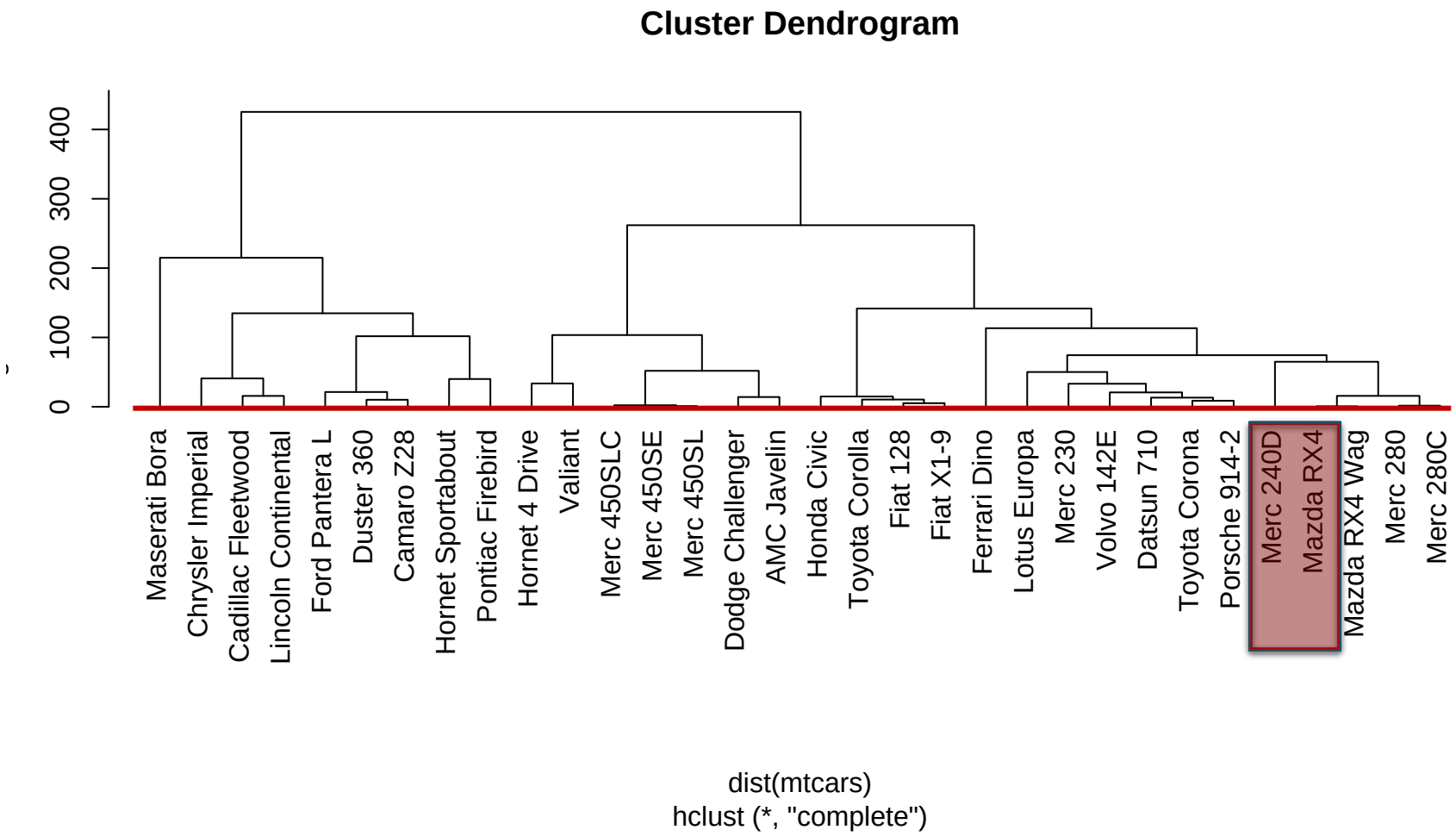
Hierarchical Clustering Algorithm

An Illustration with mtcars



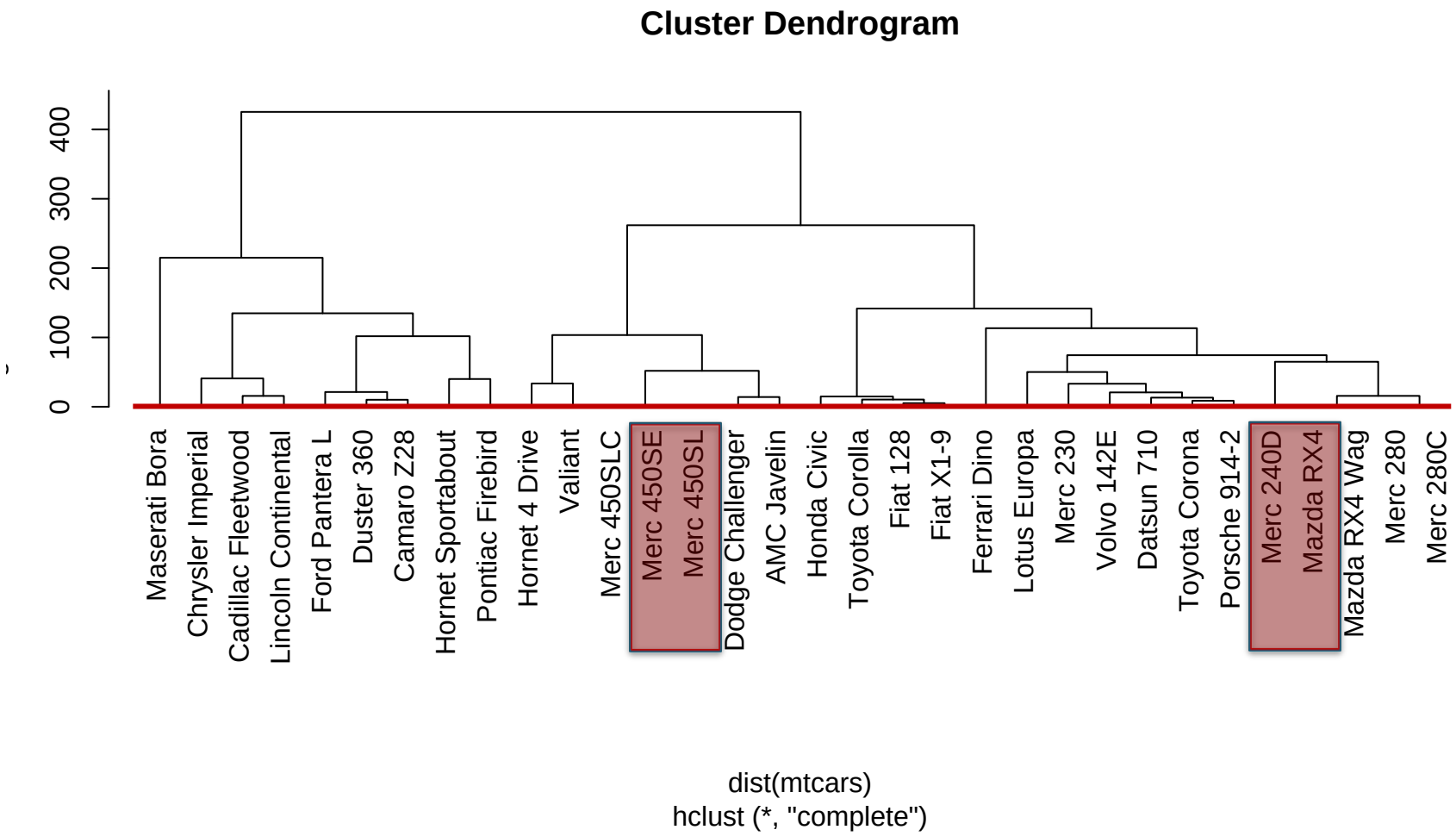
Hierarchical Clustering Algorithm

An Illustration with mtcars



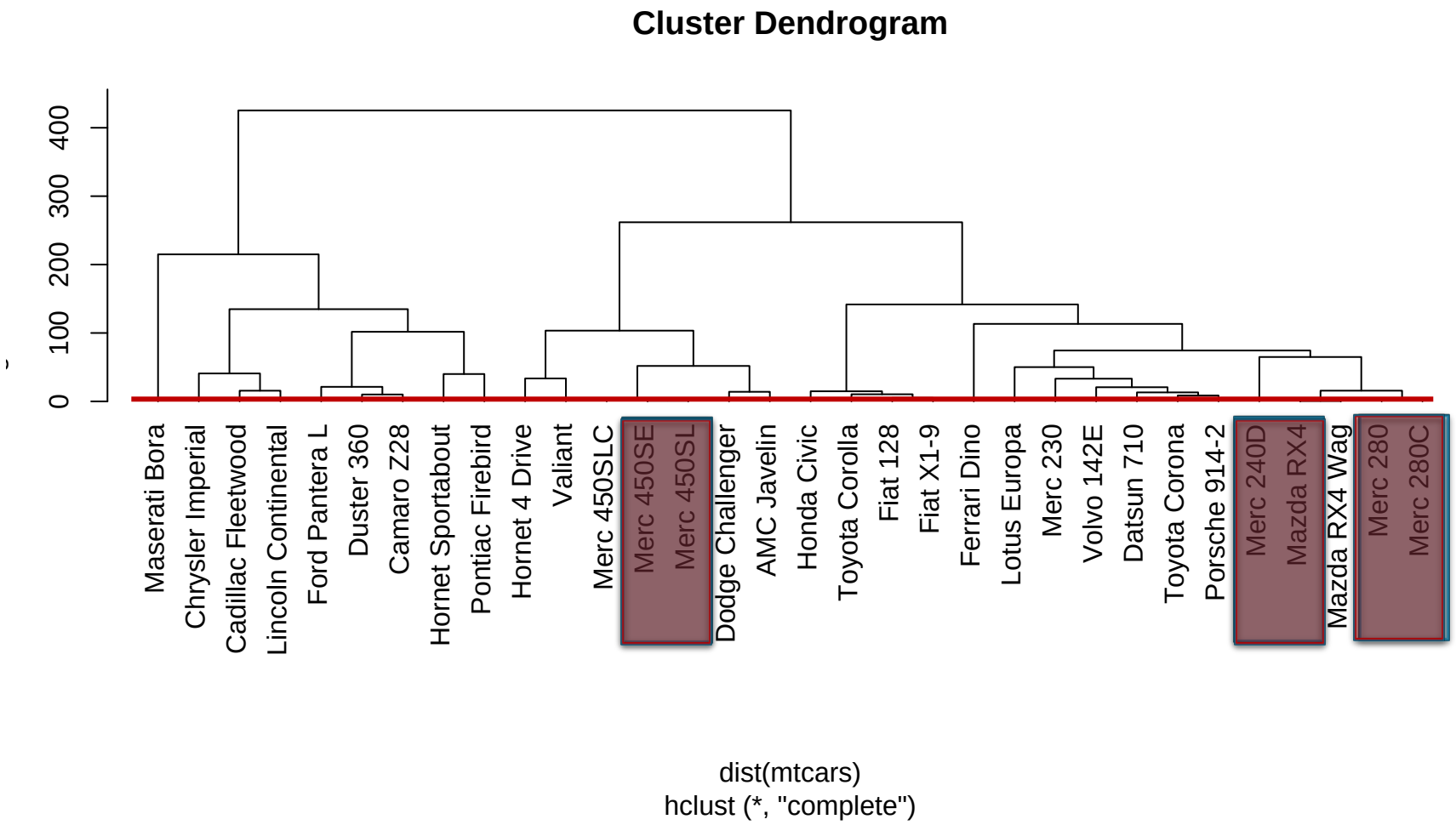
Hierarchical Clustering Algorithm

An Illustration with mtcars



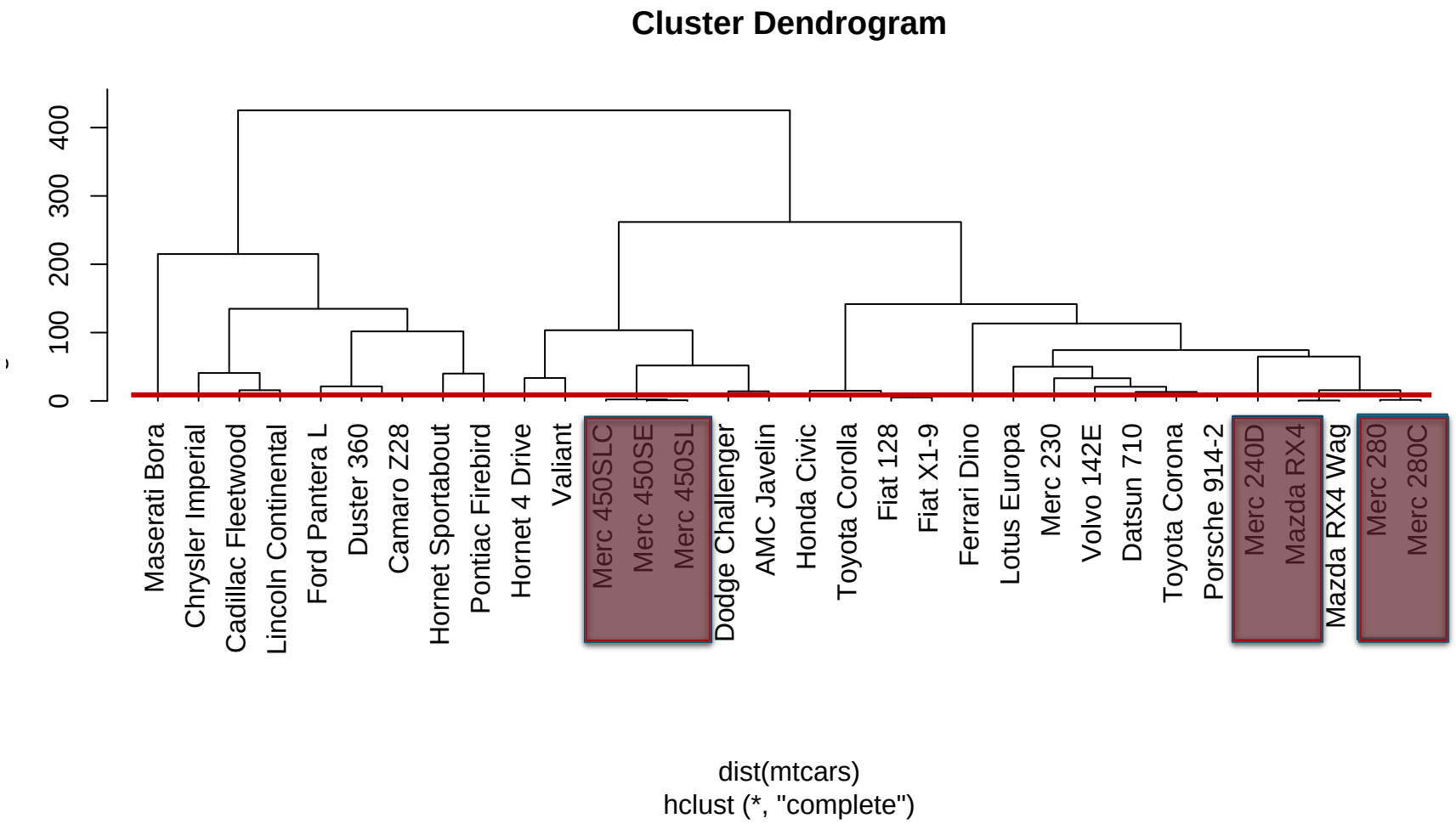
Hierarchical Clustering Algorithm

An Illustration with mtcars



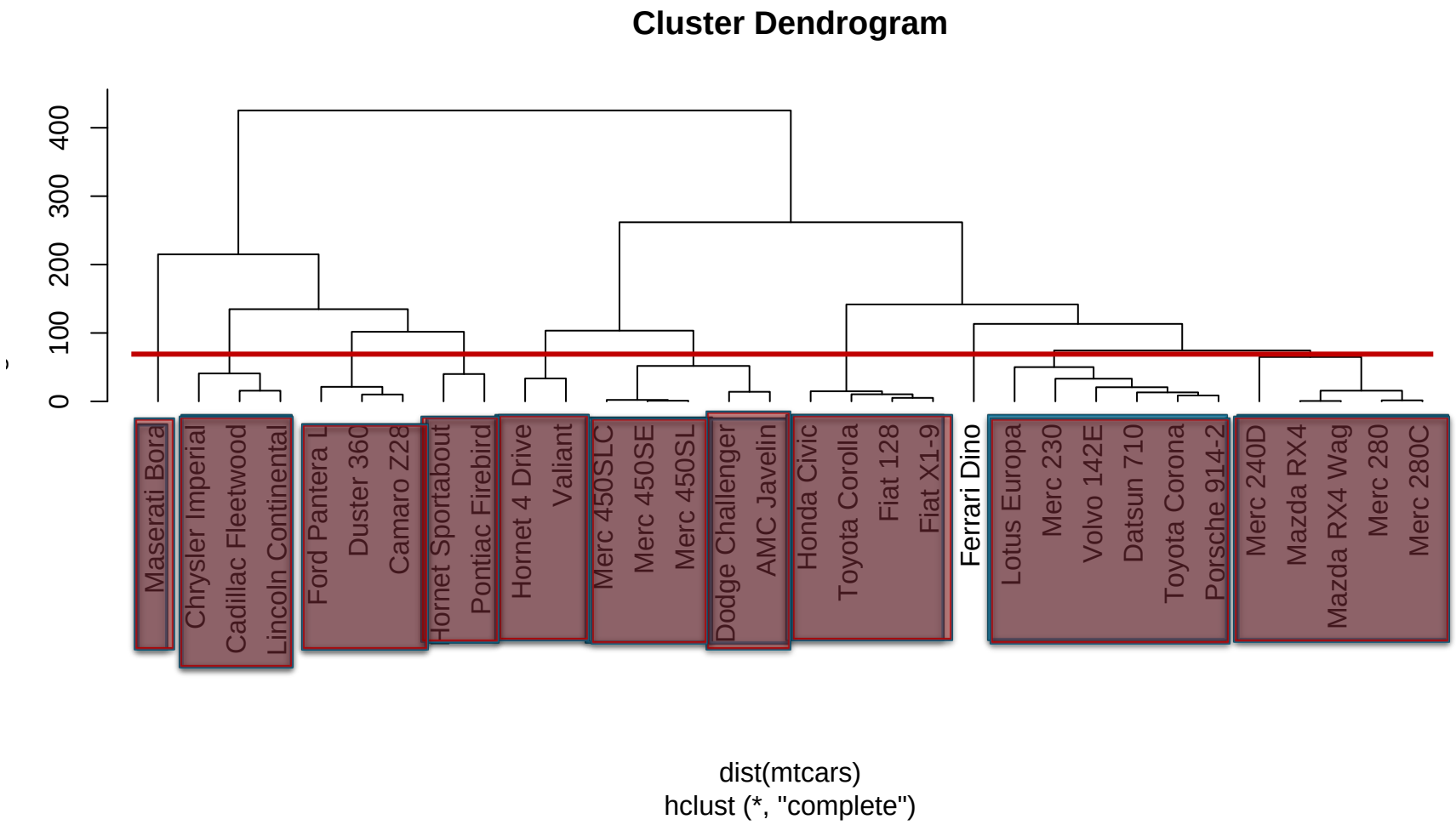
Hierarchical Clustering Algorithm

An Illustration with mtcars



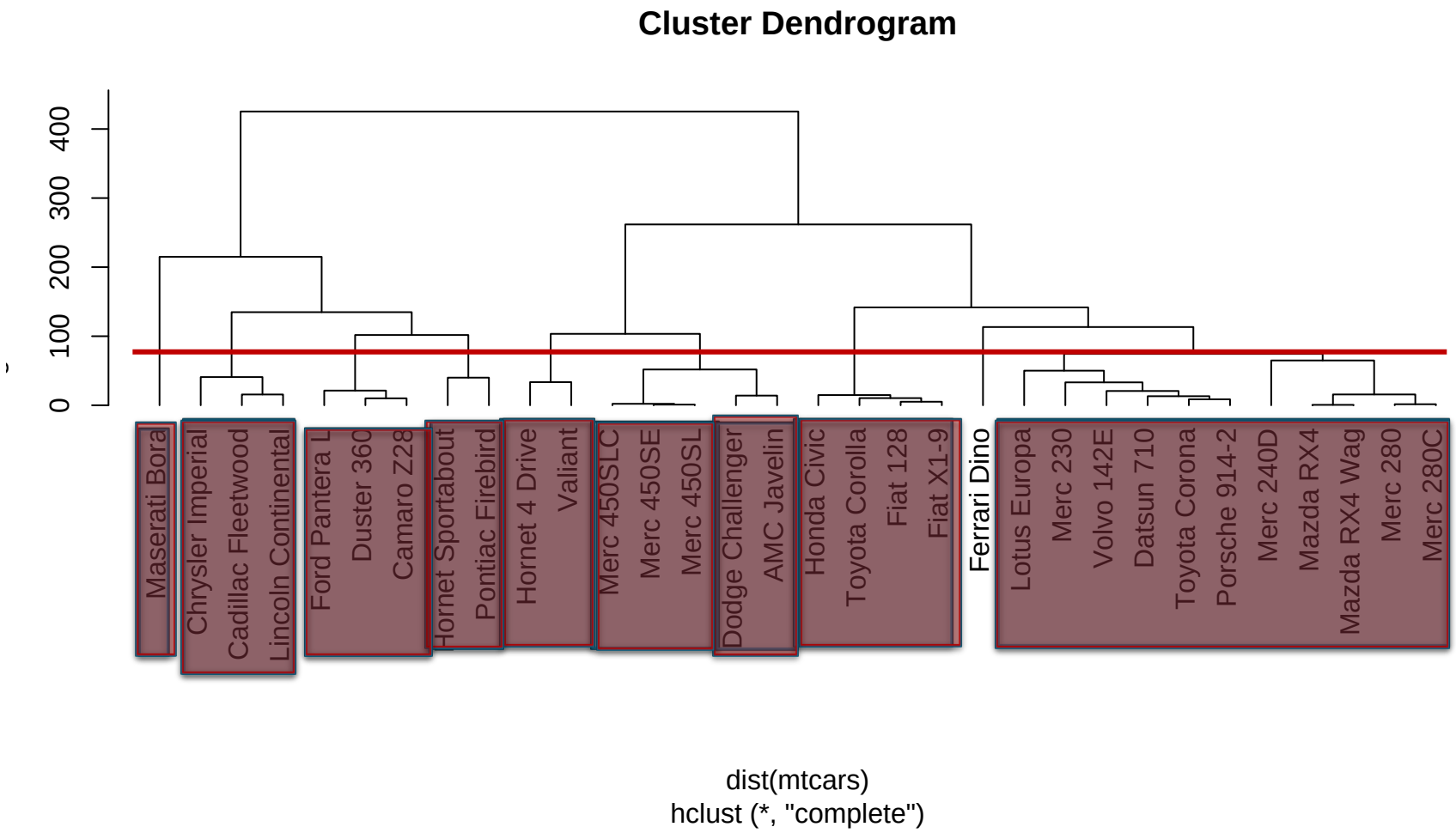
Hierarchical Clustering Algorithm

An Illustration with mtcars



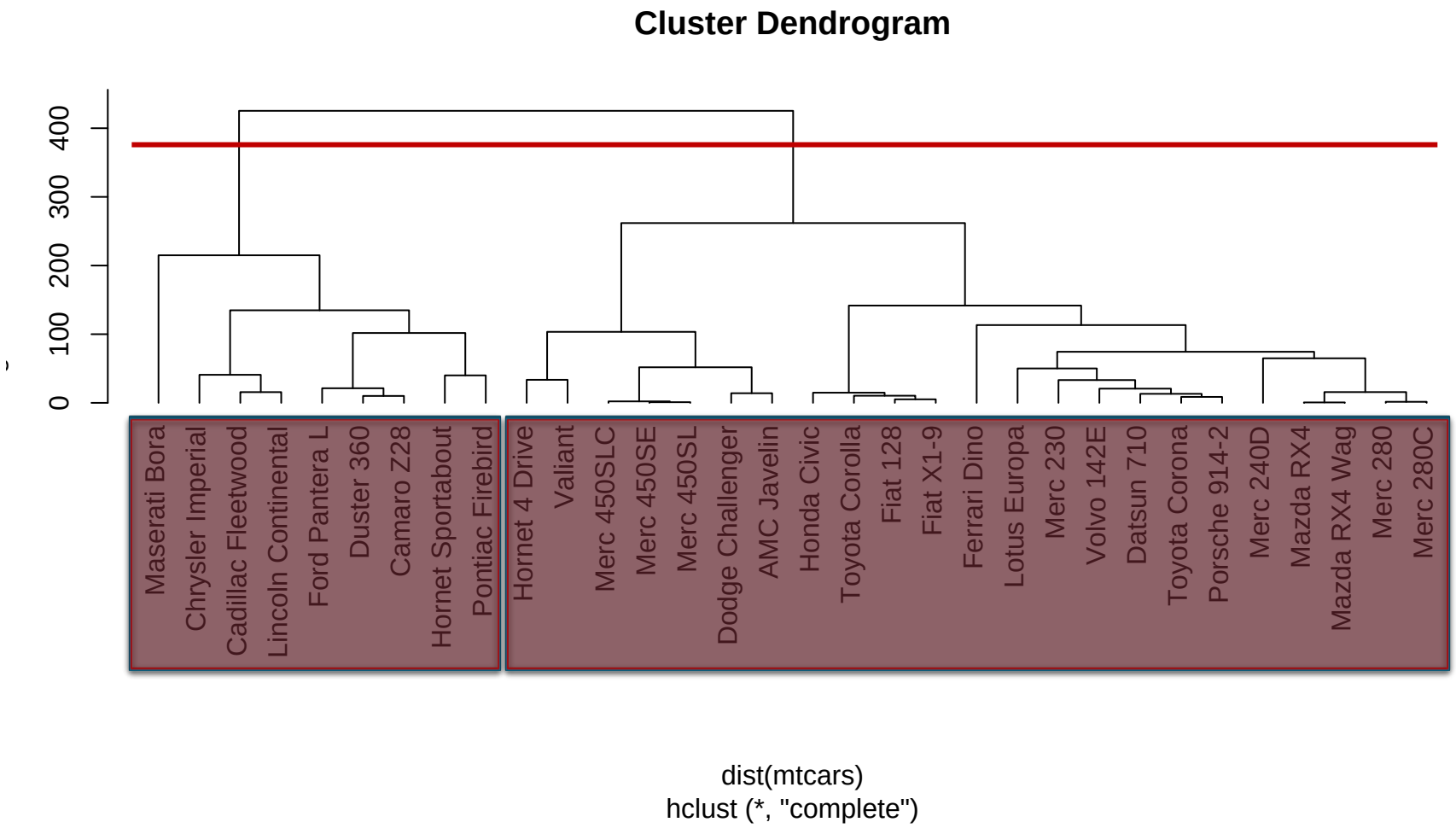
Hierarchical Clustering Algorithm

An Illustration with mtcars



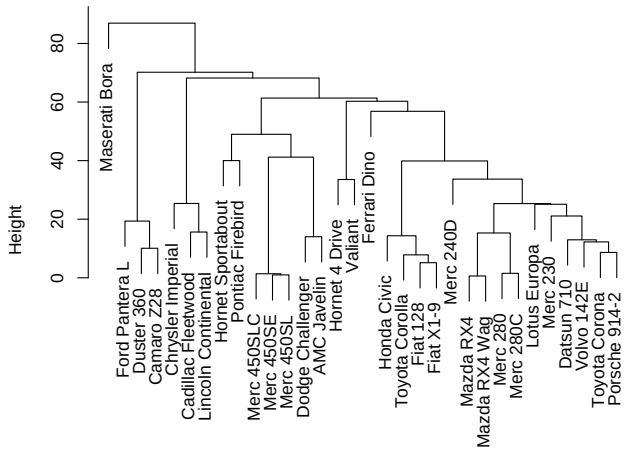
Hierarchical Clustering Algorithm

An Illustration with mtcars

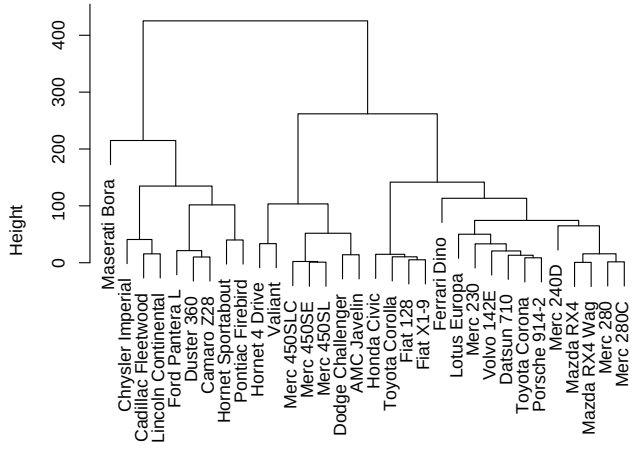


Hierarchical Clustering Algorithm

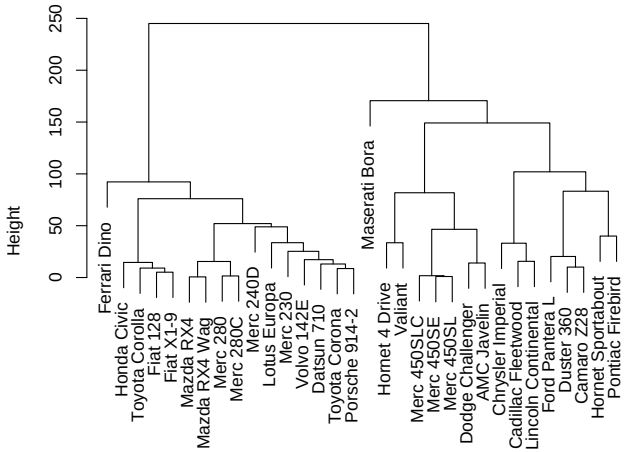
Same Cluster, Different Parameters



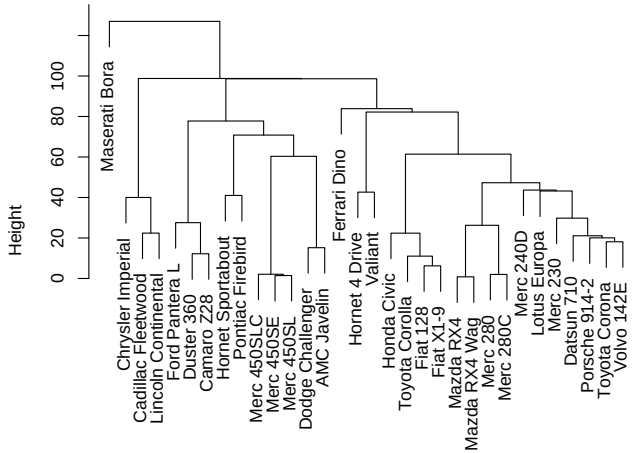
Euclidean Distance Single Link



Euclidean Distance Complete Link



Euclidean Distance Average Link



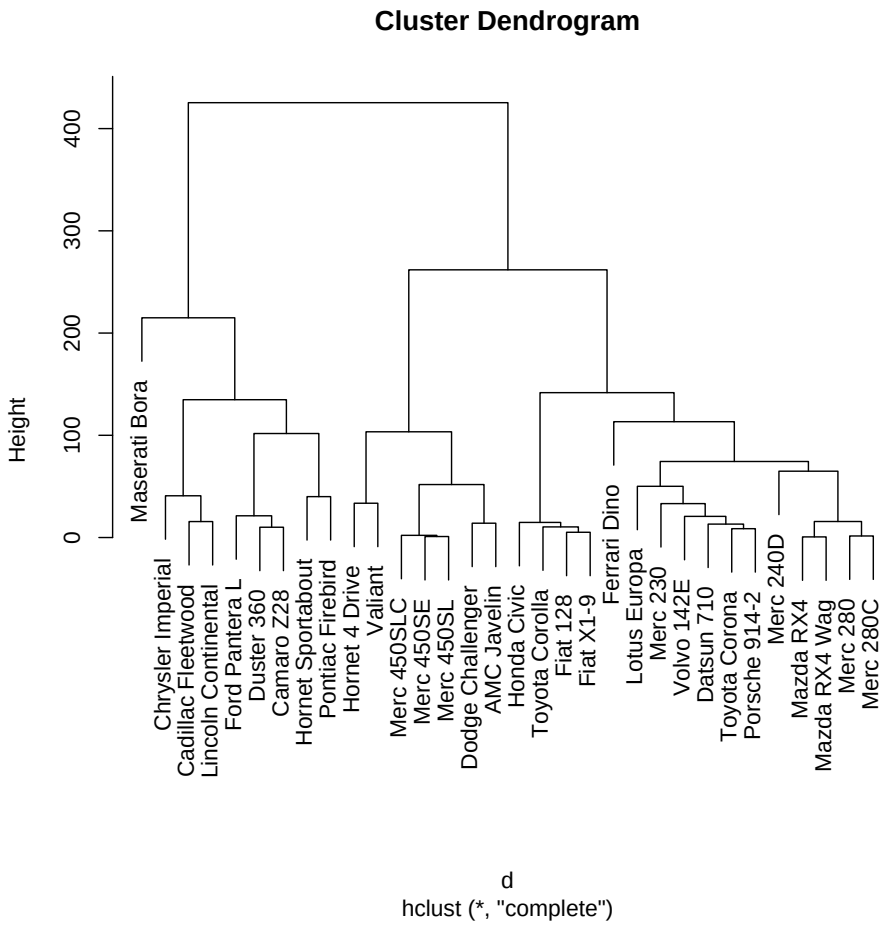
Manhattan Distance Single Link

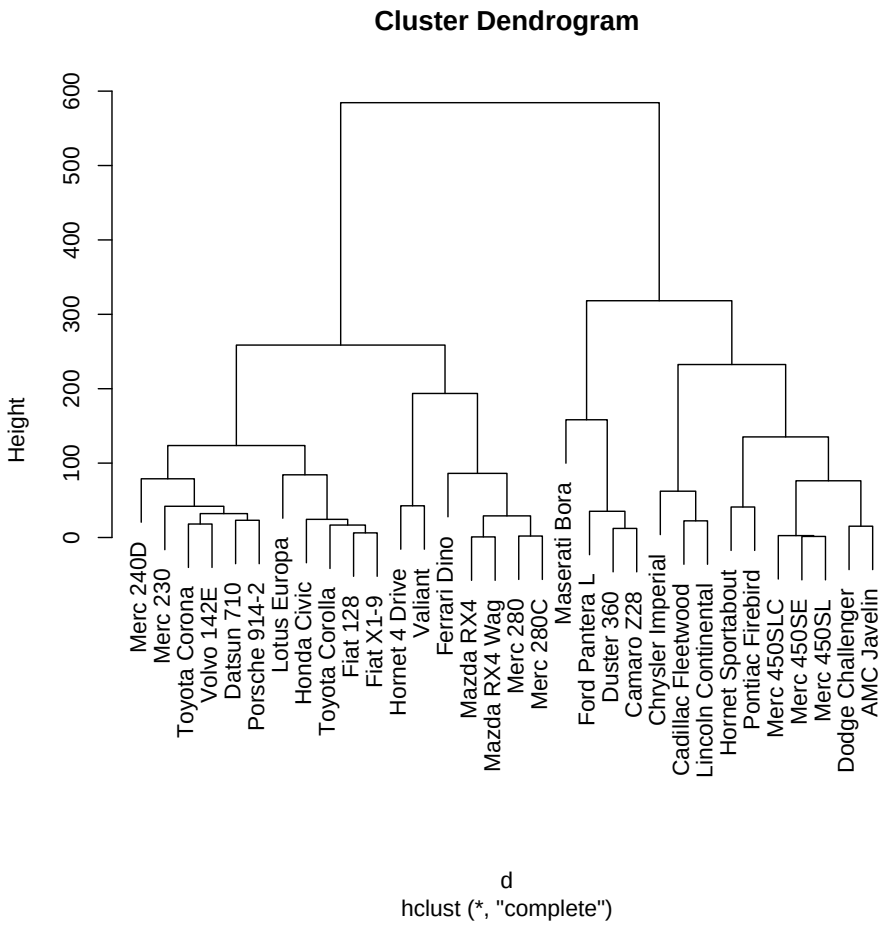
Same data clustered using different parameter settings:

- **distance** metric
- **linkage** strategy

Hierarchical Clustering Algorithm

Parameters: Distance Metric





Same data clustered using two different distance metrics (euclidean, manhattan)

Hierarchical Clustering Algorithm

Similarity-Dissimilarity

Compare objects

```
> mtcars
                    mpg cyl  disp  hp drat    wt
Mazda RX4          21.0   6 160.0 110 3.90 2.620
Mazda RX4 Wag      21.0   6 160.0 110 3.90 2.875
Datsun 710         22.8   4 108.0  93 3.85 2.320
Hornet 4 Drive     21.4   6 258.0 110 3.08 3.215
Hornet Sportabout  18.7   8 360.0 175 3.15 3.440
Valiant            18.1   6 225.0 105 2.76 3.460
Duster 360         14.3   8 360.0 245 3.21 3.570
Merc 240D          24.4   4 146.7  62 3.69 3.190
Merc 230           22.8   4 140.8  95 3.92 3.150
Merc 280           19.2   6 167.6 123 3.92 3.440
Merc 280C          17.8   6 167.6 123 3.92 3.440
Merc 450SE         16.4   8 275.8 180 3.07 4.070
Merc 450SL         17.3   8 275.8 180 3.07 3.730
Merc 450SLC        15.2   8 275.8 180 3.07 3.780
Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250
Lincoln Continental 10.4  8 460.0 215 3.00 5.424
Chrysler Imperial  14.7   8 440.0 230 3.23 5.345
Fiat 128           32.4   4  78.7  66 4.08 2.200
Honda Civic        30.4   4  75.7  52 4.93 1.615
Toyota Corolla     33.9   4  71.1  65 4.22 1.835
Toyota Corona      21.5   4 120.1  97 3.70 2.465
Dodge Challenger   15.5   8 318.0 150 2.76 3.520
AMC Javelin        15.2   8 304.0 150 3.15 3.435
Camaro Z28         13.3   8 350.0 245 3.73 3.840
Pontiac Firebird   19.2   8 400.0 175 3.08 3.845
Fiat X1-9          27.3   4  79.0  66 4.08 1.935
Porsche 914-2      26.0   4 120.3  91 4.43 2.140
Lotus Europa       30.4   4  95.1 113 3.77 1.513
Ford Pantera L     15.8   8 351.0 264 4.22 3.170
Ferrari Dino       19.7   6 145.0 175 3.62 2.770
Maserati Bora      15.0   8 301.0 335 3.54 3.570
Volvo 142E         21.4   4 121.0 109 4.11 2.780
```

Compare values

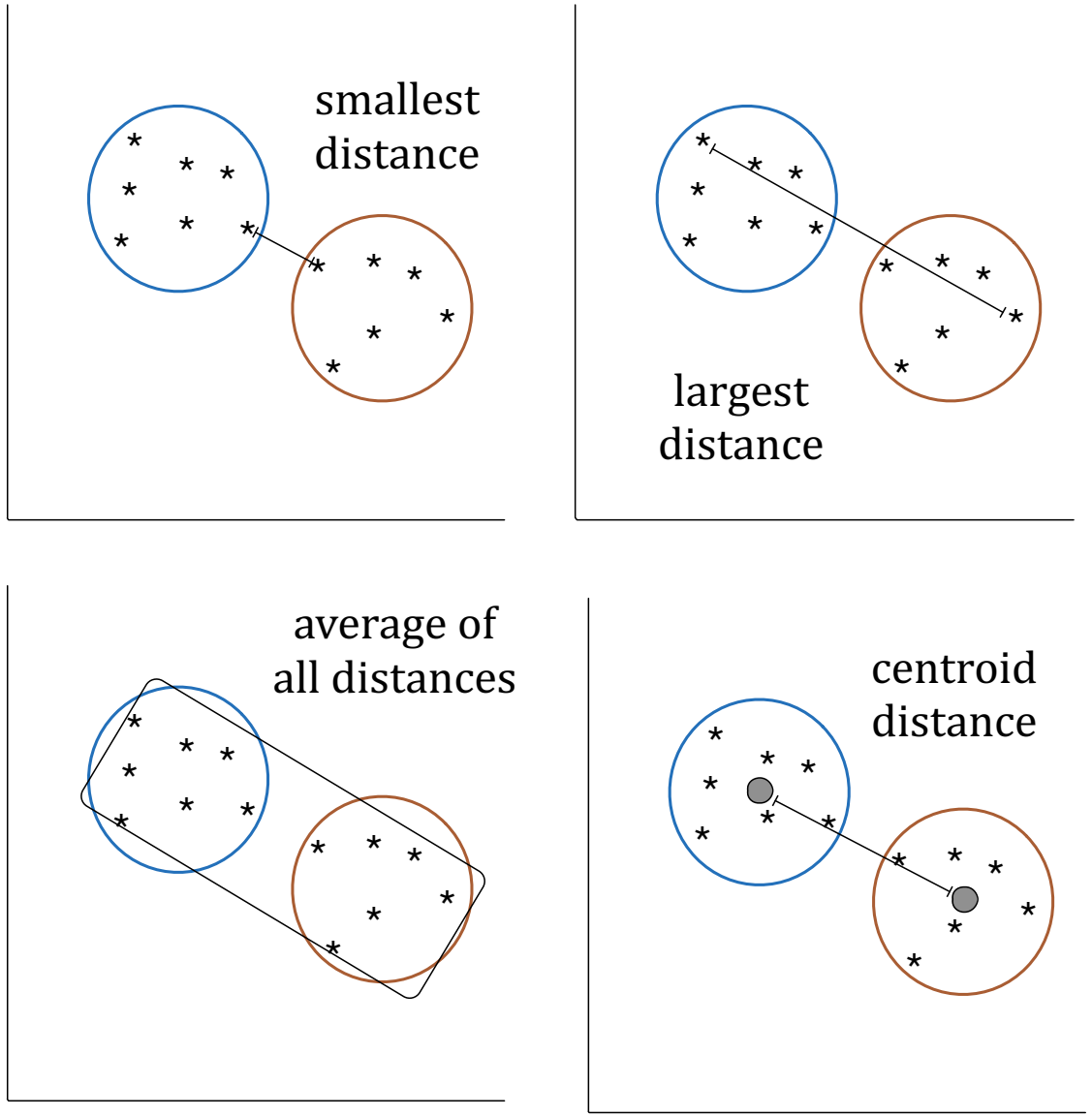
Compare groups

Compare variables

```
> mtcars
                    mpg cyl  disp  hp drat    wt
Mazda RX4          21.0   6 160.0 110 3.90 2.620
Mazda RX4 Wag      21.0   6 160.0 110 3.90 2.875
Datsun 710         22.8   4 108.0  93 3.85 2.320
Hornet 4 Drive     21.4   6 258.0 110 3.08 3.215
Hornet Sportabout  18.7   8 360.0 175 3.15 3.440
Valiant            18.1   6 225.0 105 2.76 3.460
Duster 360         14.3   8 360.0 245 3.21 3.570
Merc 240D          24.4   4 146.7  62 3.69 3.190
Merc 230           22.8   4 140.8  95 3.92 3.150
Merc 280           19.2   6 167.6 123 3.92 3.440
Merc 280C          17.8   6 167.6 123 3.92 3.440
Merc 450SE         16.4   8 275.8 180 3.07 4.070
Merc 450SL         17.3   8 275.8 180 3.07 3.730
Merc 450SLC        15.2   8 275.8 180 3.07 3.780
Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250
Lincoln Continental 10.4  8 460.0 215 3.00 5.424
Chrysler Imperial  14.7   8 440.0 230 3.23 5.345
Fiat 128           32.4   4  78.7  66 4.08 2.200
Honda Civic        30.4   4  75.7  52 4.93 1.615
Toyota Corolla     33.9   4  71.1  65 4.22 1.835
Toyota Corona      21.5   4 120.1  97 3.70 2.465
Dodge Challenger   15.5   8 318.0 150 2.76 3.520
AMC Javelin        15.2   8 304.0 150 3.15 3.435
Camaro Z28         13.3   8 350.0 245 3.73 3.840
Pontiac Firebird   19.2   8 400.0 175 3.08 3.845
Fiat X1-9          27.3   4  79.0  66 4.08 1.935
Porsche 914-2      26.0   4 120.3  91 4.43 2.140
Lotus Europa       30.4   4  95.1 113 3.77 1.513
Ford Pantera L     15.8   8 351.0 264 4.22 3.170
Ferrari Dino       19.7   6 145.0 175 3.62 2.770
Maserati Bora      15.0   8 301.0 335 3.54 3.570
Volvo 142E         21.4   4 121.0 109 4.11 2.780
```

Hierarchical Clustering Algorithm

Parameters: Linkage



The chosen linkage algorithm affects which clusters are merged, and the shape of the resulting clusters (e.g. tighter, looser)

Strengths and Limitations – Linkages

Single Linkage (smallest distance)

- can handle non-blob shapes
- sensitive to noise and outliers
- produces elongated clusters

Complete Linkage (largest distance)

- balanced clusters, with similar diameters
- not overly sensitive to noise
- tends to split large clusters
- all clusters tend to have similar diameters

Strengths and Limitations – Linkages

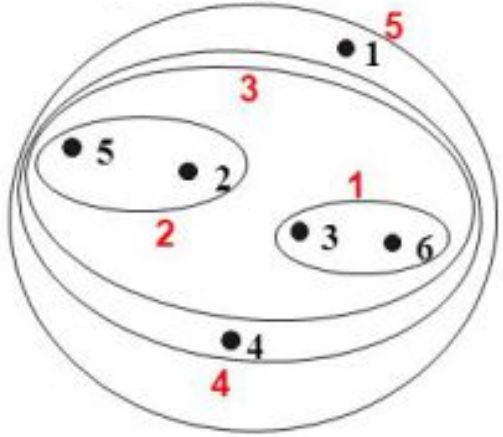
Average Linkage (average distance)

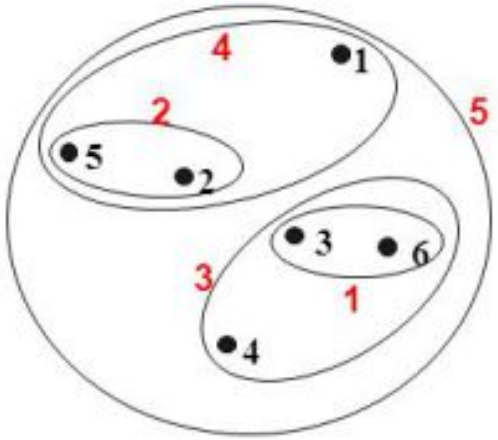
- compromise between single and complete linkages
- not too sensitive to noise and outliers
- tends to produce blob-shaped clusters

Centroid Linkage (centroid distance)

- clusters can have a lot of internal variance

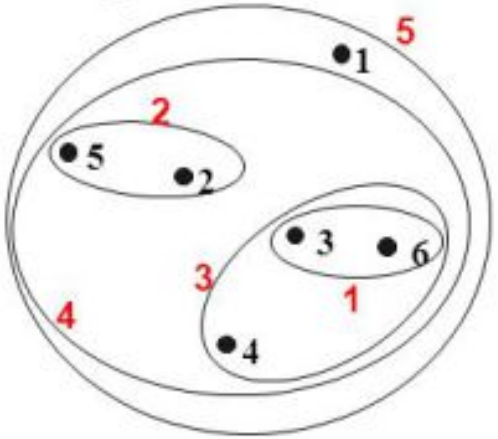
Simple Link

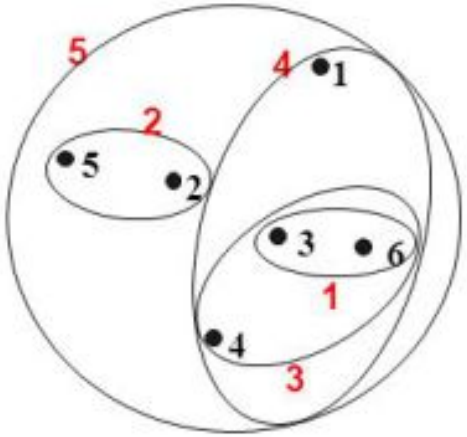




Complete Link

Average Link



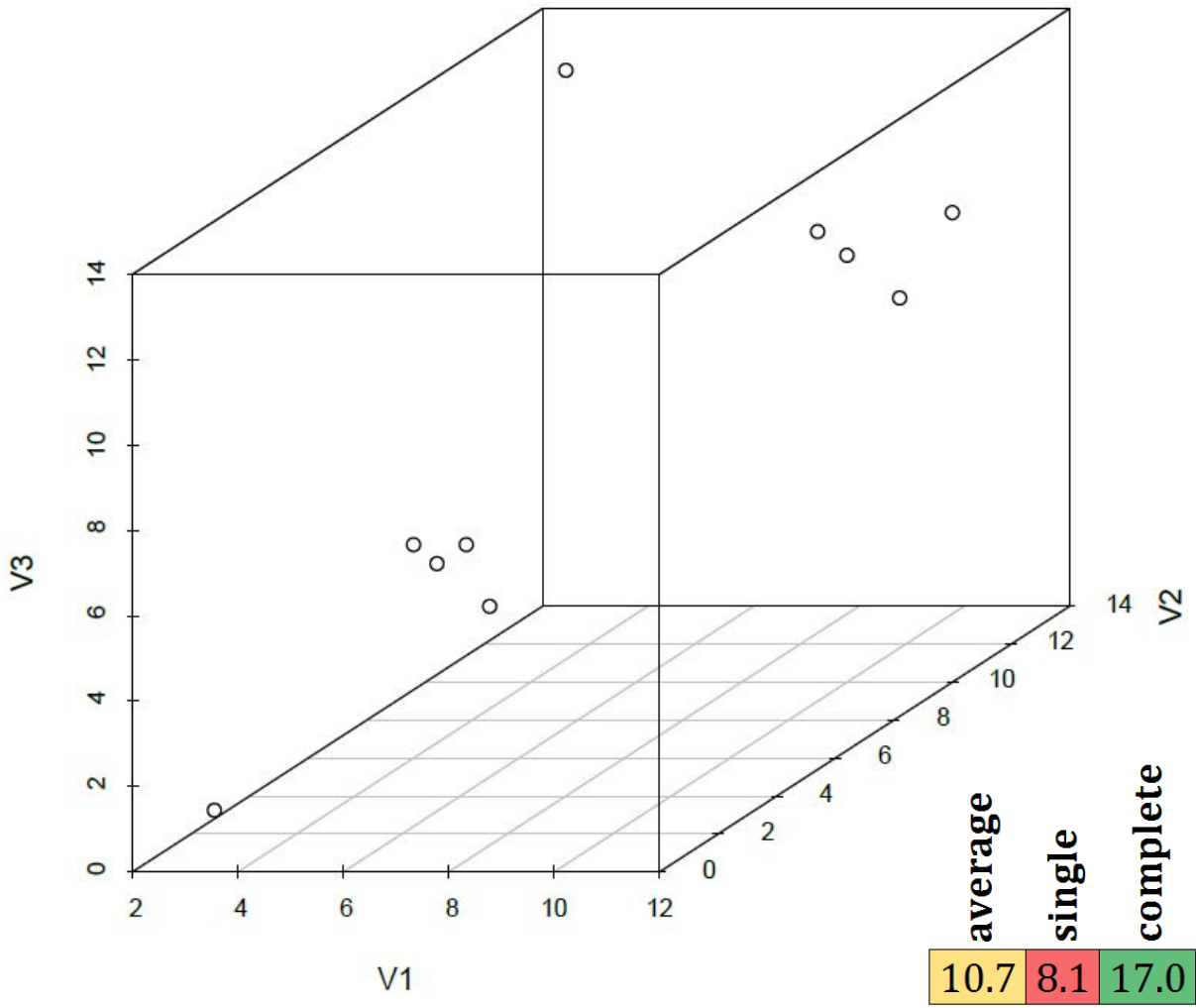


Centroid Link

[https://dataaspirant.com/2018/01/08/hierarchical-clustering-r/]

Hierarchical Clustering Algorithm

Parameters: Linkage – Example



Data

Cluster A	Cluster B
1: (5,5,5) 2: (5,6,5) 3: (4,6,5) 4: (6,5,4) 5: (3,1,1)	6: (10,10,10) 7: (11,10,9) 8: (12,10,11) 9: (10,9,11) 10: (13,13,13)
c: (4.6,4.6,4)	c: (11.2,10.4,10.8)

average	single	complete	centroid
10.7	8.1	17.0	11.1

Linkage

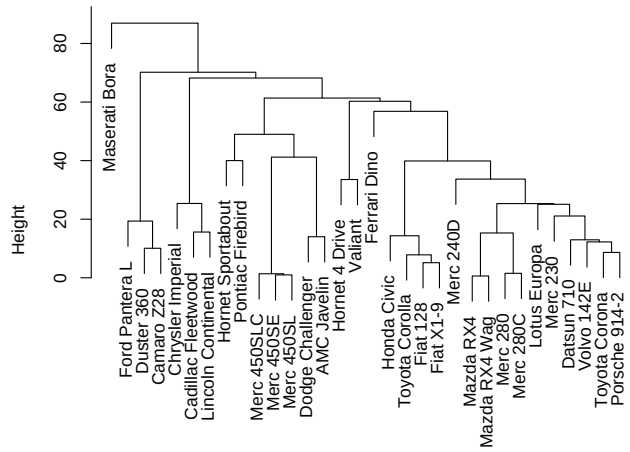
Cluster *A*

Cluster *B*	1	2	3	4	5
6	8.7	8.1	8.8	8.8	14.5
7	8.8	8.2	9.0	8.7	14.5
8	10.5	10.0	10.8	10.5	16.2
9	8.8	8.4	9.0	9.0	14.6
10	11.5	10.8	10.7	12.4	17.0

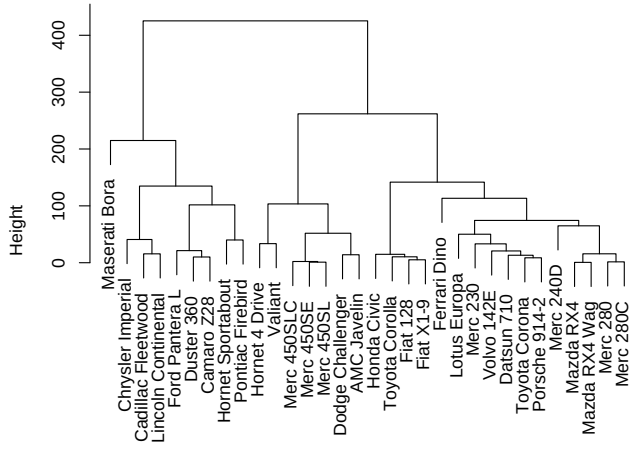
Distance matrix

Hierarchical Clustering Algorithm

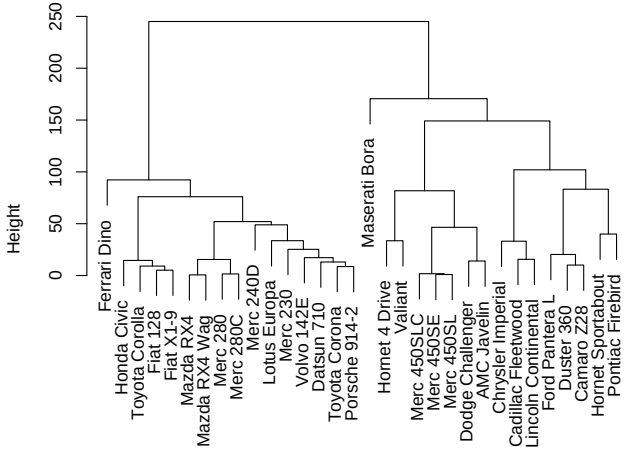
Returning to Our Clustering Results



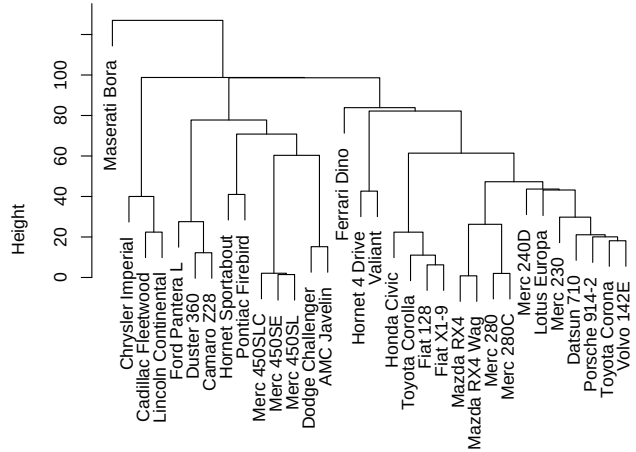
Euclidean Distance Single Link



Euclidean Distance Complete Link



Euclidean Distance Average Link



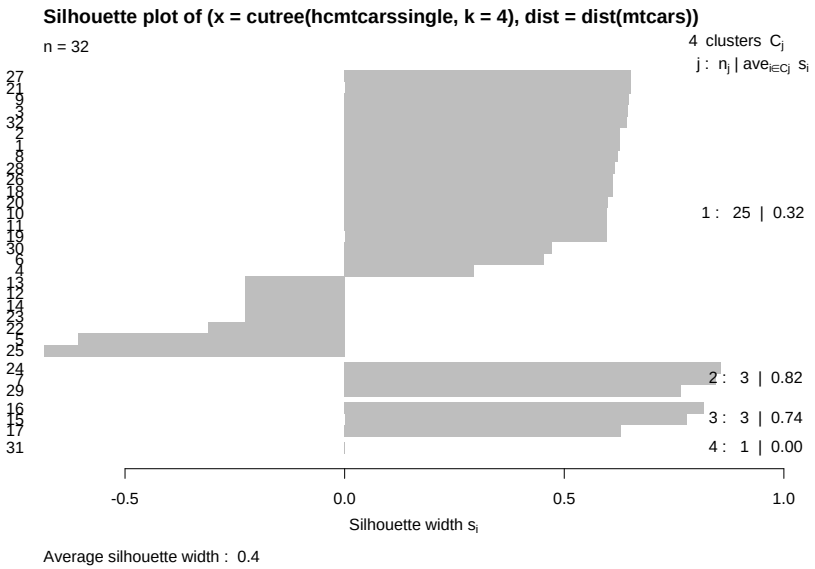
Manhattan Distance Single Link

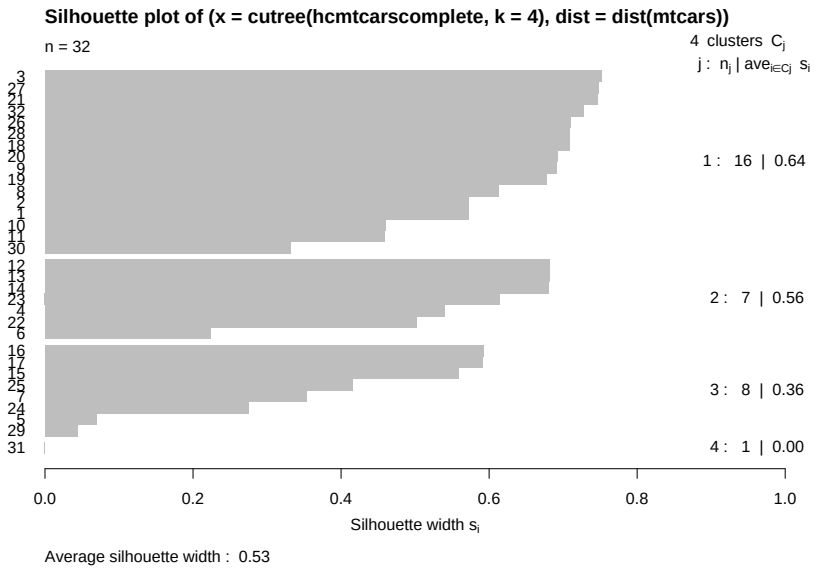
It is often said: choose the distance metric that is most **meaningful** for your data.

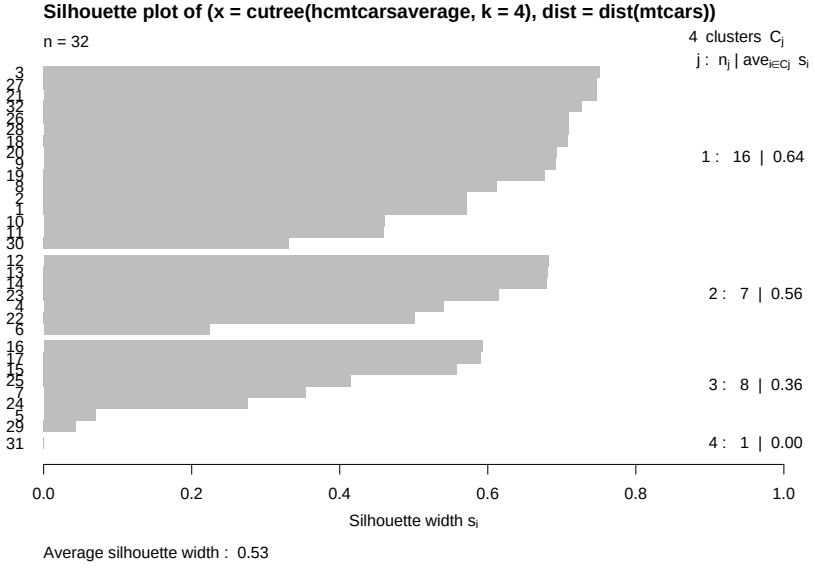
Can evaluation metrics also inform this choice?

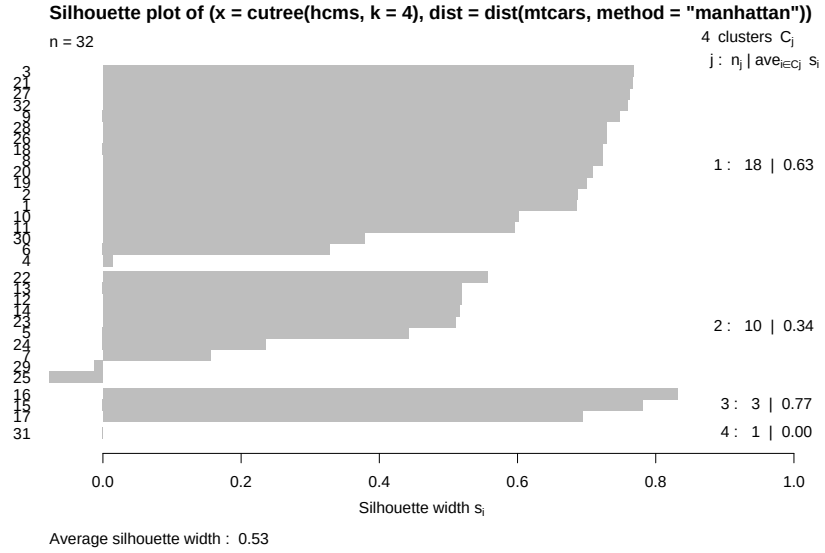
Hierarchical Clustering Algorithm

Silhouette of Clustering Results







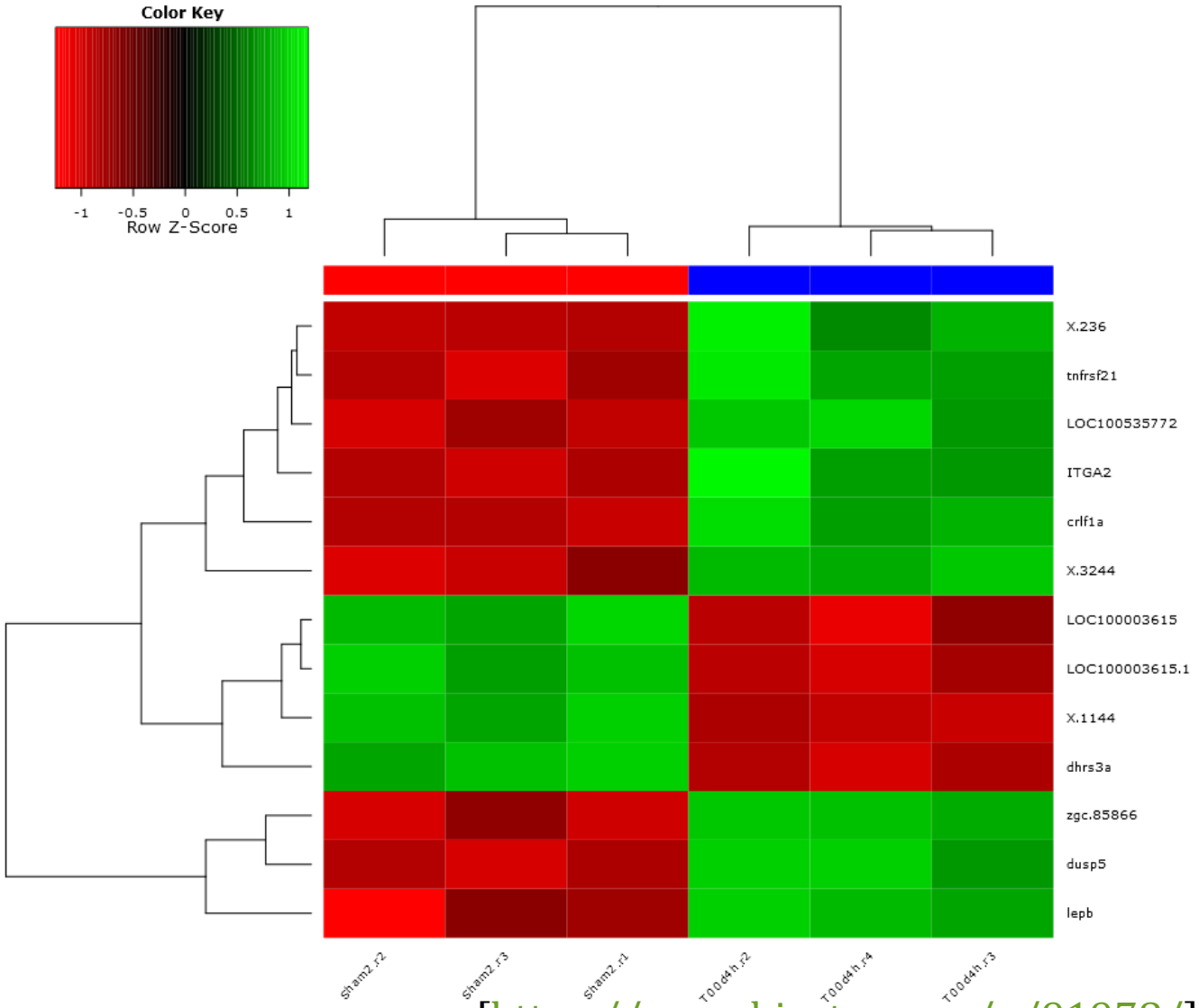


Hierarchical Clustering Notes

HC is deterministic, for a given choice of metric and linkage.

Space and time requirements do make HC unattractive for medium-to-large datasets.

Various linkage strategies: be sure to check out **Wald's method**!



[https://www.biostars.org/p/91978/]

Hierarchical Clustering Notes

Easy to understand and implement, but rarely optimal.

No real strong theoretical or first principle approach to specify the distance metric and linkage criteria (arbitrary decisions).

Cannot handle missing values or mixed data types.

Dendrograms can only be used to select the number of clusters when the ultrametric tree inequality holds (rarely does in practice).

Consider using **latent class analysis** instead.

HC Examples and Case Studies
Clustering Myths

Comparative Mythology

- Studying myths from different cultures to understand their similarities and possibly shared origins
- Many myths have splintered off and evolved from common sources

Julien d'Huy (2016): Used a variety of data mining techniques, including hierarchical clustering, to trace the evolution of myths.

Collection of myths broken down into common story elements.

A myth across cultures: the hunter in the sky

J.d'Huy [2016], *Scientists Trace Society's Myths to Primordial Origins*, Scientific American (Online)

HC Examples and Case Studies
Clustering Myths

Myths categorized based on presence/ absence of elements

Myths are clustered based on this categorization.

Result shows myths clustering together – could this suggest a possible common origin for these myths?

Remember, **clustering is knowledge discovery**!

A myth across cultures: the hunter in the sky

J.d'Huy [2016], *Scientists Trace Society's Myths to Primordial Origins*, Scientific American (Online)

HC Examples and Case Studies

Complex building's energy system operation patterns analysis using bag of words representation with hierarchical clustering

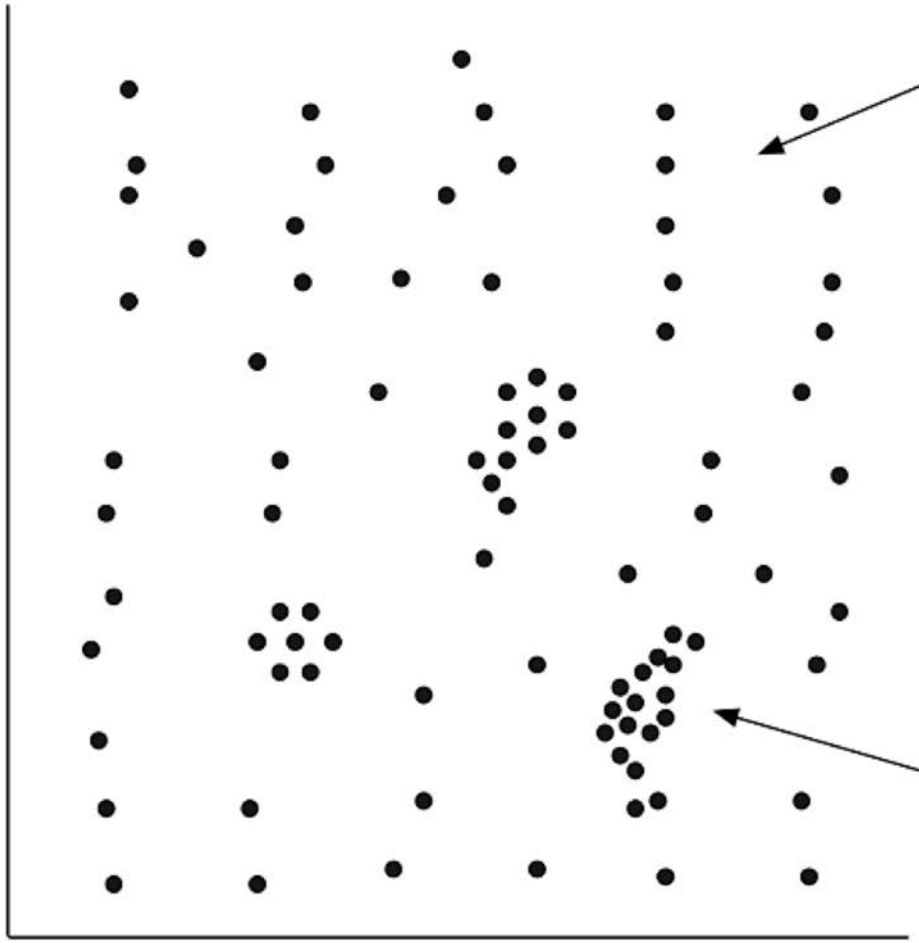
A Comparison of Antioxidant, Antibacterial, and Anticancer Activity of the Selected Thyme Species by Means of Hierarchical Clustering and Principal Component Analysis

Use of hierarchical cluster analysis to classify prisons in Ireland into mutually exclusive drug-use risk categories

Divisive Analysis (DIANA) of hierarchical clustering and GPS data for level of service criteria of urban streets

Density Based Clustering

Data Point Density



Area of low density

In density-based clustering, the **density** of observations determines the clusters.

How do we measure density?

Area of high density

DBSCAN Algorithm – Parameters

DBSCAN uses 2 parameters:

- a **distance** parameter to create ε-neighbourhoods, and
- the **minimum number of points** in an ε-neighbourhood required to include the n'hood in the cluster being constructed (including the centre)

3 distinct types of points:

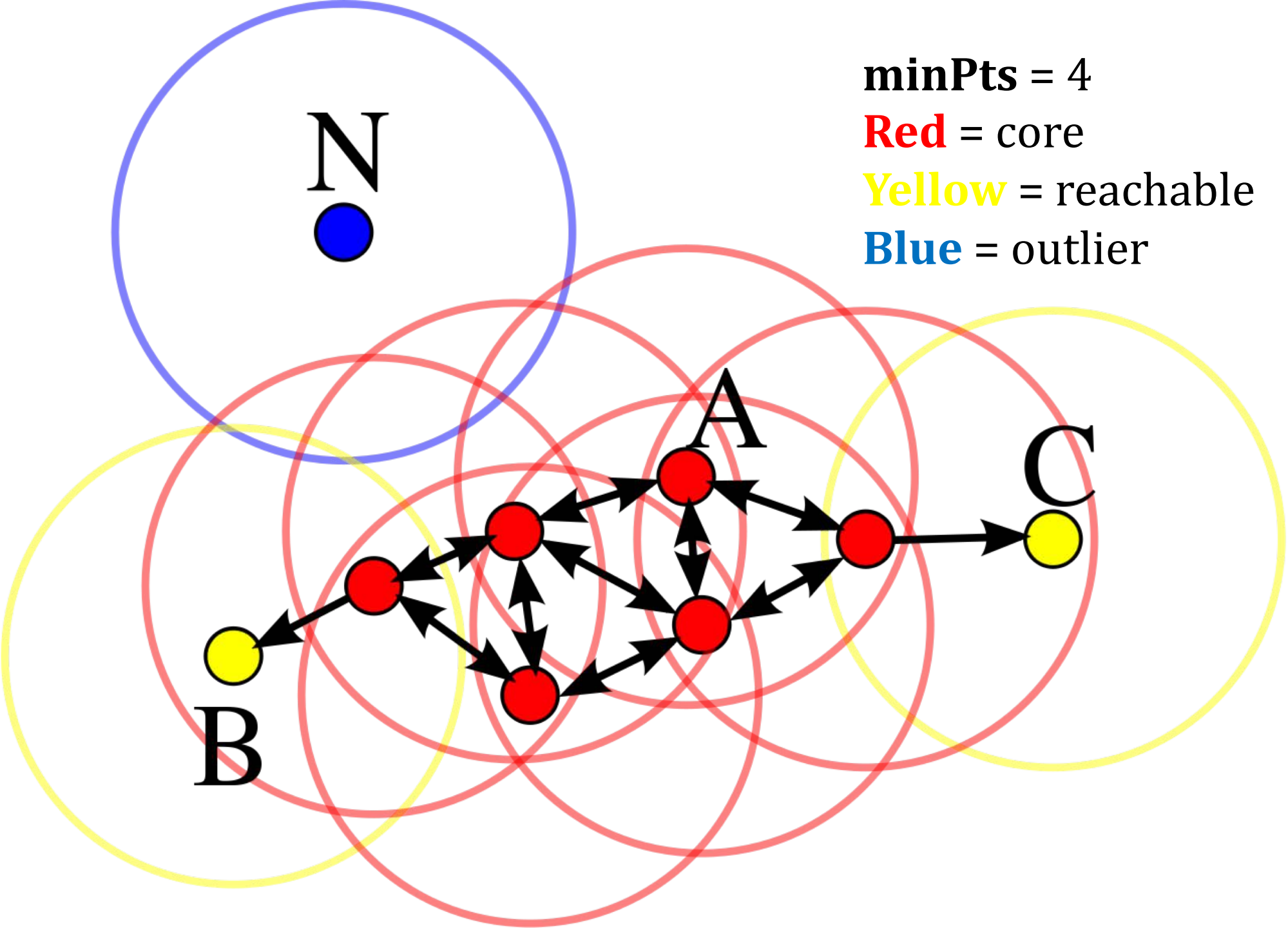
- **outliers**: out of reach of every other point
- **non-core** (**reachable**): within reach of some number of points below the min. threshold
- **core**: within reach of at least the minimum number of other points

DBSCAN Algorithm – Parameters

Reachability is not a symmetric relation: **no point is reachable from a non-core point** (a non-core point may be reachable, but nothing can be reached from it).

Two points *p* and *q* are **density-connected** if there is a point *o* such that both *p* and *q* are reachable from *o* (but density-connectedness *is* symmetric).

All points within a cluster are mutually density-connected. If a point is density-reachable from any point of the cluster, **it is part of the cluster as well**.


minPts = 4
Red = core
Yellow = reachable
Blue = outlier
N
A
B
C

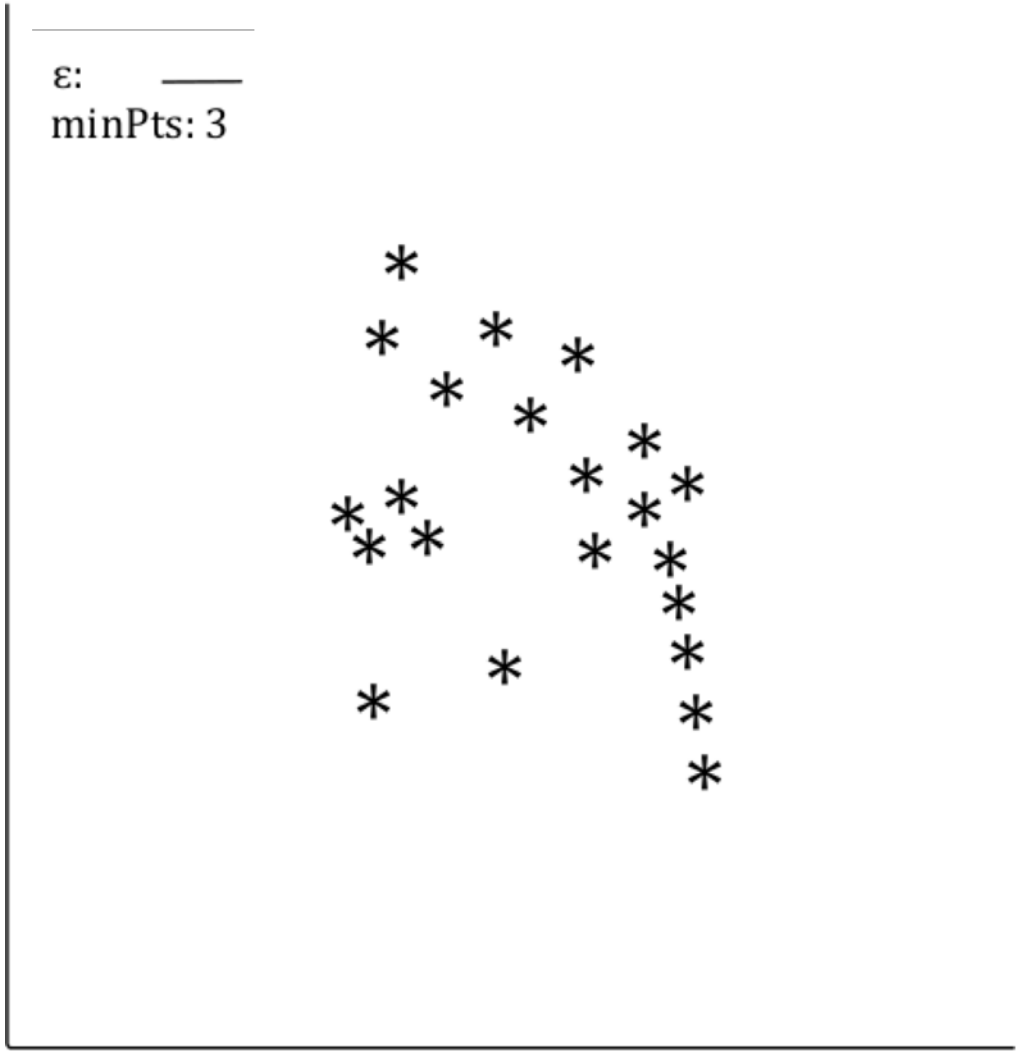
DBSCAN Algorithm

Given $\varepsilon > 0$ and minPts (as well as a distance metric d):

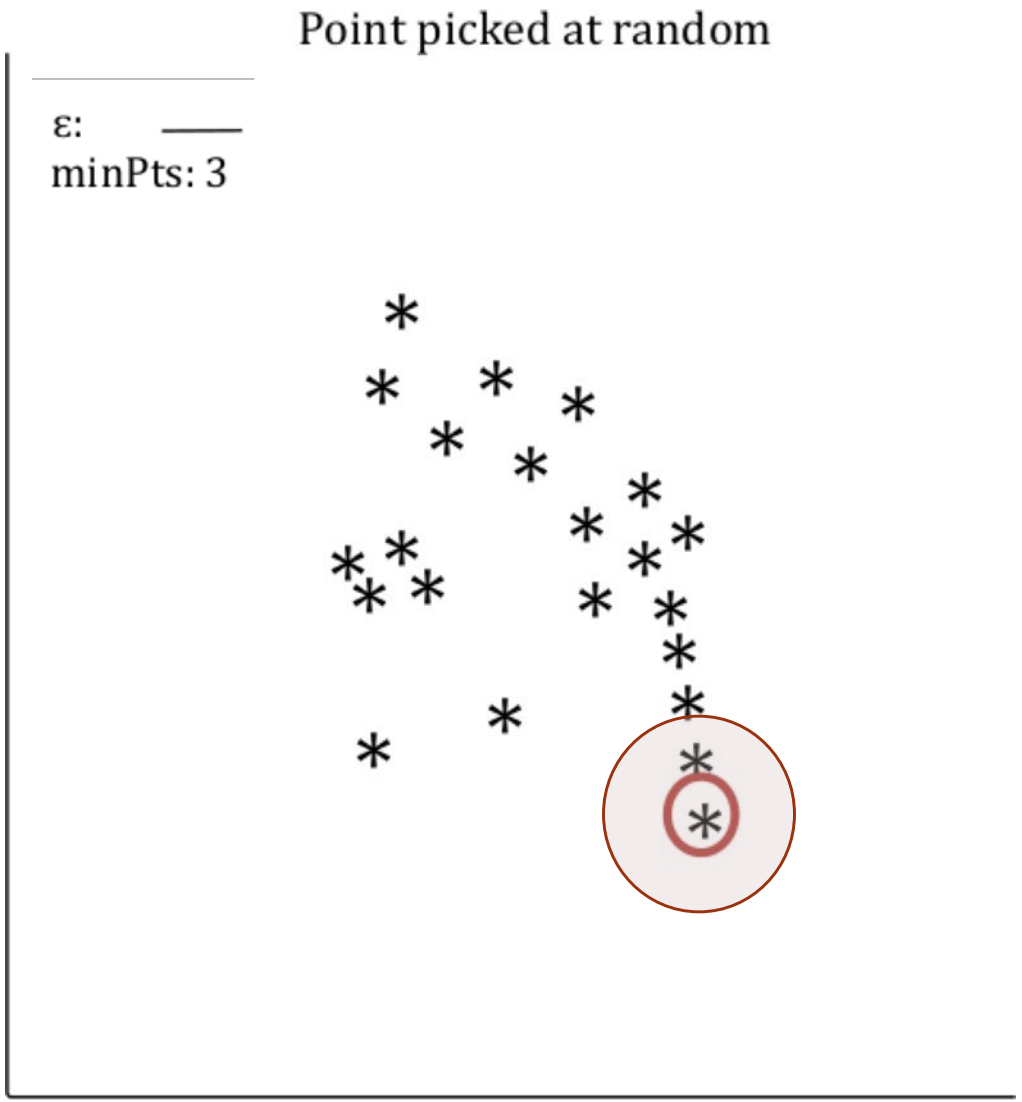
1. Find the ε-neighbours of every point, and identify the core points with more than minPts neighbours (including the core point).
2. Find the connected components of **core** points on the neighbor graph, ignoring all non-core points.
3. Assign each non-core point to a nearby cluster if the cluster is an ε neighbor, otherwise assign it to noise.

That's really all there is to it…

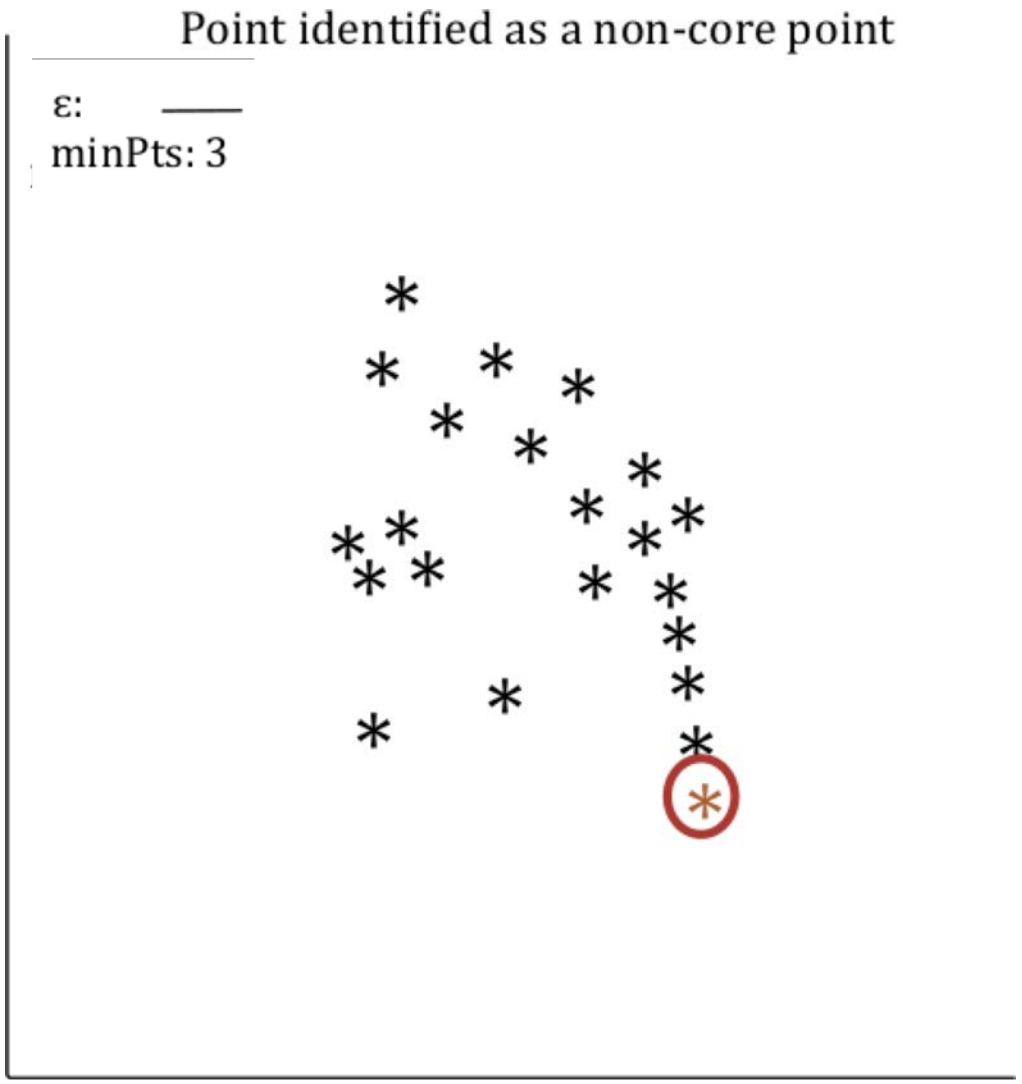
DBSCAN Example – Artificial Dataset



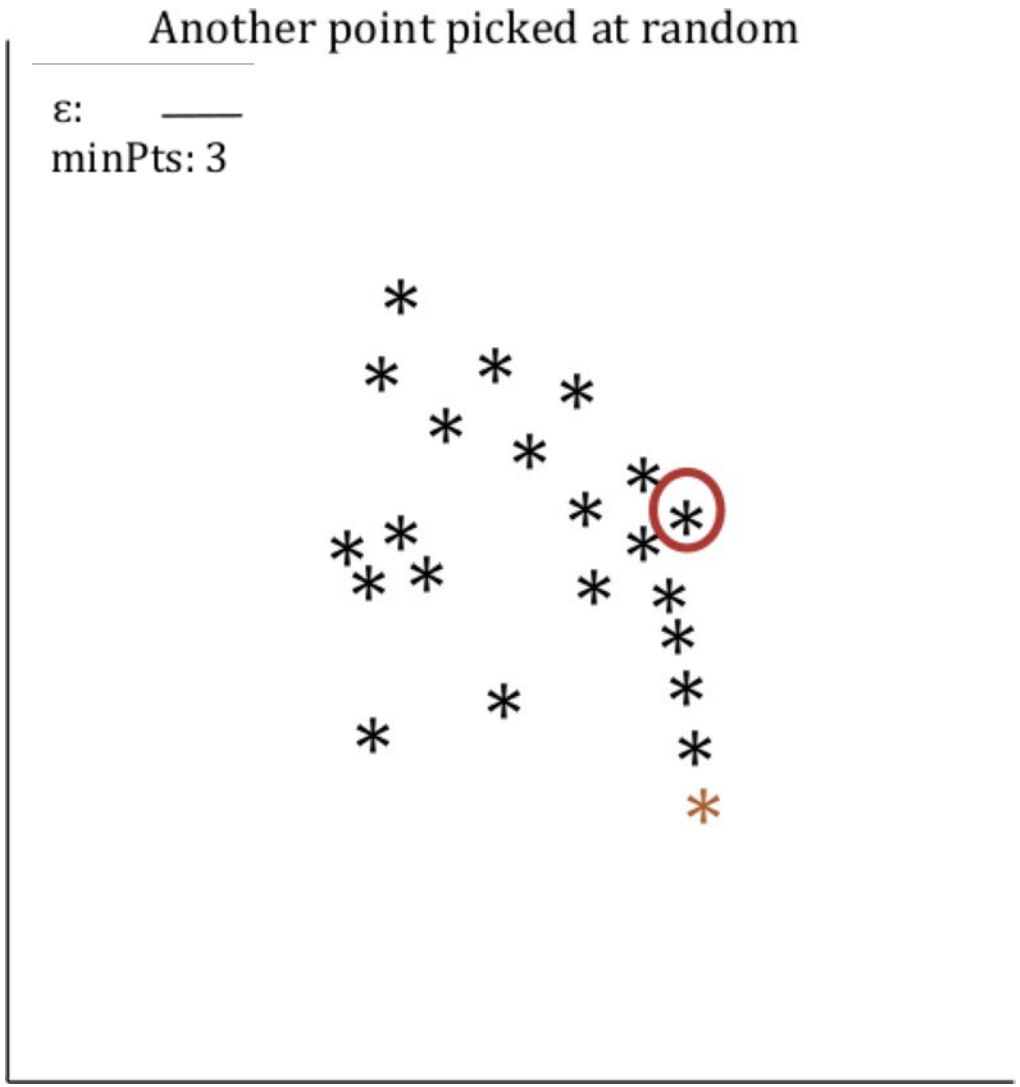
DBSCAN Example – Artificial Dataset



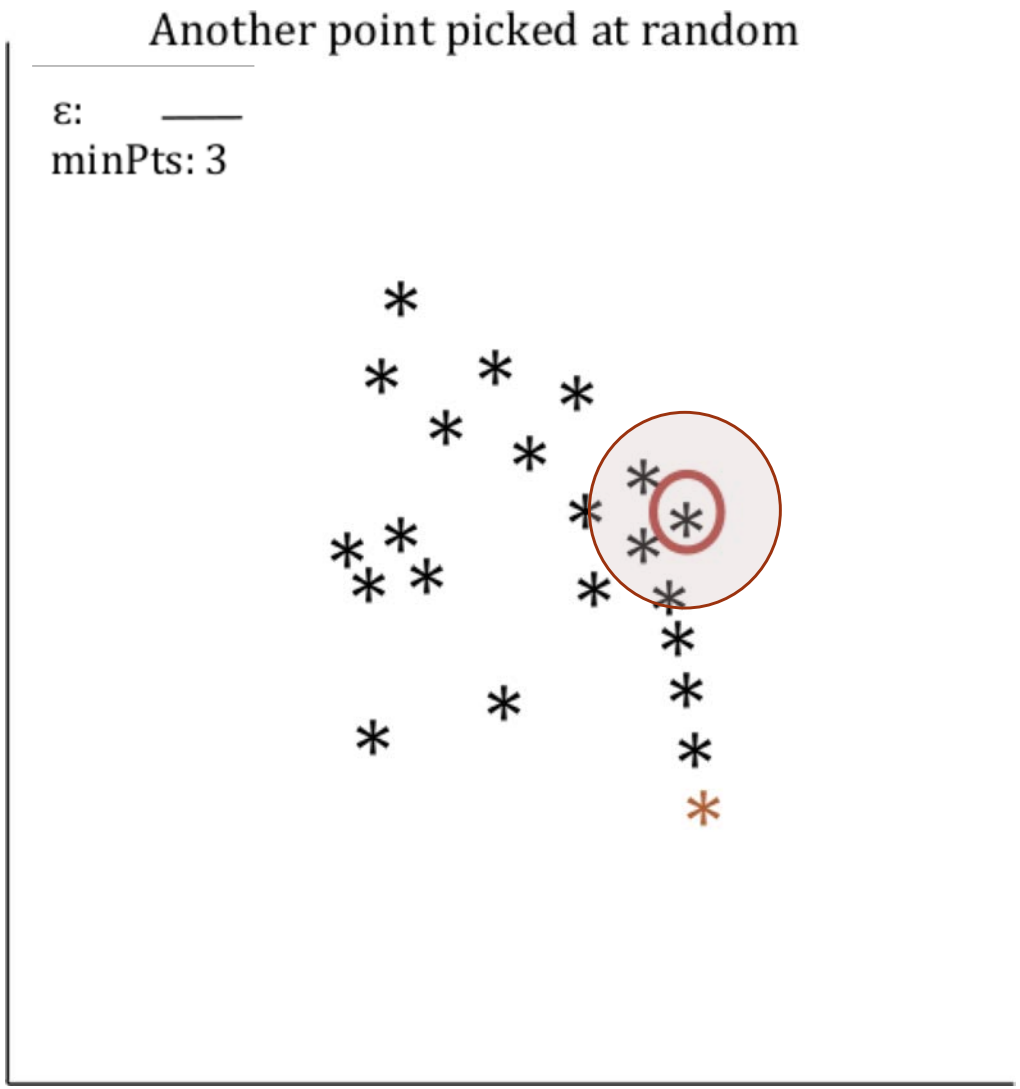
DBSCAN Example – Artificial Dataset



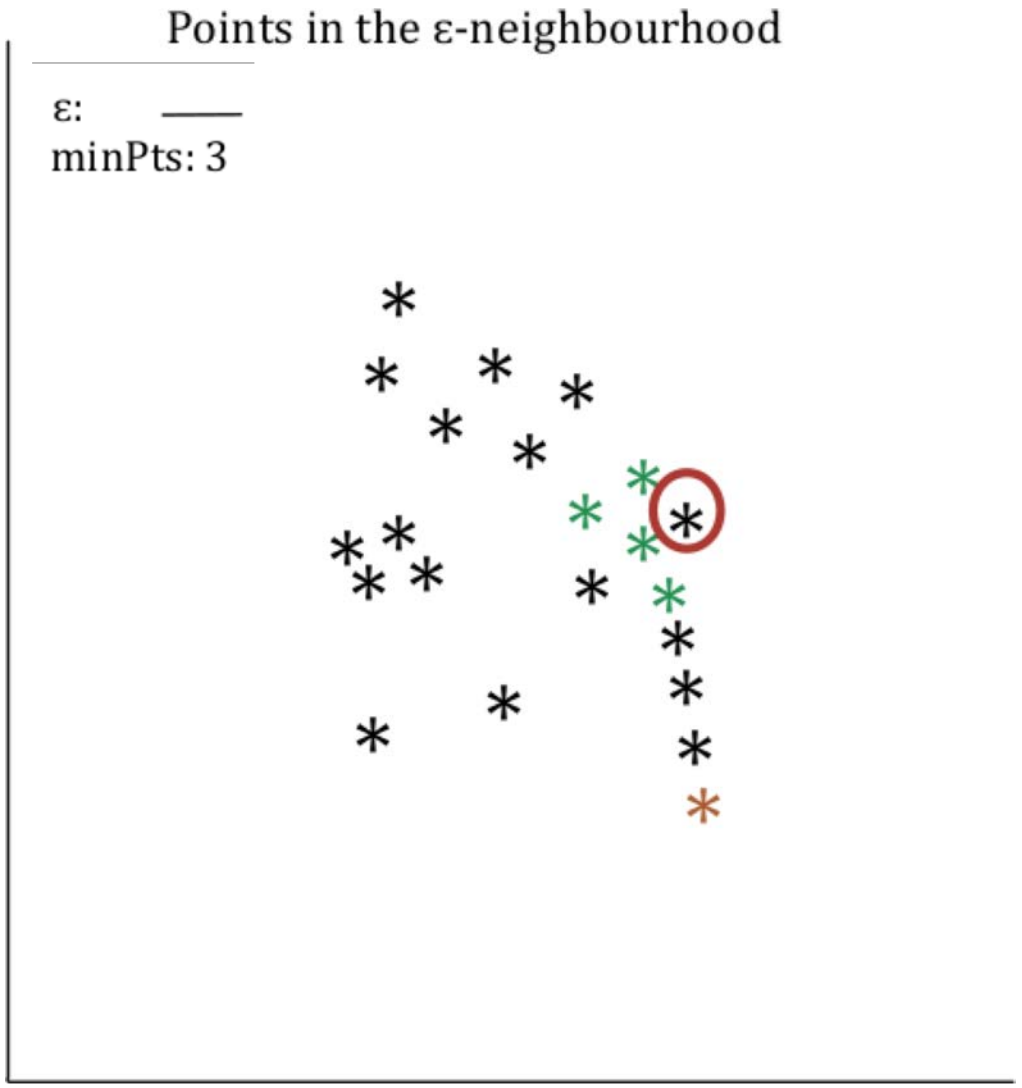
DBSCAN Example – Artificial Dataset



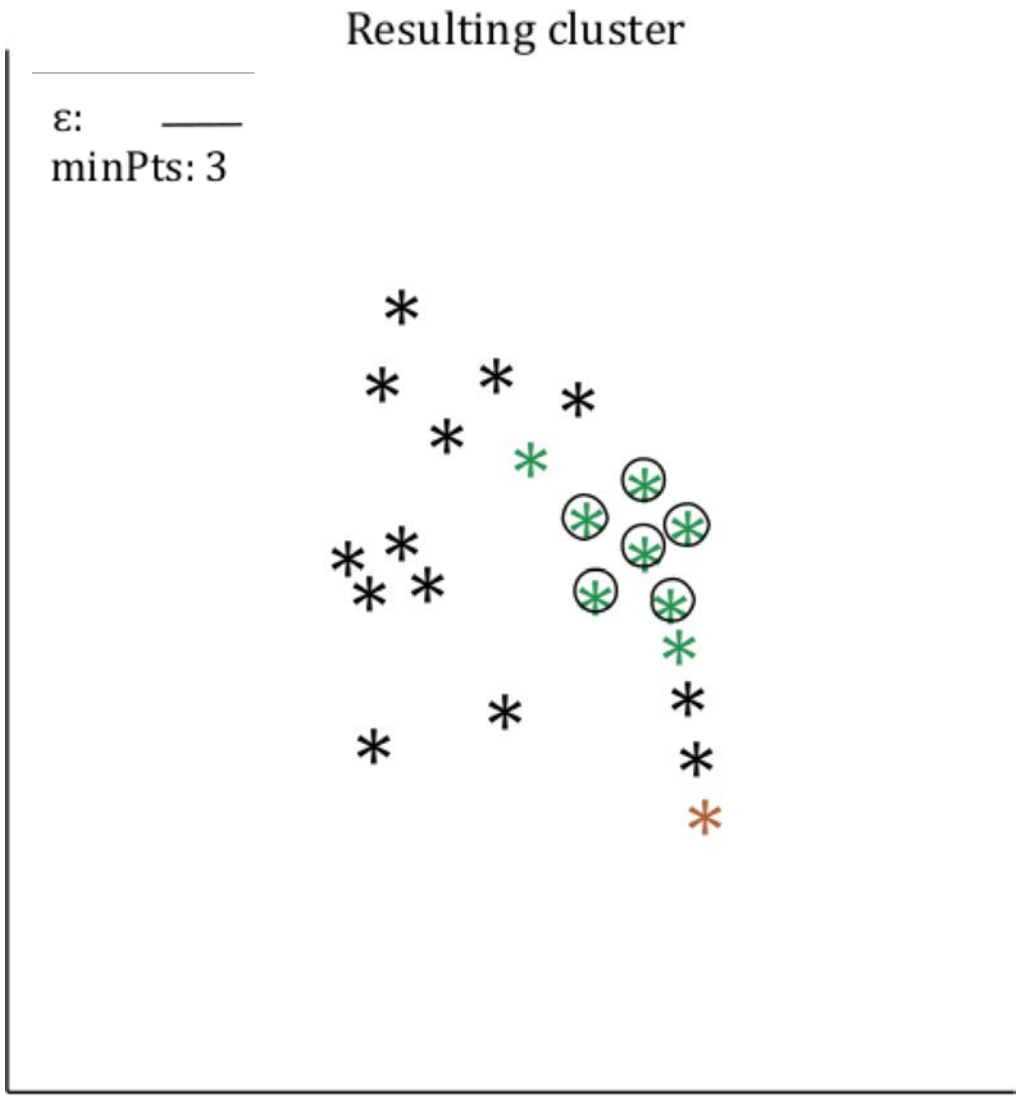
DBSCAN Example – Artificial Dataset



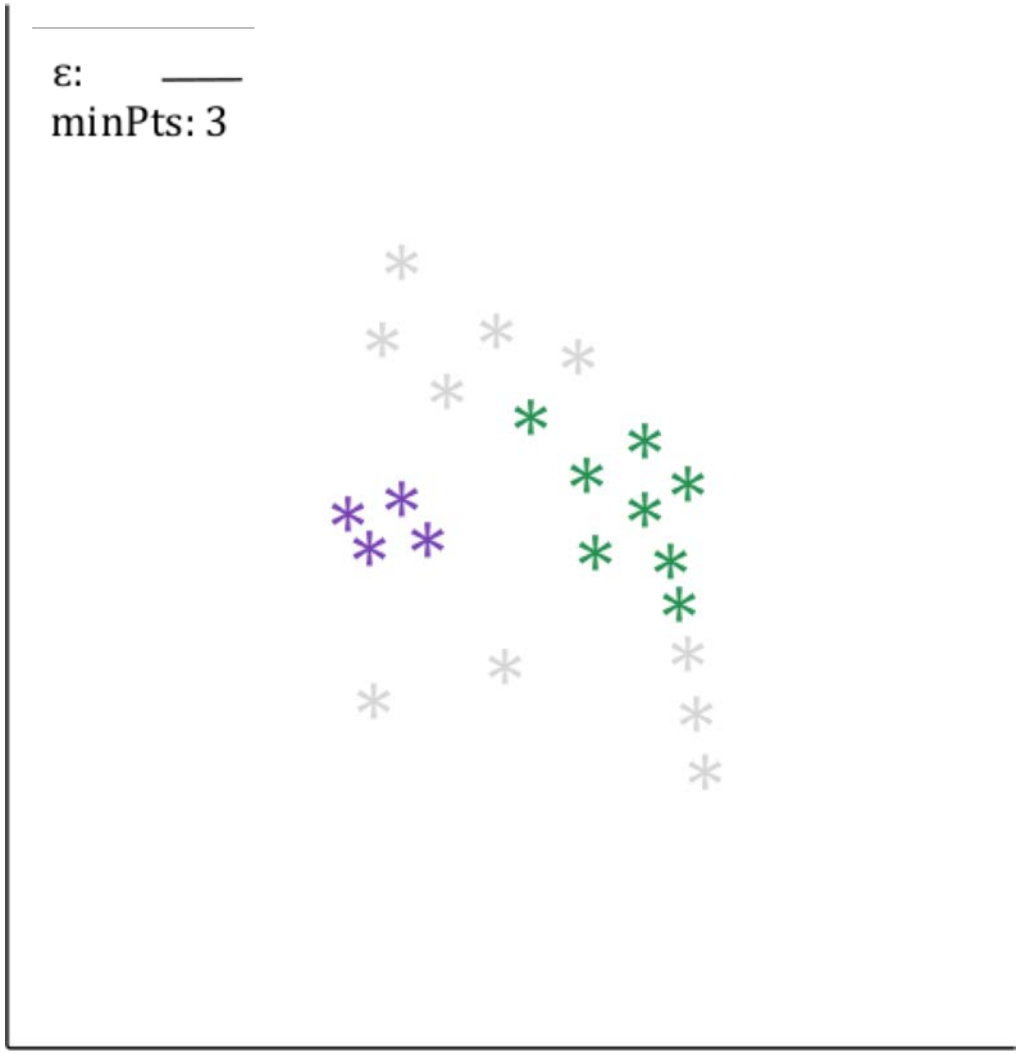
DBSCAN Example – Artificial Dataset

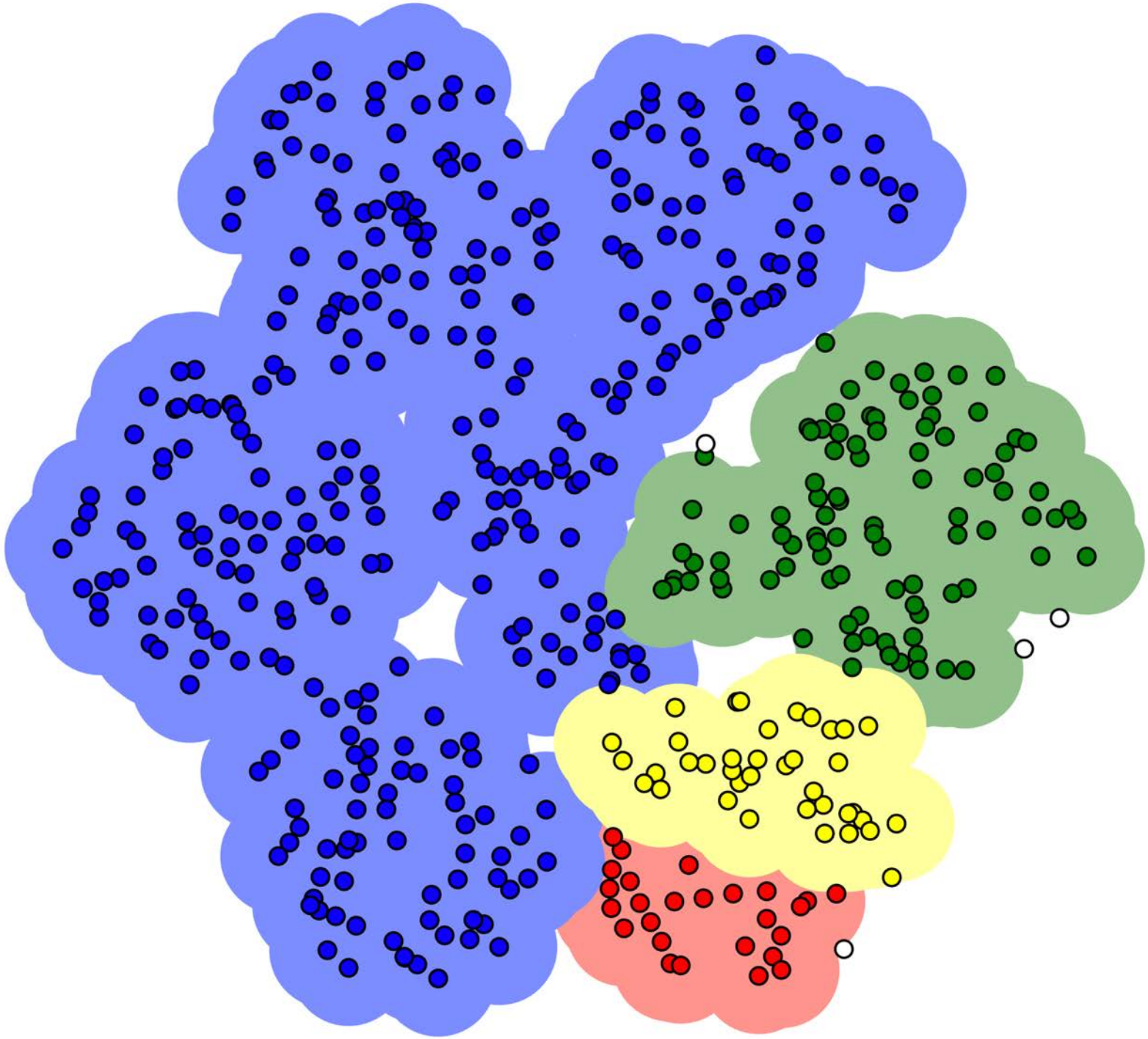


DBSCAN Example – Artificial Dataset

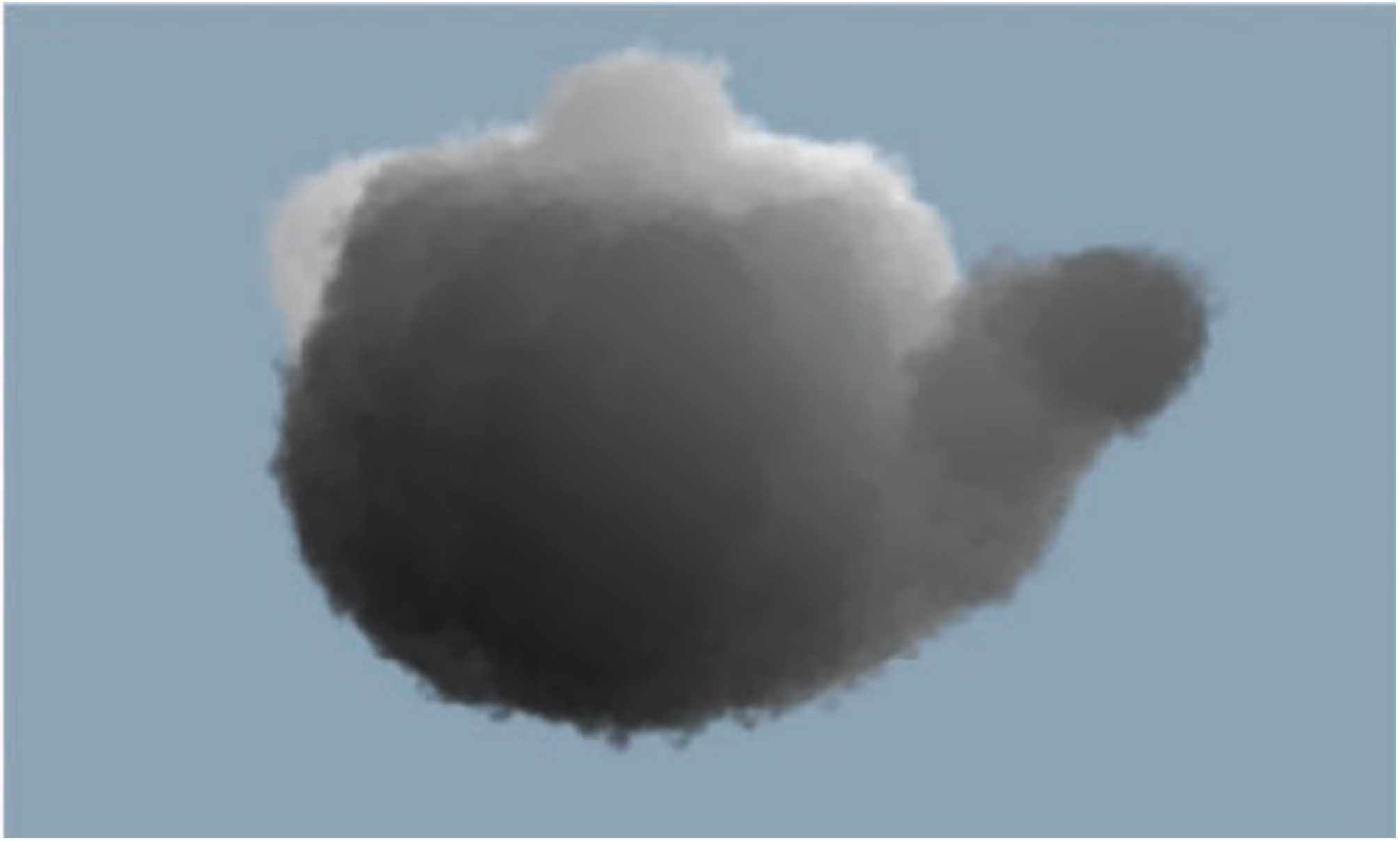


DBSCAN Example – Artificial Dataset





[https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/]

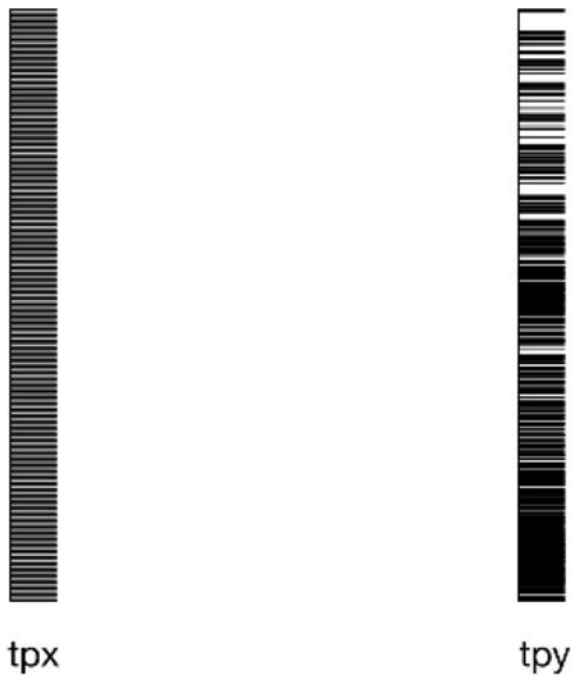


[Adapted from https://library.creativecow.net/articles/ussing_jonas/clouds_3dmax.php]

DBSCAN Clustering Challenge

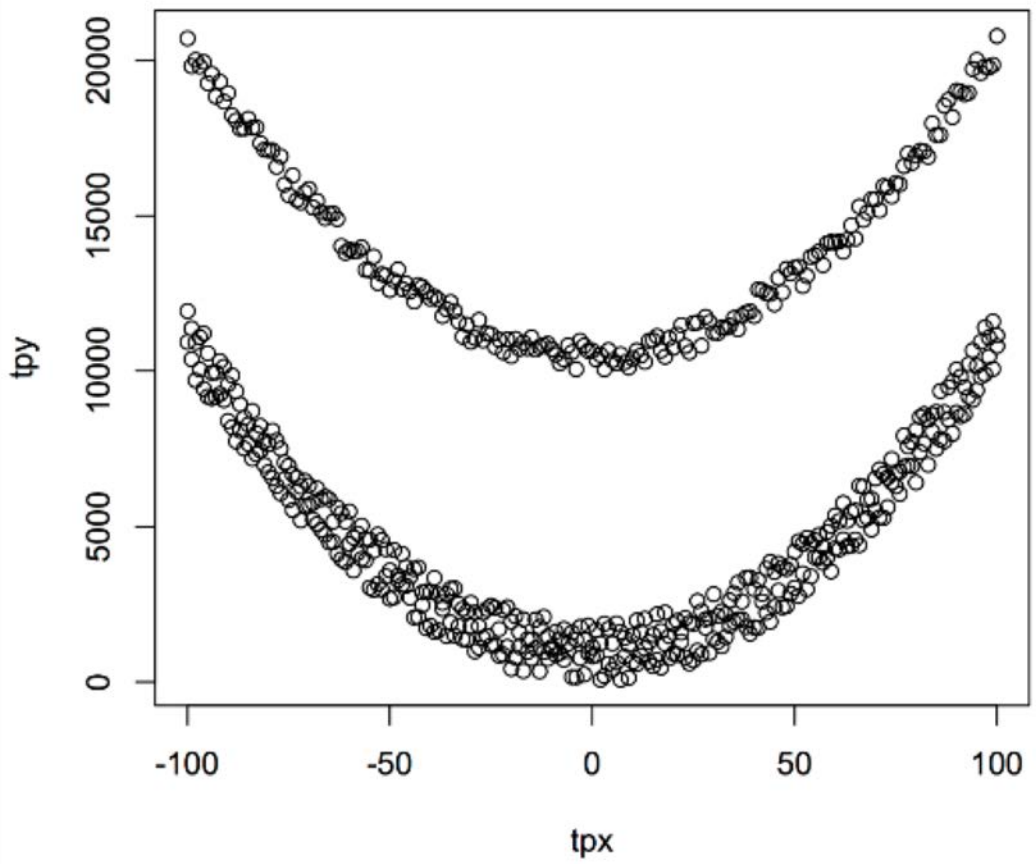
	tpx	tpy
[1,]	-100	10928.249
[2,]	-99	10376.446
[3,]	-98	9696.948
[4,]	-97	10049.223
[5,]	-96	9420.883
[6,]	-95	9171.636
[7,]	-94	9118.230
[8,]	-93	9166.522
[9,]	-92	9251.633
[10,]	-91	9059.243
[11,]	-90	8390.208
[12,]	-89	8186.269
[13,]	-88	7749.231
[14,]	-87	8092.249
[15,]	-86	7518.351
[16,]	-85	7674.044
[17,]	-84	7194.916
[18,]	-83	7340.763
[19,]	-82	7456.145
[20,]	-81	6990.375

First 20 of 603 data points from an artificially-constructed dataset.



1 dimensional plot of points from each column.

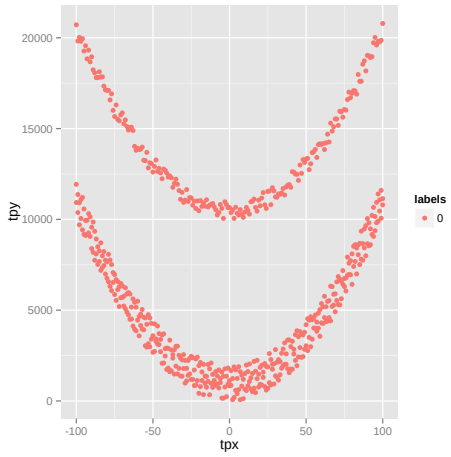
DBSCAN Clustering Challenge



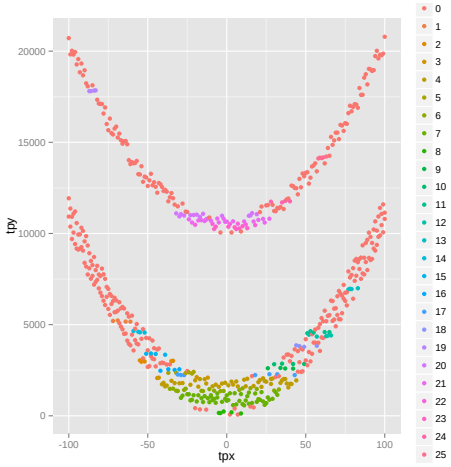
This looks like something DBSCAN should be able to handle…

… and better than k-means, too.

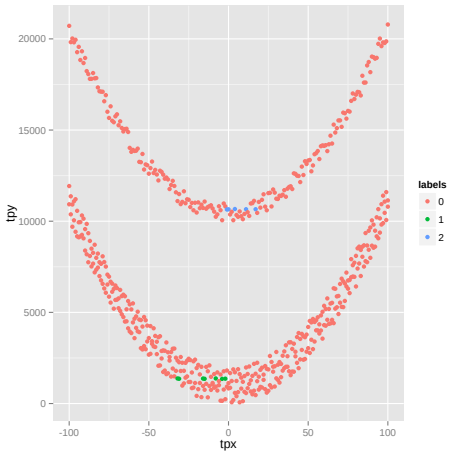
DBSCAN Clustering Challenge



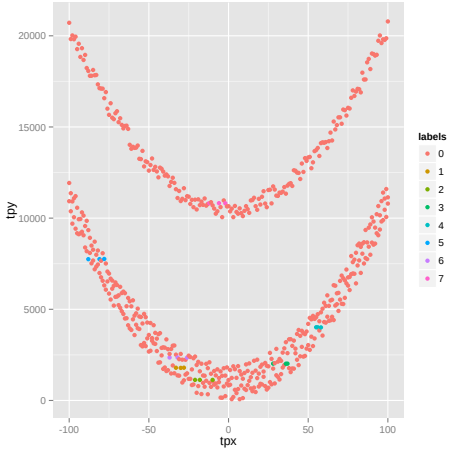
eps = 0.5 , minpts = 10 ,
0 clusters, 603 noise points



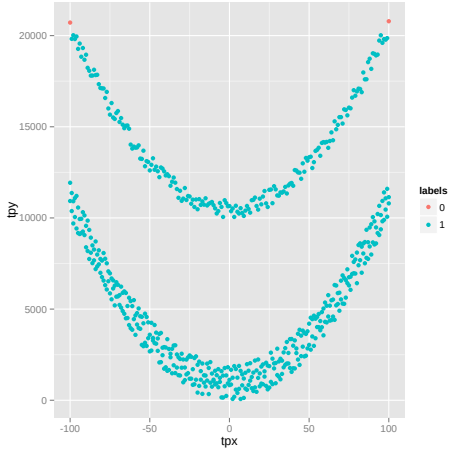
eps = 50, minpts = 4,
25 clusters, 357 noise points



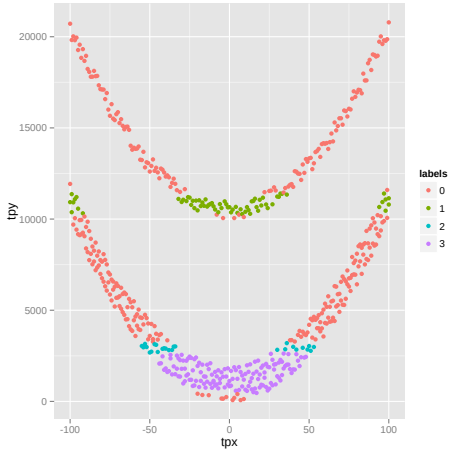
eps = 20, minpts = 5 ,
2 clusters, 591 noise points



eps = 10, minpts = 3,
7 clusters, 582 noise points



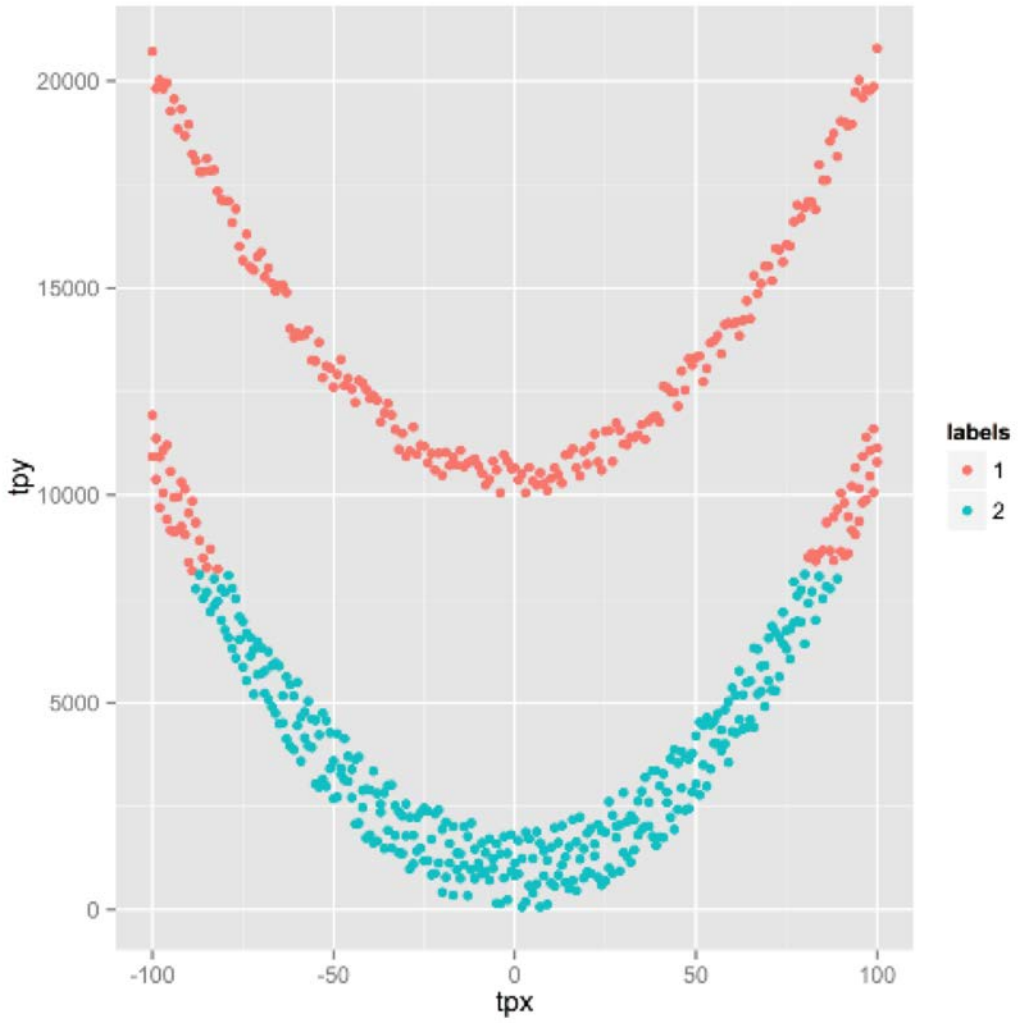
eps = 500, minpts = 5 ,
1 cluster, 2 noise points



eps = 200, minpts = 20 ,
3 clusters, 357 noise points

But it turns out that DBSCAN isn't working out so well for this dataset…

DBSCAN Clustering Challenge

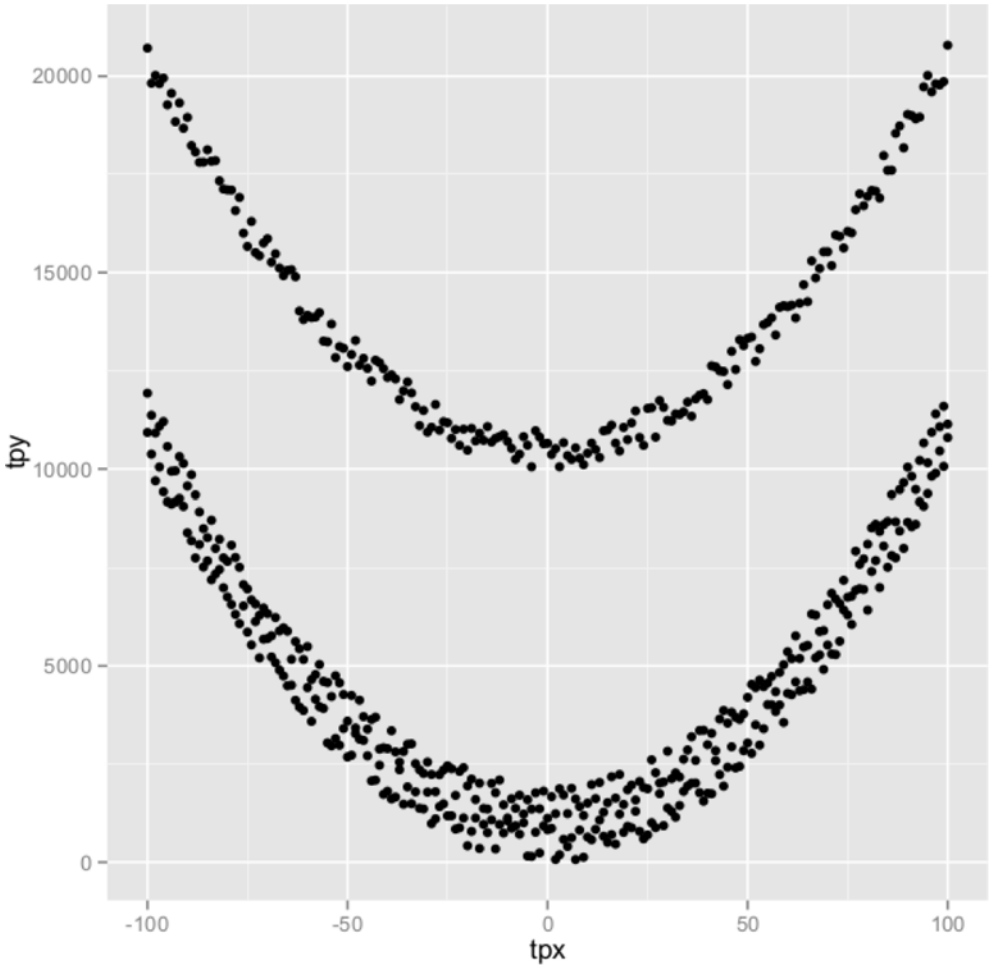


k-means appears to be doing a better job.

But is it really detecting the clusters more accurately, or just taking advantage of the separation between the two clusters?

Is there a way to get DBSCAN to work?

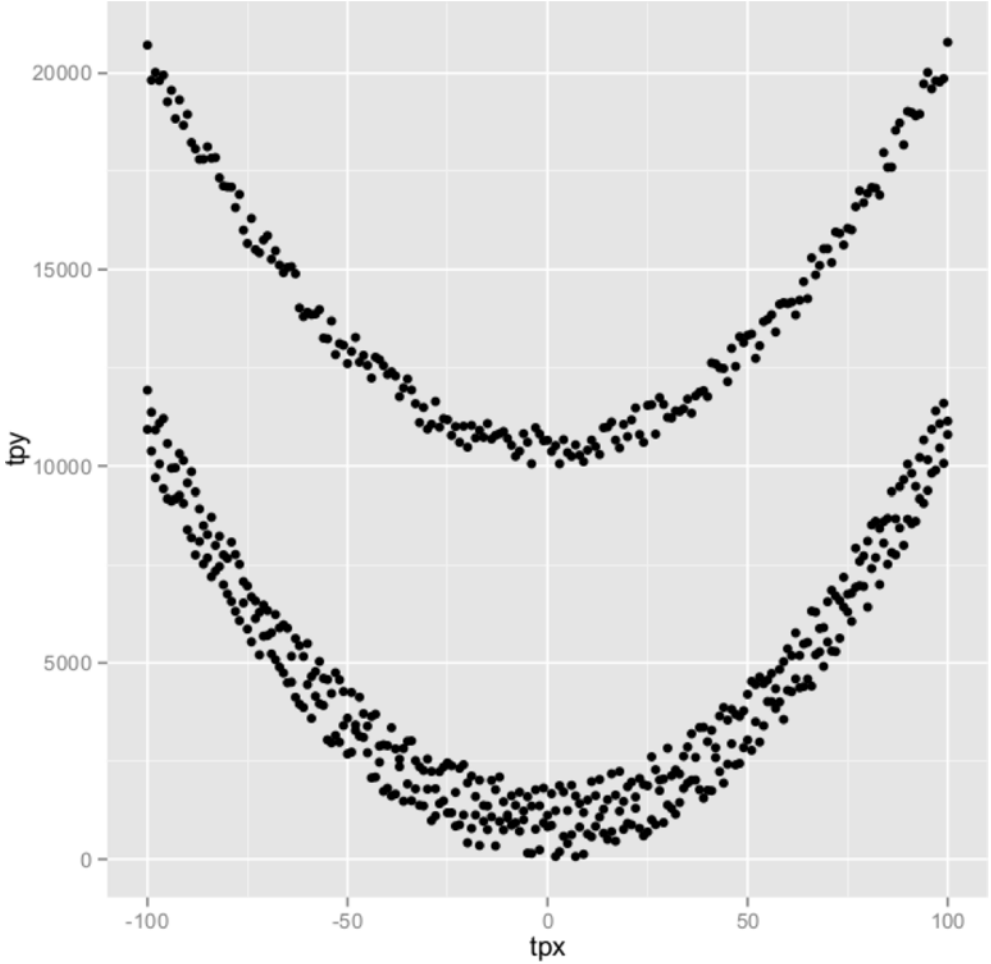
DBSCAN Clustering Challenge

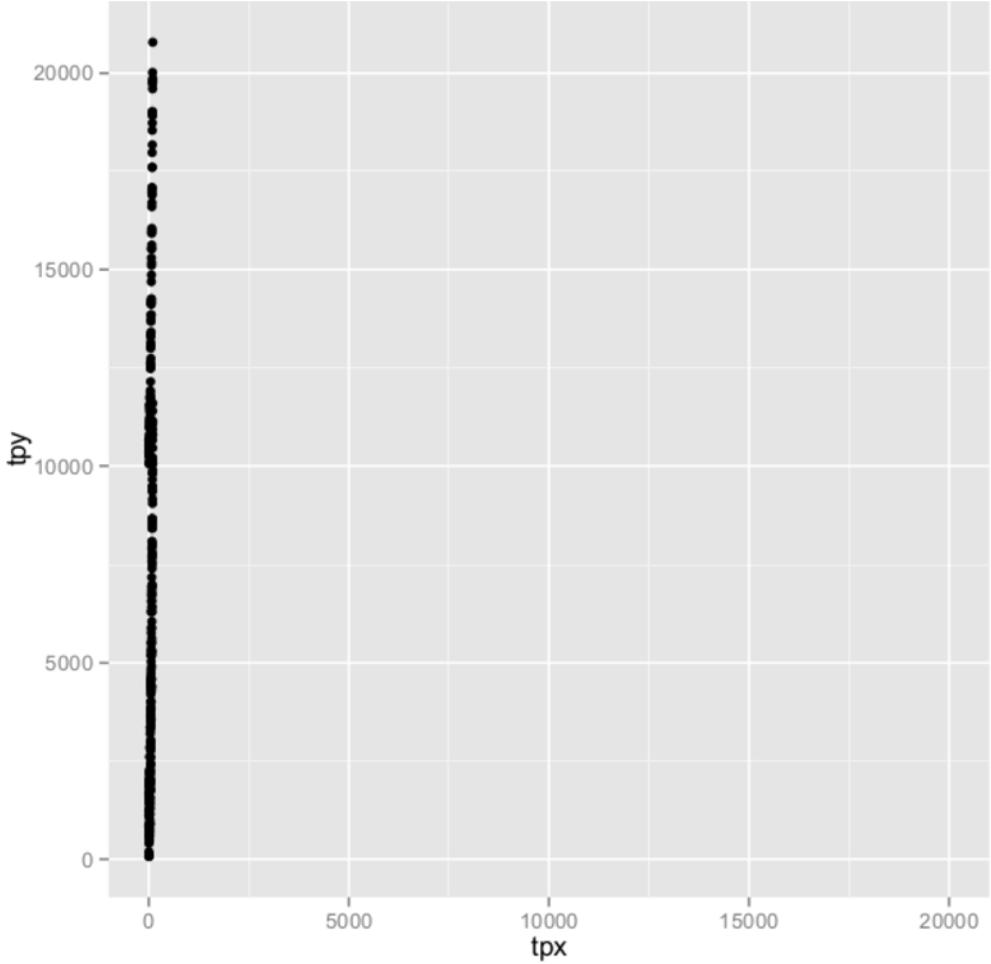


Take a closer look at the axes on this plot…

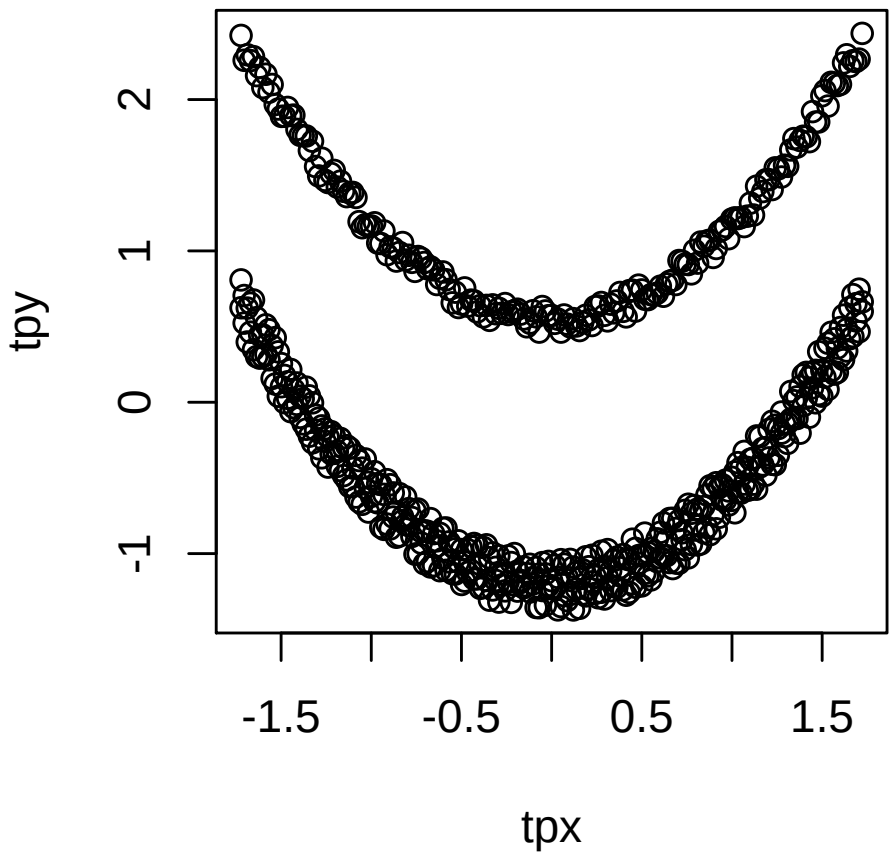
DBSCAN Clustering Challenge

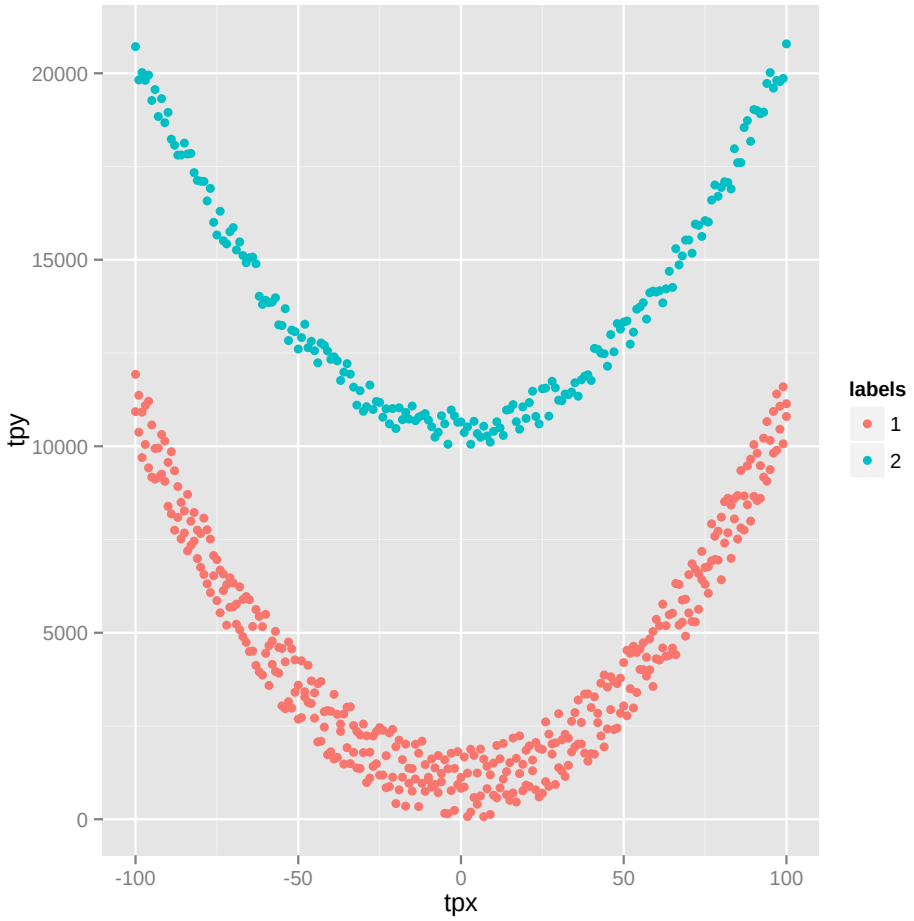
Re-plotted, with the axis adjusted to match the axis:





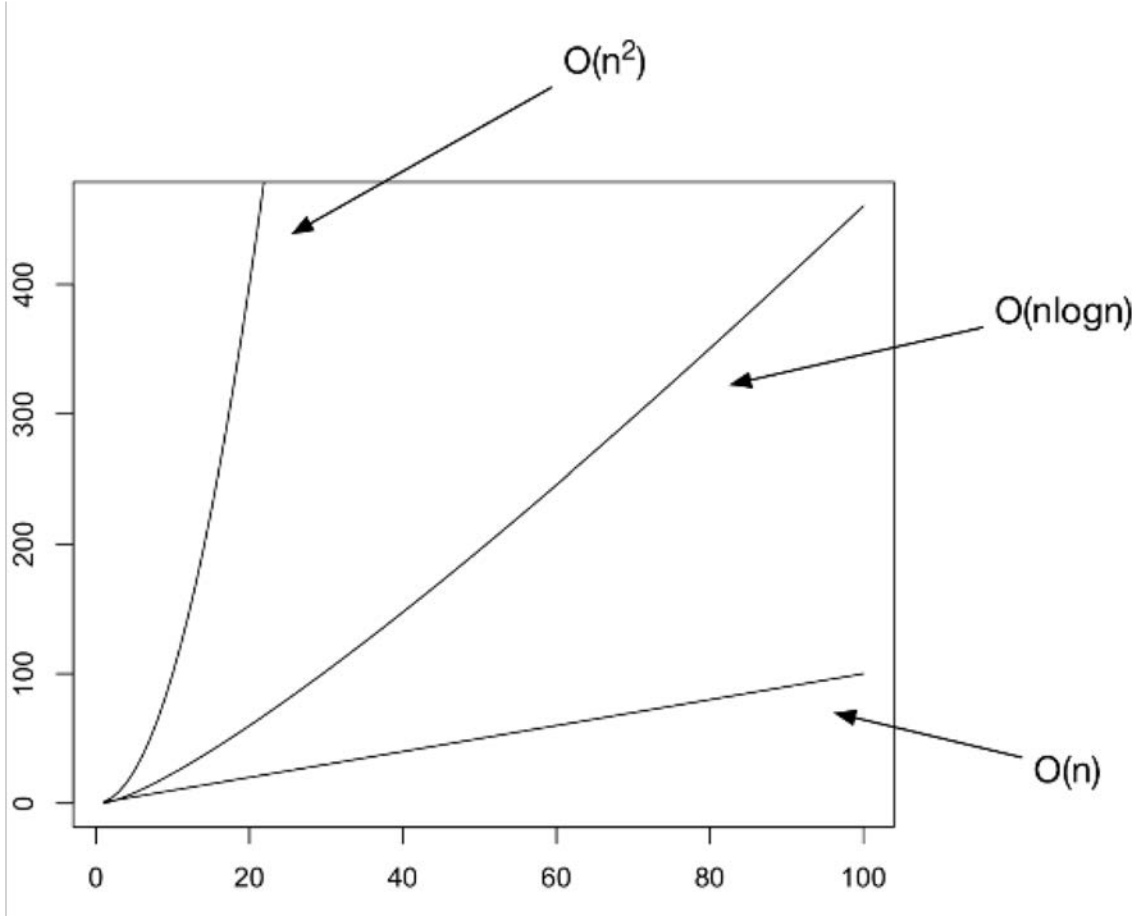
DBSCAN Clustering Challenge





When data is scaled, position and length of vectors are adjusted to normalize the distribution of the data. This pleases DBSCAN!

Comparing Algorithmic Complexity



DBSCAN can handle globular clusters and non-globular clusters — why isn't it being used all the time?

DBSCAN is $O(n\log n)$ in the best case scenario, whereas k-means is $O(nk)$ (more or less)

When the number of observations increases, DBSCAN is less efficient than k-means .

DBSCAN Advantages

No need to specify the number of clusters.

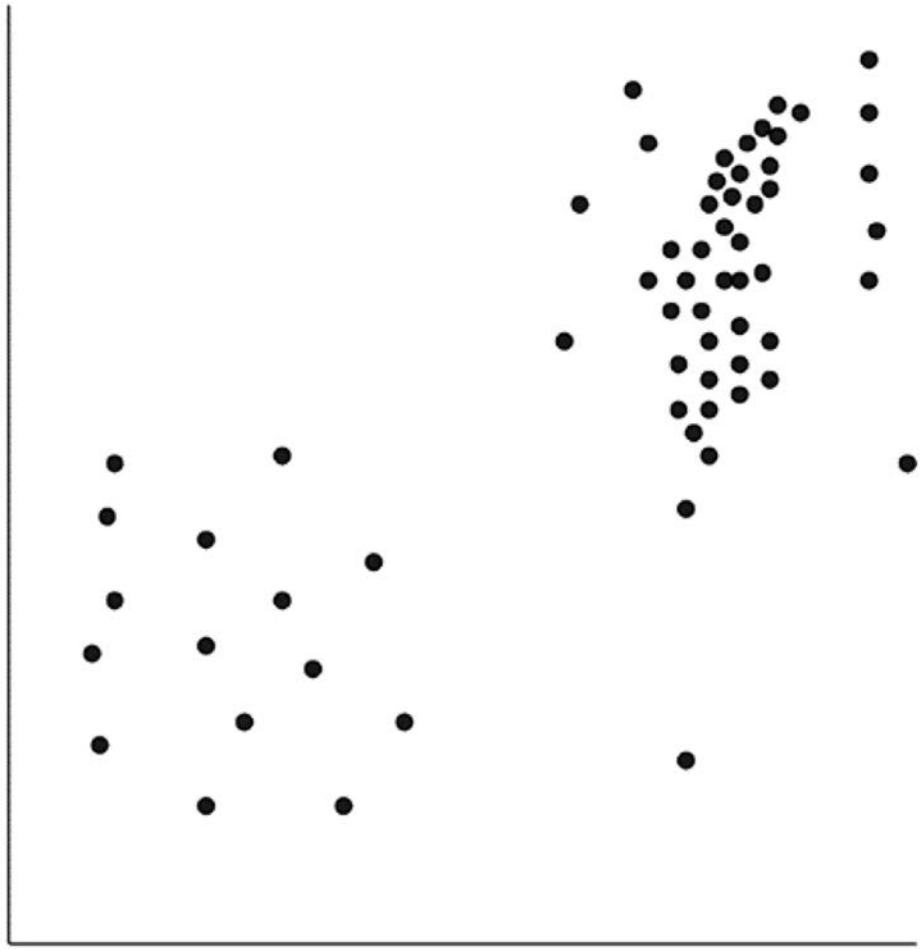
Can find arbitrarily shaped clusters.

Can recognize “noisy” points.

Robust to outliers.

Requires only two parameters (minPts and ε) which can be set by domain experts if the data is well understood.

DBSCAN Limitations



DBSCAN's clustering **kryptonite**: datasets where cluster density is not consistent across clusters.

Hard to set parameters that consistently capture clusters while identifying outliers.

Not entirely deterministic: border points that are reachable from more than one cluster can be part of either cluster, depending on the order.

Parameters Estimation

minPts:

- minPts $\geq$ # features + 1
- larger values are better for noisy data sets
- minPts $\geq 2 \times dim$ for large datasets or sets with duplicates

ε:

- if too small, a large prop. of observations is not clustered
- if too high, majority of observations are in the same cluster
- in general, small values are preferable

Distance function:

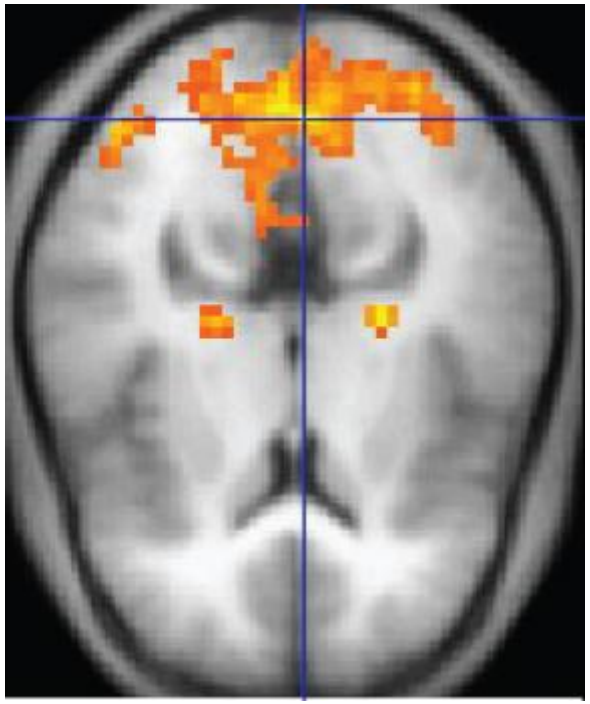
- has a major impact on the results
- should be selected before ε is chosen

DBSCAN Examples
Detecting Alzheimer's Disease

Mild cognitive impairments (MCI) are a known to be a risk for factor for development of Alzheimer's Disease

MCI are accompanied by changes in brain structure

But which changes indicate that people will go on to develop Alzheimer's?



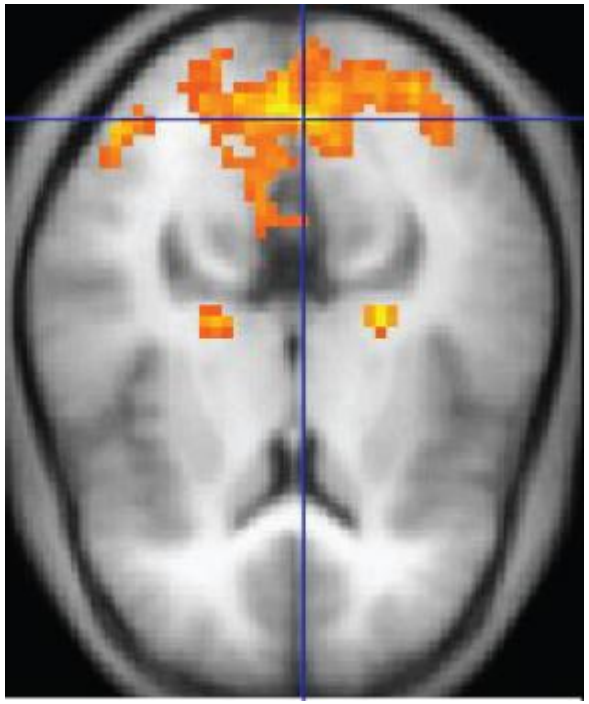
FMRI highlighting some areas of the pre-frontal cortex.

Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease, by C. Plant, S.J. Teipel, A. Oswald, C. Böhm, T. Meindl, J. Mourao-Miranda, A.W. Bokde, H. Hampel, M. Ewers.

DBSCAN Examples
Detecting Alzheimer's Disease

A number of different data science techniques applied to MRI data:

- Support Vector Machines
- Bayesian Statistics
- Voting Feature Intervals
- Feature Extraction
- DBSCAN



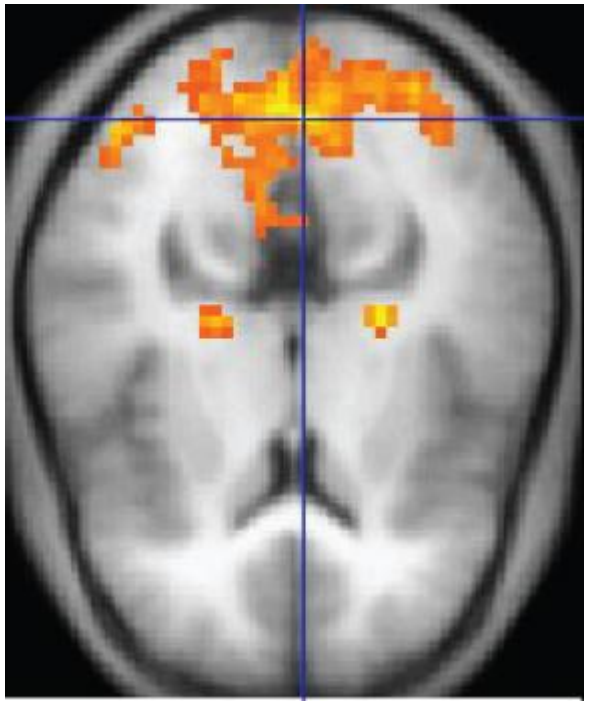
FMRI highlighting some areas of the pre-frontal cortex.

Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease, by C. Plant, S.J. Teipel, A. Oswald, C. Böhm, T. Meindl, J. Mourao-Miranda, A.W. Bokde, H. Hampel, M. Ewers.

DBSCAN Examples
Detecting Alzheimer's Disease

DBSCAN is used once voxels that provide high information about the classification of the image are identified using entropy based measures

DBSCAN then groups pixels with similar spatial and information levels to determine which parts of the brain are the most important for the diagnosis



FMRI highlighting some areas of the pre-frontal cortex.

Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease, by C. Plant, S.J. Teipel, A. Oswald, C. Böhm, T. Meindl, J. Mourao-Miranda, A.W. Bokde, H. Hampel, M. Ewers.

DBSCAN Examples

Some More Examples

A novel approach for predicting the length of hospital stay with DBSCAN and supervised classification algorithms

Simulation of DNA damage clustering after proton irradiation using an adapted DBSCAN algorithm

Where traffic meets DNA: mobility mining using biological sequence analysis revisited

Individual Movements and Geographical Data Mining. Clustering Algorithms for Highlighting Hotspots in Personal Navigation Routes

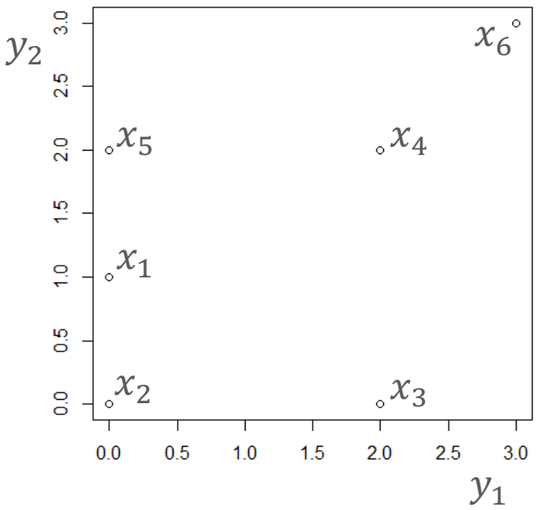
Spectral Clustering

Clustering in General

Spectral Clustering Overview

Dataset

i	y_1	y_2
x_1	0	1
x_2	0	0
x_3	2	0
x_4	2	2
x_5	0	2
x_6	3	3

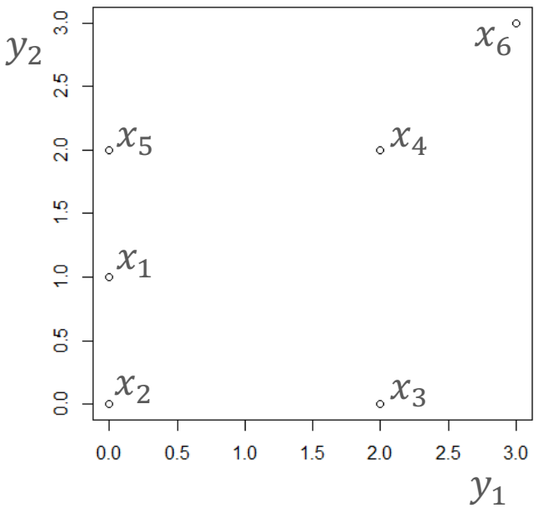


Clustering in General

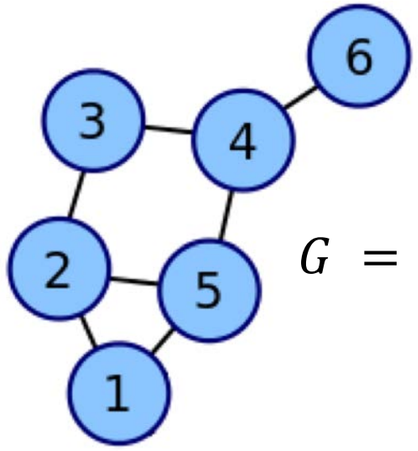
Spectral Clustering Overview

Dataset

i	y_1	y_2
x_1	0	1
x_2	0	0
x_3	2	0
x_4	2	2
x_5	0	2
x_6	3	3



Similarity Graph



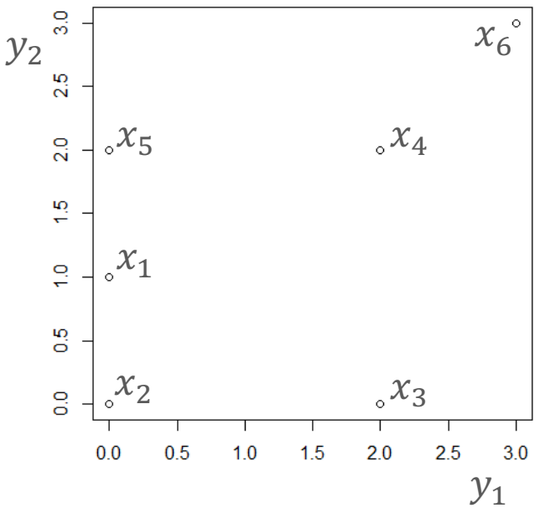
$$G = (V, E, W)$$

Clustering in General

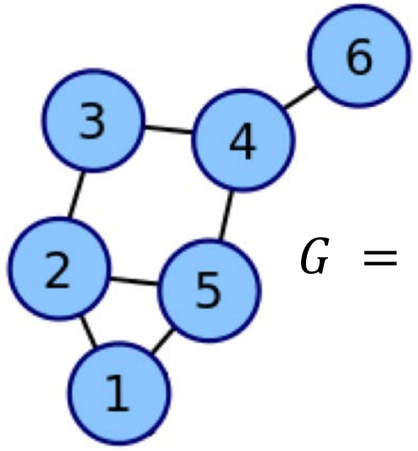
Spectral Clustering Overview

Dataset

i	y_1	y_2
x_1	0	1
x_2	0	0
x_3	2	0
x_4	2	2
x_5	0	2
x_6	3	3



Similarity Graph



$G = (V, E, W)$

Eigenvalue Problem

$L\vec{u} = \lambda\vec{u}$

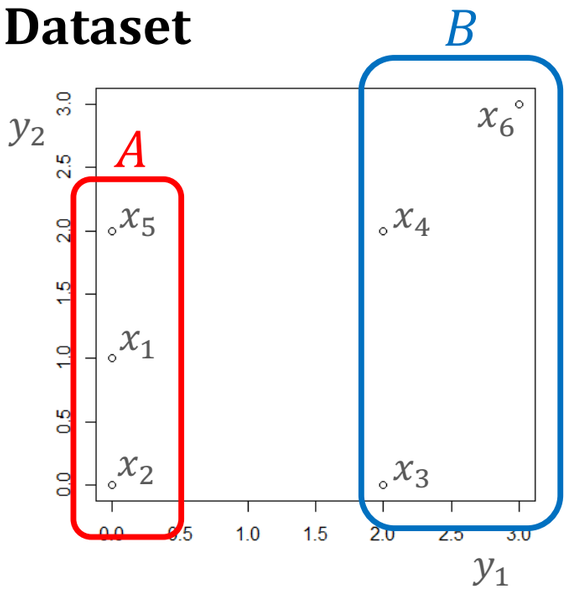
and **???**

Clustering in General

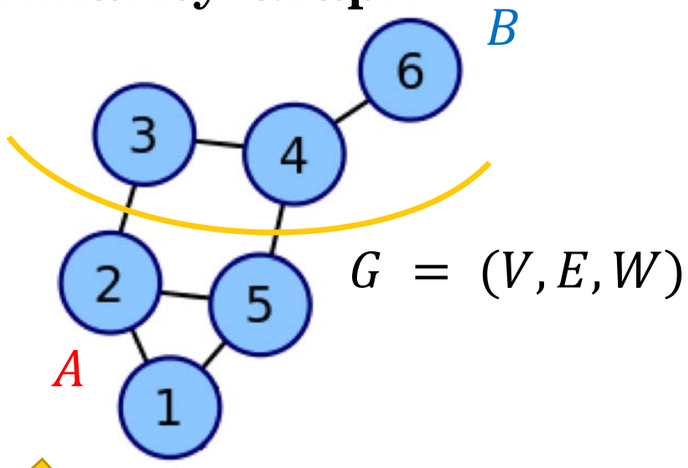
Spectral Clustering Overview

Dataset

i	y_1	y_2
x_1	0	1
x_2	0	0
x_3	2	0
x_4	2	2
x_5	0	2
x_6	3	3



Similarity Graph



$G = (V, E, W)$

Eigenvalue Problem

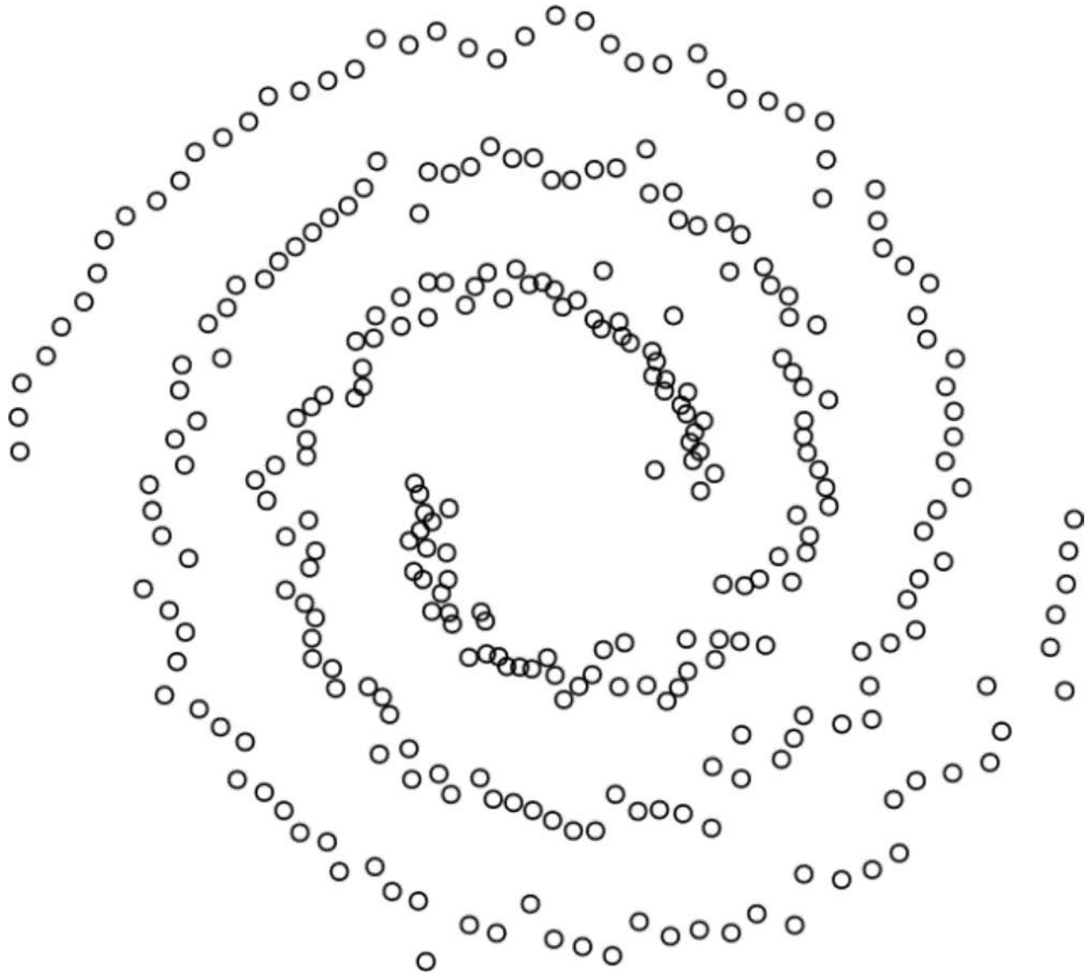
$L\vec{u} = \lambda\vec{u}$

and **???**

??????

Clustering in General

Motivation for Spectral Clustering



Spectral clustering makes no assumption on the shape of a cluster

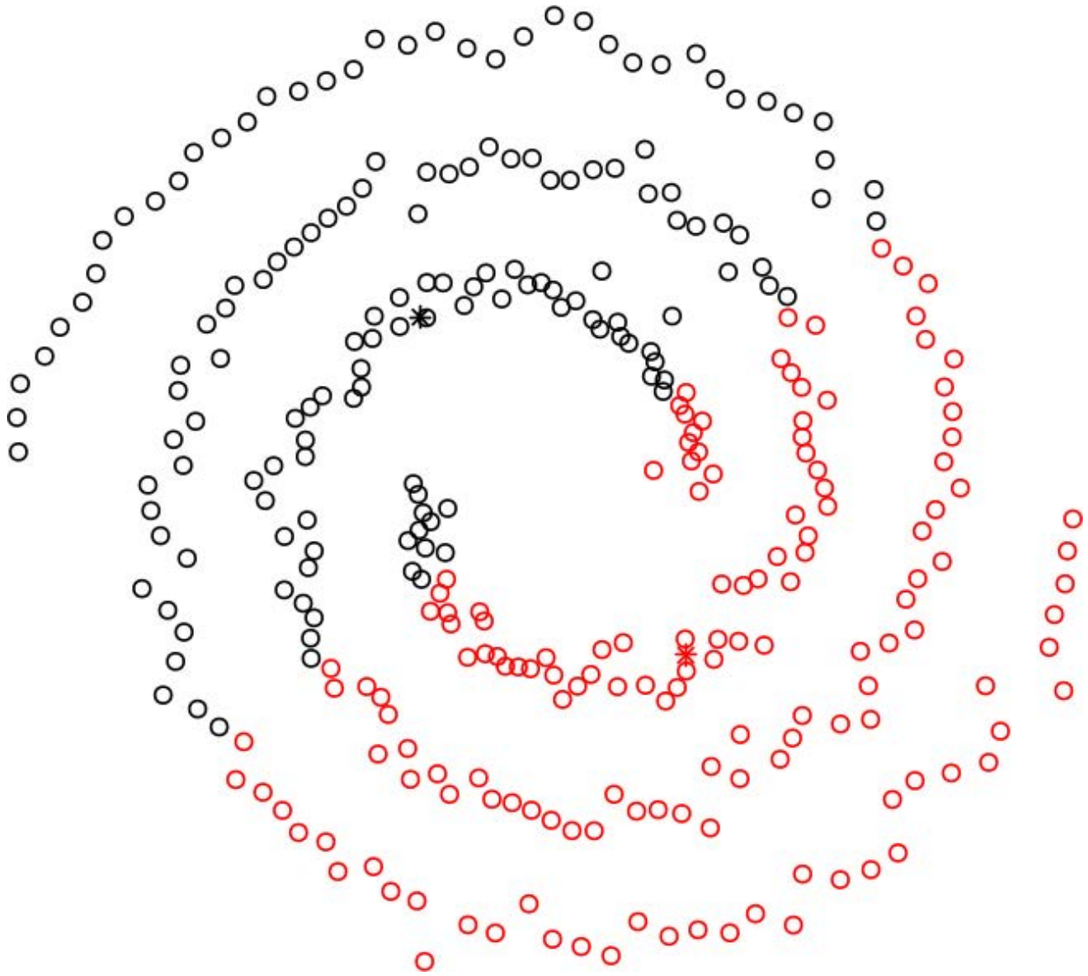
- compactness vs. connectedness

Spectral clustering can be implemented efficiently for large datasets

- computing eigenvalues is numerically "efficient"

Clustering in General

Motivation for Spectral Clustering



Spectral clustering makes no assumption on the shape of a cluster

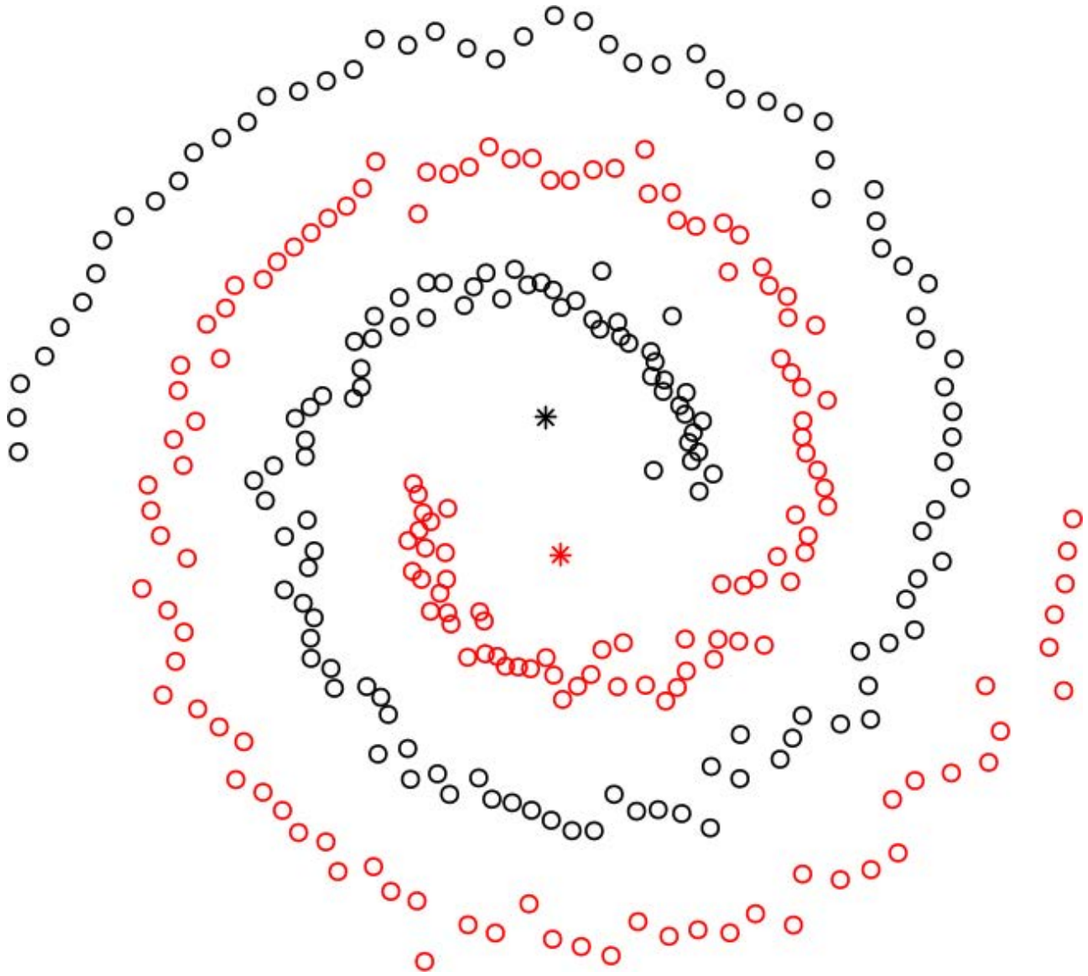
- compactness vs. connectedness

Spectral clustering can be implemented efficiently for large datasets

- computing eigenvalues is numerically "efficient"

Clustering in General

Motivation for Spectral Clustering



Spectral clustering makes no assumption on the shape of a cluster

- compactness vs. connectedness

Spectral clustering can be implemented efficiently for large datasets

- computing eigenvalues is numerically "efficient"

Data Pre-Processing
Similarity Graph

In graph theory, the notation of a **similarity graph** is $G = (V, E, W)$.

1. Data points x are **vertices** $v \in V$.
2. A pair of vertices v_i, v_j are connected by an **edge** $e_{ij} = 1$ if the **similarity weight** $w_{ij} > \tau$ for a given threshold $\tau \in [0,1)$.
3. The edges e_{ij} form the **adjacency matrix** E.
4. The weights w_{ij} form the **similarity matrix** W.
5. The (diagonal) **degree matrix** D provides information about the number of edges attached to a vertex: $d_{ii} = \sum_{j=1}^{n} e_{ij}$.

External requirements: threshold τ, similarity measure w.

Data Pre-Processing

Similarity Graph – Example

With the **Gower** similarity measure on data with m features

$$w_G(v_i, v_j) = 1 - \frac{1}{m}\sum_{k=1}^{m} \frac{|x_{i,k} - x_{j,k}|}{\text{range of } k^{\text{th}} \text{ feature}}$$

the similarity matrix of the previous data is

$$W = \begin{pmatrix} 0 & 5/6 & 1/2 & 1/2 & 5/6 & 1/6 \\ 5/6 & 0 & 2/3 & 1/3 & 2/3 & 0 \\ 1/2 & 2/3 & 0 & \boxed{2/3} & 1/3 & 1/3 \\ 1/2 & 1/3 & \boxed{2/3} & 0 & 2/3 & 2/3 \\ 5/6 & 2/3 & 1/3 & 2/3 & 0 & 1/3 \\ 1/6 & 0 & 1/3 & 2/3 & 1/3 & 0 \end{pmatrix}$$

For instance, $w_G(v_3, v_4) = w_{34} = w_{43} = 1 - \frac{1}{2}\left\{\frac{|x_{3,1} - x_{4,1}|}{r_1} + \frac{|x_{3,2} - x_{4,2}|}{r_2}\right\}$.

But $r_1 = r_2 = 3$, so $w_{34} = w_{43} = 1 - \frac{1}{6}\{|2-2| + |0-2|\} = \frac{2}{3}$.

Data Pre-Processing

Similarity Graph – Example

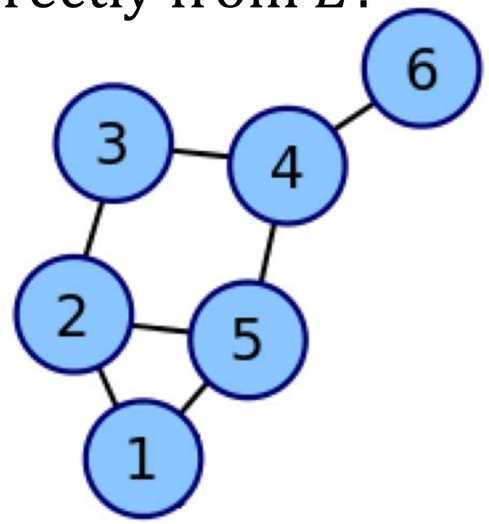
- Let's use a threshold value $\tau = 0.6$. The adjacency matrix is thus

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- Incidentally, the degree matrix is

$$D = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- The similarity graph G is read directly from E:



- Now all that is left is to **partition** the graph!

Graph Partitions

Graph Cuts

- A **graph cut** partitions a graph into two sub-graphs (**clusters**) A, B.
- The goal is to partition the graph so that edges within a group have large weights (so the vertices they join are **similar**) and edges across groups have small weights (so the vertices they join are **dissimilar**).
- We focus on one way to do this: the **Normalized Cut**.

Other partition schemes: Min Cut, Ratio Cut, Min Max Cut

- An objective function $J(A, B)$ must be minimized against the set of all possible partitions (A, B).
- The partition which minimizes J gives rise to the first clustering level.
- The procedure can be repeated as necessary on the cluster sub-graphs.

Graph Partitions

Normalized Cut – Example

Objective function:

$$J_{\mathrm{NCut}} = \mathrm{Cut}(A,B)\left(\frac{1}{\mathrm{Vol}(A)} + \frac{1}{\mathrm{Vol}(B)}\right)$$

Sum of all the weights on edges emanating from B

Sum of all the weights on edges emanating from A

Sum of all the weights on edges starting in one group and ending in the other

Advantages:

- Takes into consideration the size of partitioned groups
- Tends to avoid isolating vertices
- Takes into consideration intra-group variance

Limitations

- Not an easy optimization problem to solve (**NP-hard!!**)

Graph Partitions

Normalized Cut – Example

Objective function:

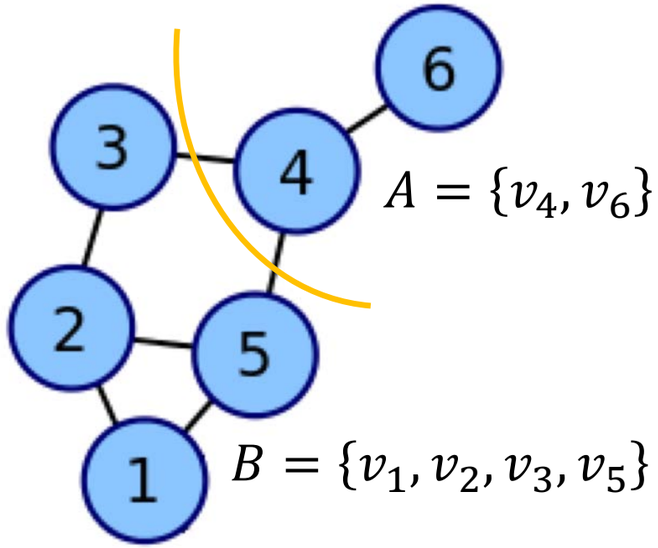
$$J_{\mathrm{NCut}} = \mathrm{Cut}(A,B)\left(\frac{1}{\mathrm{Vol}(A)} + \frac{1}{\mathrm{Vol}(B)}\right)$$

Advantages:

- Takes into consideration the size of partitioned groups
- Tends to avoid isolating vertices
- Takes into consideration intra-group variance

Limitations

- Not an easy optimization problem to solve (**NP-hard!!**)



$\mathrm{Cut}(A,B) = \sum_{i\in A, j\in B} w_{ij} = 3$

$\mathrm{Vol}(A) = \sum_{i\in A, j\in V} w_{ij} = 13/3$

$\mathrm{Vol}(B) = \sum_{i\in V, j\in B} w_{ij} = 32/3$

$$J_{\mathrm{NCut}}(A,B) = 0.97$$

The Eigenvalue Problem

How Spectral Clustering Got Its Name

Spectral clustering is a **compromise**: it solves an *easier* problem than Normalized Cut optimization, but with *similar* solutions.

The **Laplacian matrix** is a spectral representation of a graph.

- Simple Laplacian: $L = D - E$ ← Careful! There are competing definitions.
- Symmetric Laplacian: $L_S = D^{-1/2} L D^{-1/2}$
- Asymmetric Laplacian (random walk): $L_A = D^{-1} L$

In the case of two clusters, J_{NCut} is minimized when finding the eigenvector f for the **second smallest** eigenvalue of L_S, leading to the name of the method (special case of general algorithm, see later). ← L is positive semi-definite and its smallest eigenvalue is 0

The clustering is recovered by sending $v_i \in A$ when $f_i > 0$, and $v_i \in B$ otherwise (or *vice-versa*). ← Deterministic?

Interlude

Eigenvalues and Laplacian Matrices

An **eigenvalue** λ of a matrix T is a complex number (potentially with no imaginary part) such that $\dim \ker(T - \lambda I) > 0$.

In other words, λ is an eigenvalue of T if there exists (at least) an **eigenvector** $\vec{v} \neq 0$ such $T\vec{v} = \lambda\vec{v}$.

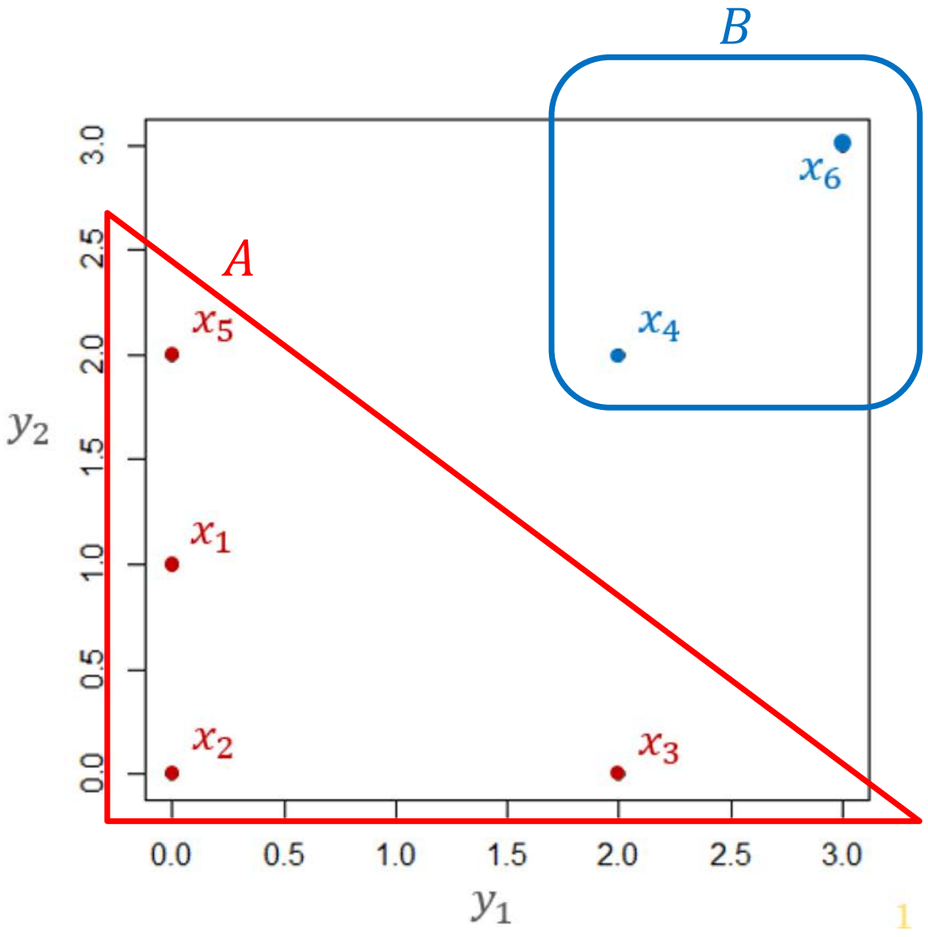
The Laplacian matrix L of a graph is a matrix representation of that graph.

The Laplacian matrix has a bevy of nice properties that ensure that its eigenvalues behave "as they should"; for instance, the dimension of the eigenspace associated with the eigenvalue $\lambda = 0$ measures the number of connected components in the graph. A first guess for # of clusters?

The Eigenvalue Problem

Simple Laplacian – Example

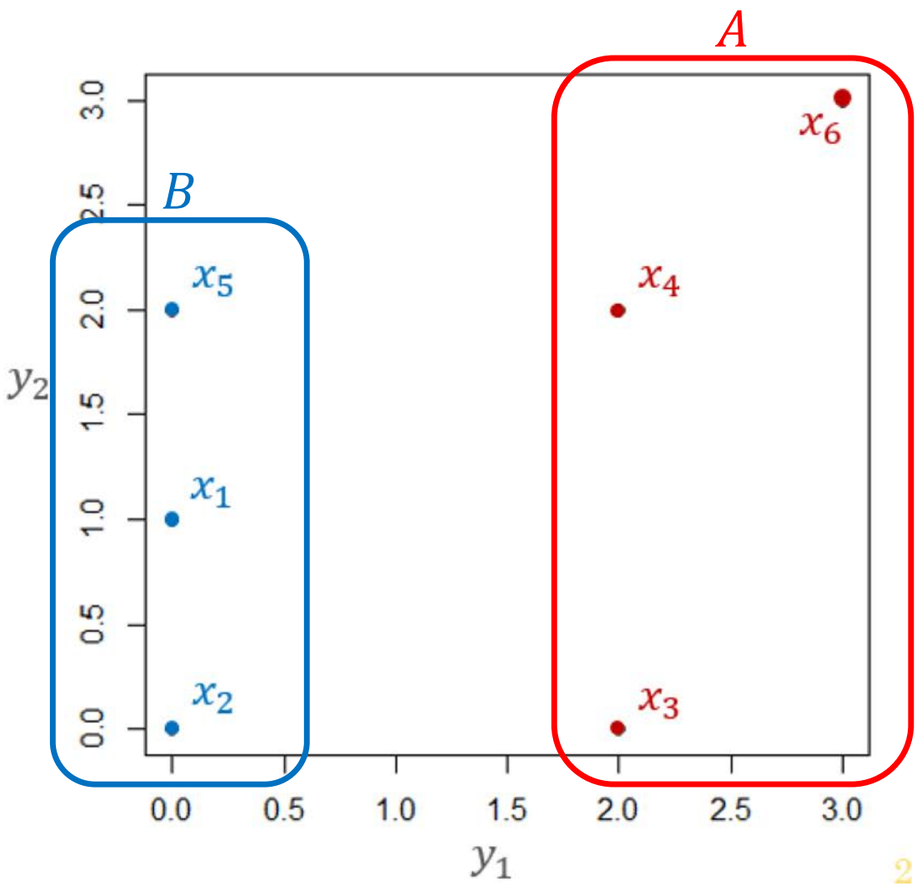
i	f_i	A/B
x_1	-0.39	A
x_2	-0.30	A
x_3	-0.18	A
x_4	0.03	B
x_5	-0.16	A
x_6	0.84	B



The Eigenvalue Problem

Symmetric Laplacian – Example

i	f_i	A/B
x_1	0.36	B
x_2	0.54	B
x_3	-0.03	A
x_4	-0.43	A
x_5	0.14	B
x_6	-0.61	A



The Eigenvalue Problem

Spectral Clustering –Algorithm

(version from von Luxburg's tutorial, with different D and L)

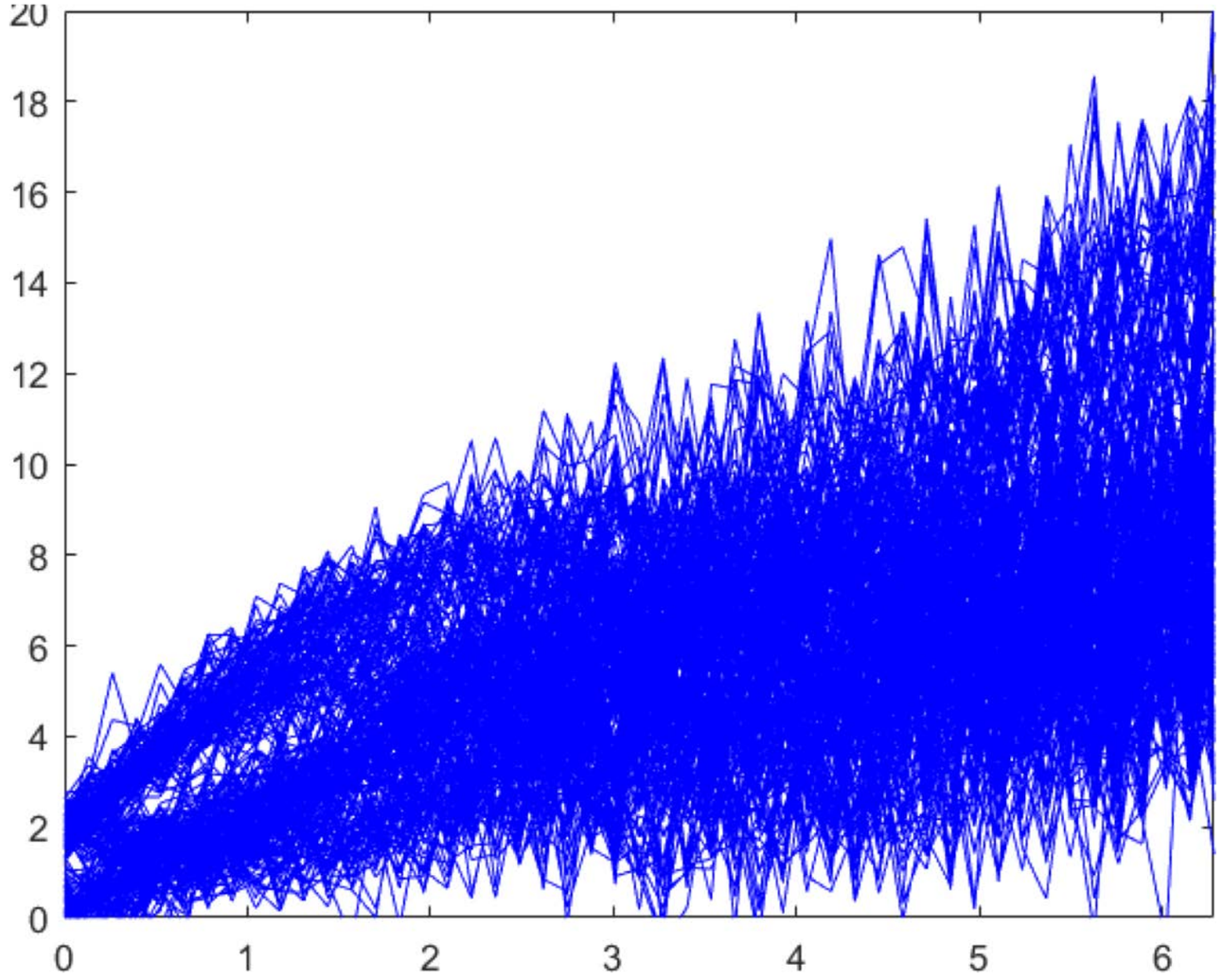
Algorithm to cluster $\{x_1, \dots, x_n\}$ into k clusters: ← Choice of # of clusters

1. Form similarity matrix W. ← Choice of similarity measure
2. Define the degree matrix D. ← Choice of adjacency threshold
3. Construct the Laplacian matrix L. ← Choice of Laplacian
4. Compute the first k orthogonal eigenvectors $\{\mu_1, \dots \mu_k\}$ of the Laplacian L corresponding to its k **smallest** eigenvalues.
5. Construct U, using $\mu_1, \dots, \mu_k$ as **columns**.
6. Normalize the **rows** of U so that they each have unit length; call the new matrix Y.
7. Cluster the rows of Y into k clusters. ← Choice of clustering method
8. Assign the original point x_i to cluster j if the ith row of Y was assigned to cluster j.

Other algorithms: un-normalized spectral clustering, Shi and Malik's algorithm (see von Luxburg's tutorial).

Examples and Case Studies

Latent Classes – Time Series



- 250 times series
- average absolute gap between series used as distance d
- Gaussian similarity measure

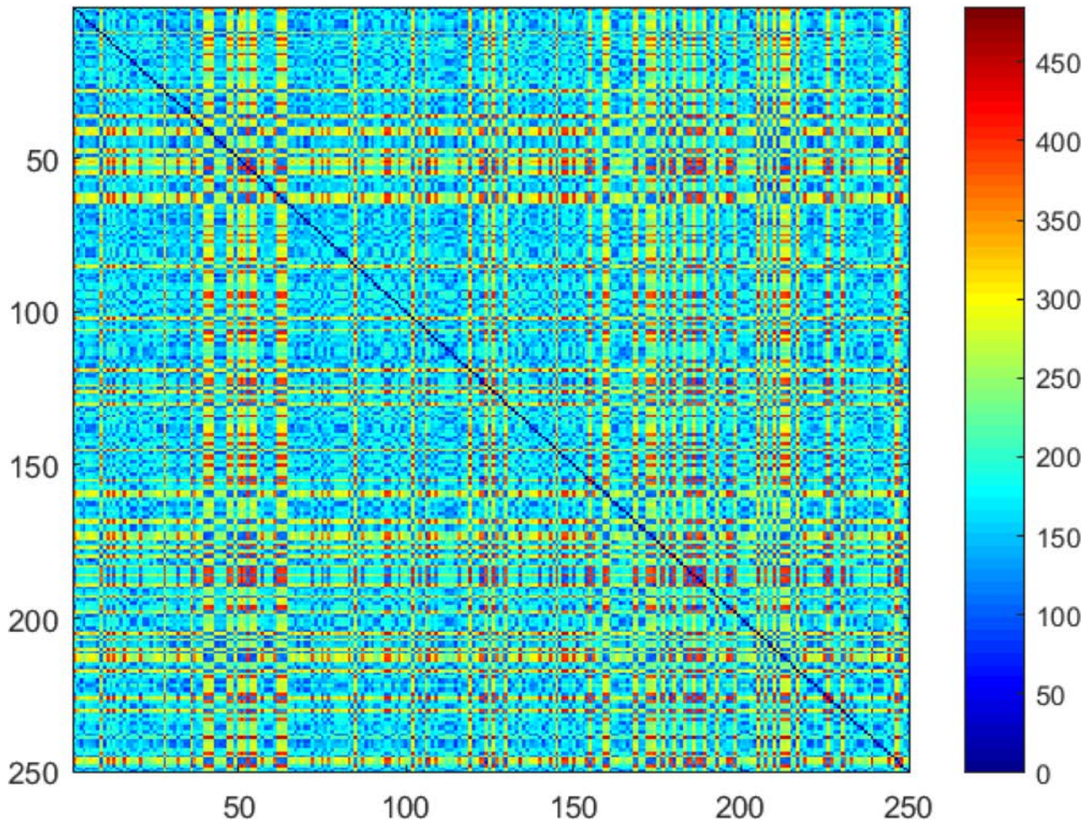
$$w = \exp\left(-\frac{d^2}{2\sigma^2}\right)$$

- $\sigma = 300$
- adjacency threshold $\tau = 0.9$
- $k = 5$ clusters

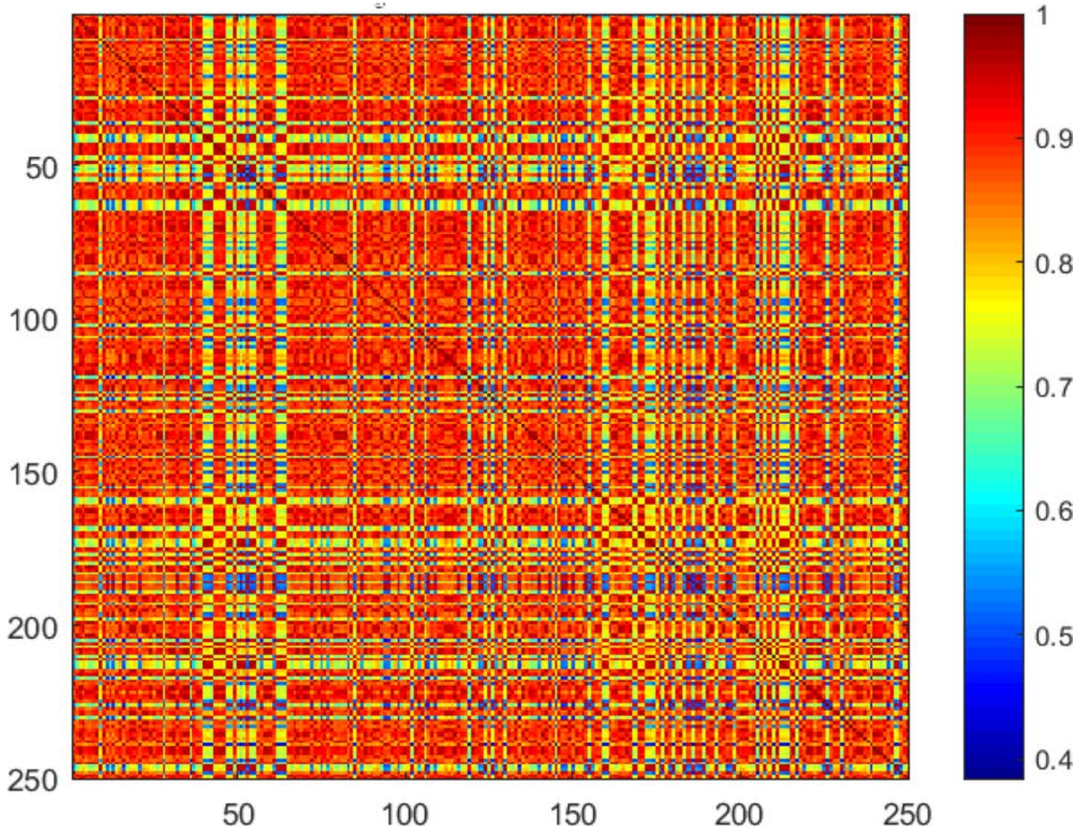
Examples and Case Studies

Latent Classes – Time Series

distance matrix



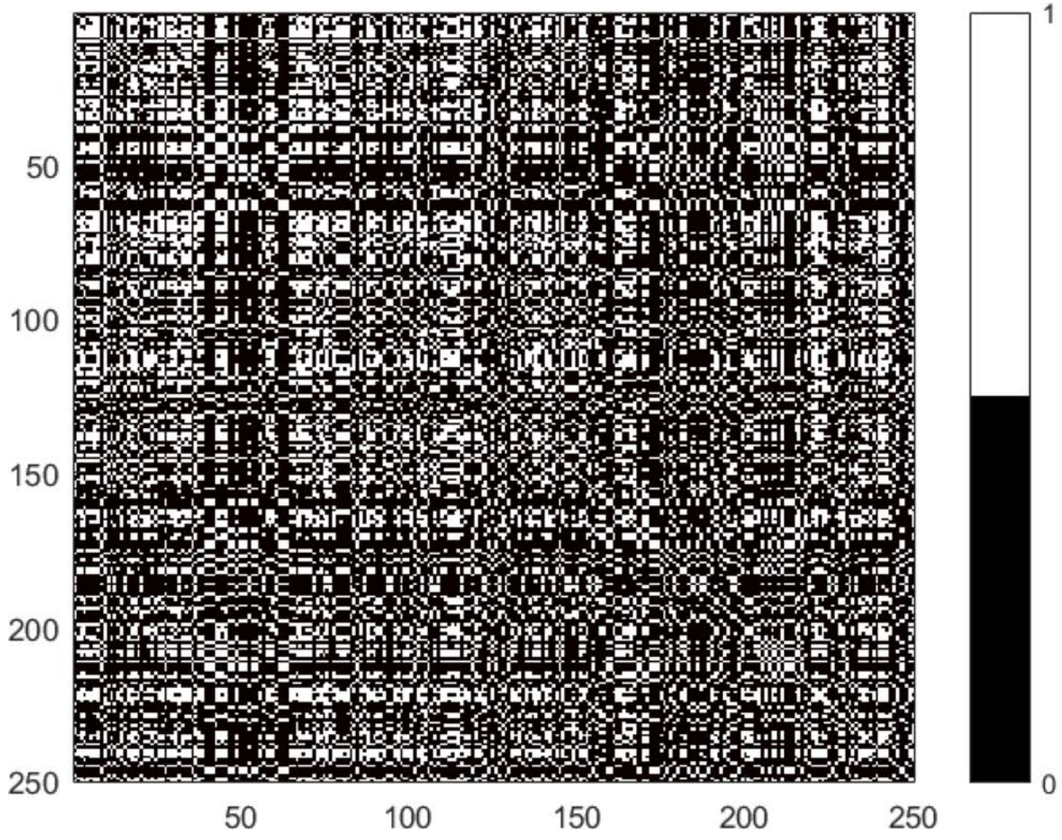
similarity matrix W



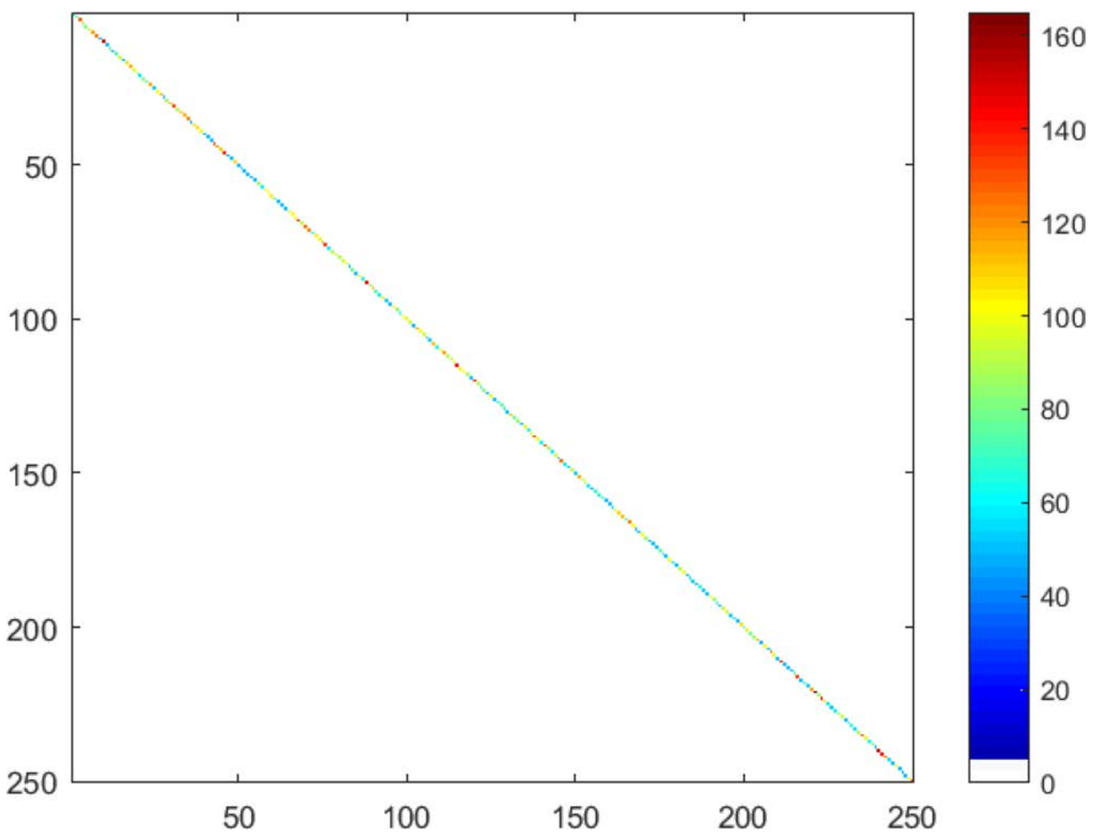
Examples and Case Studies

Latent Classes – Time Series

adjacency matrix E

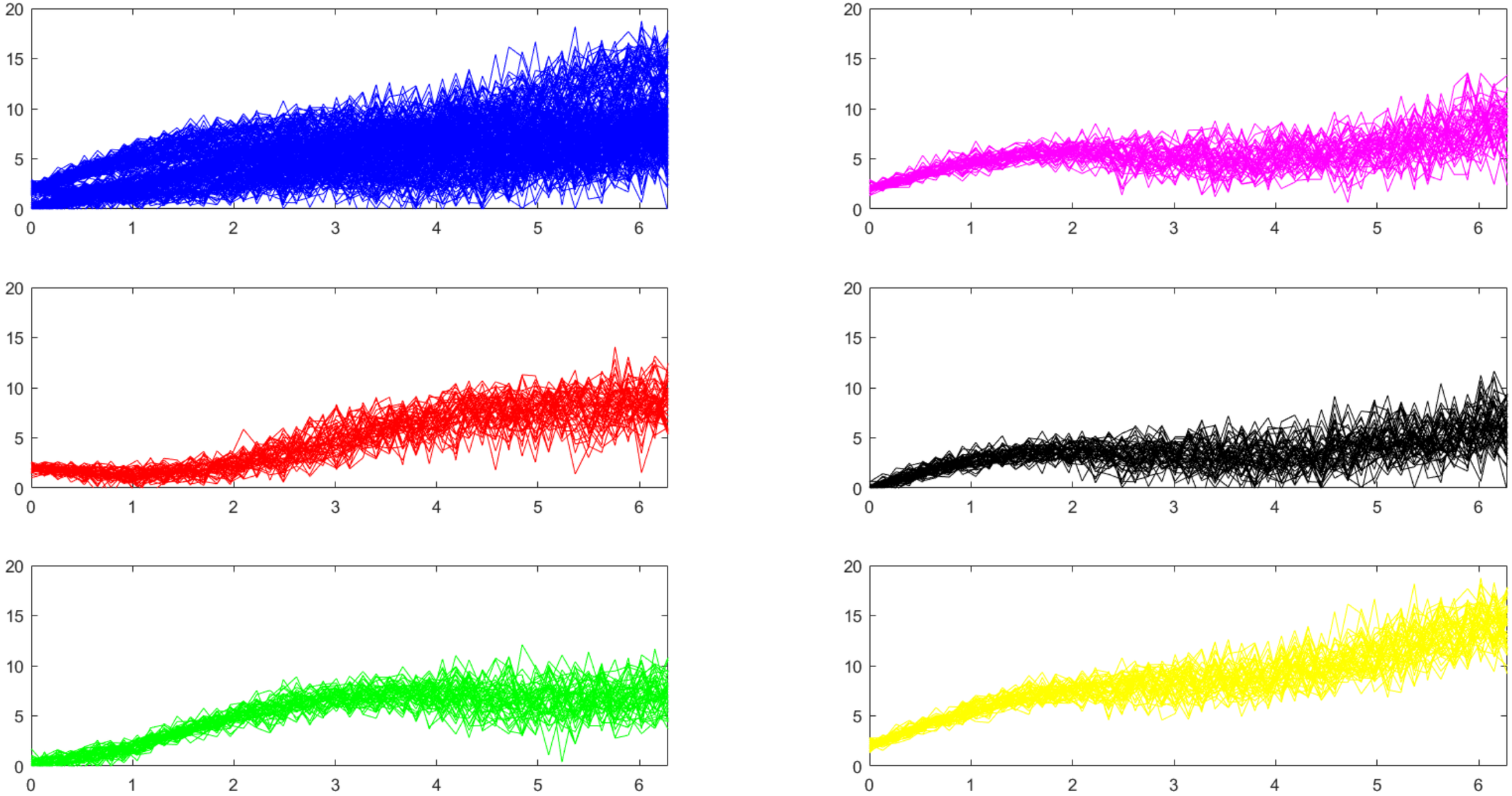


degree matrix D



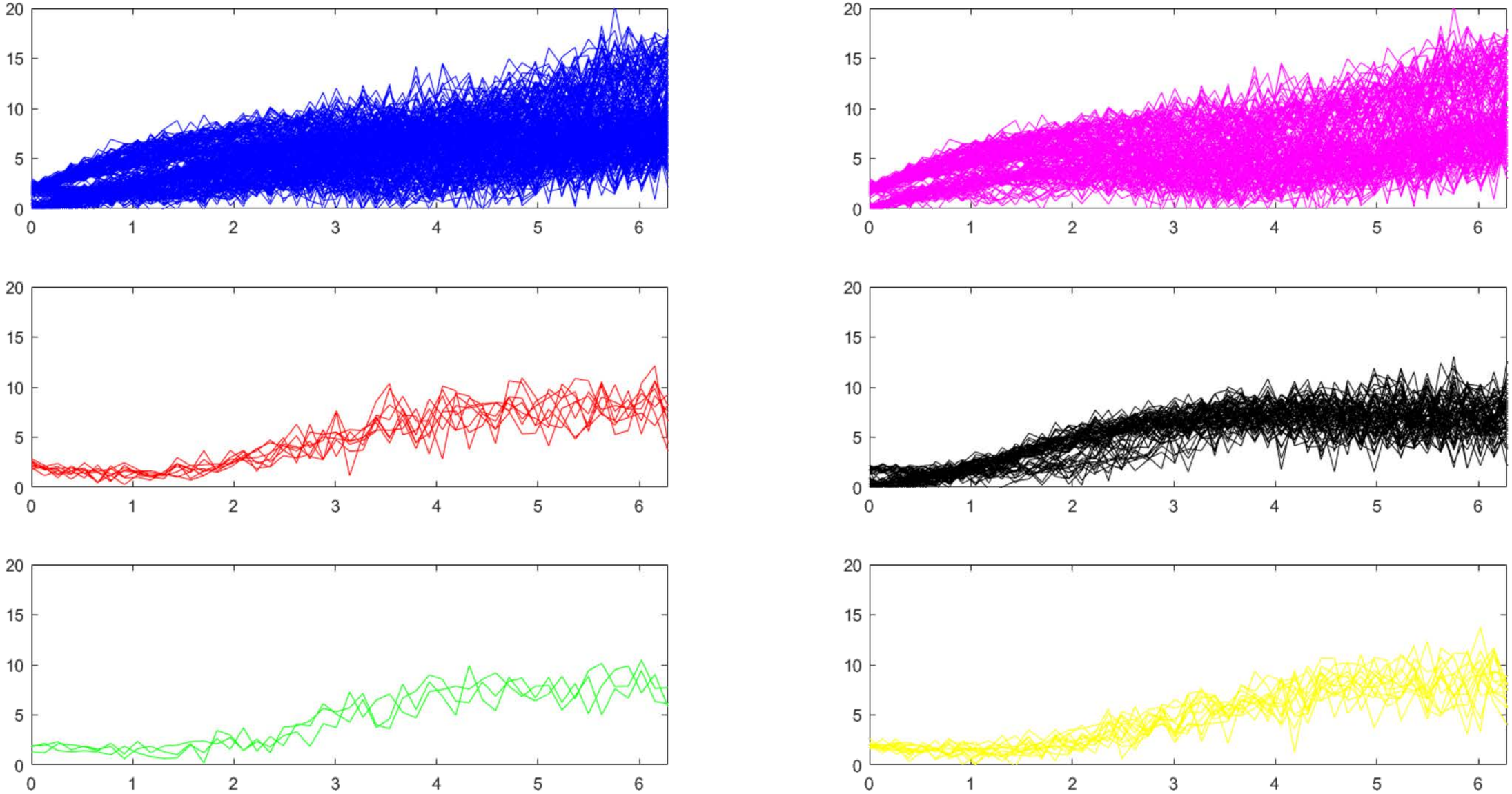
Examples and Case Studies

Latent Classes – Time Series



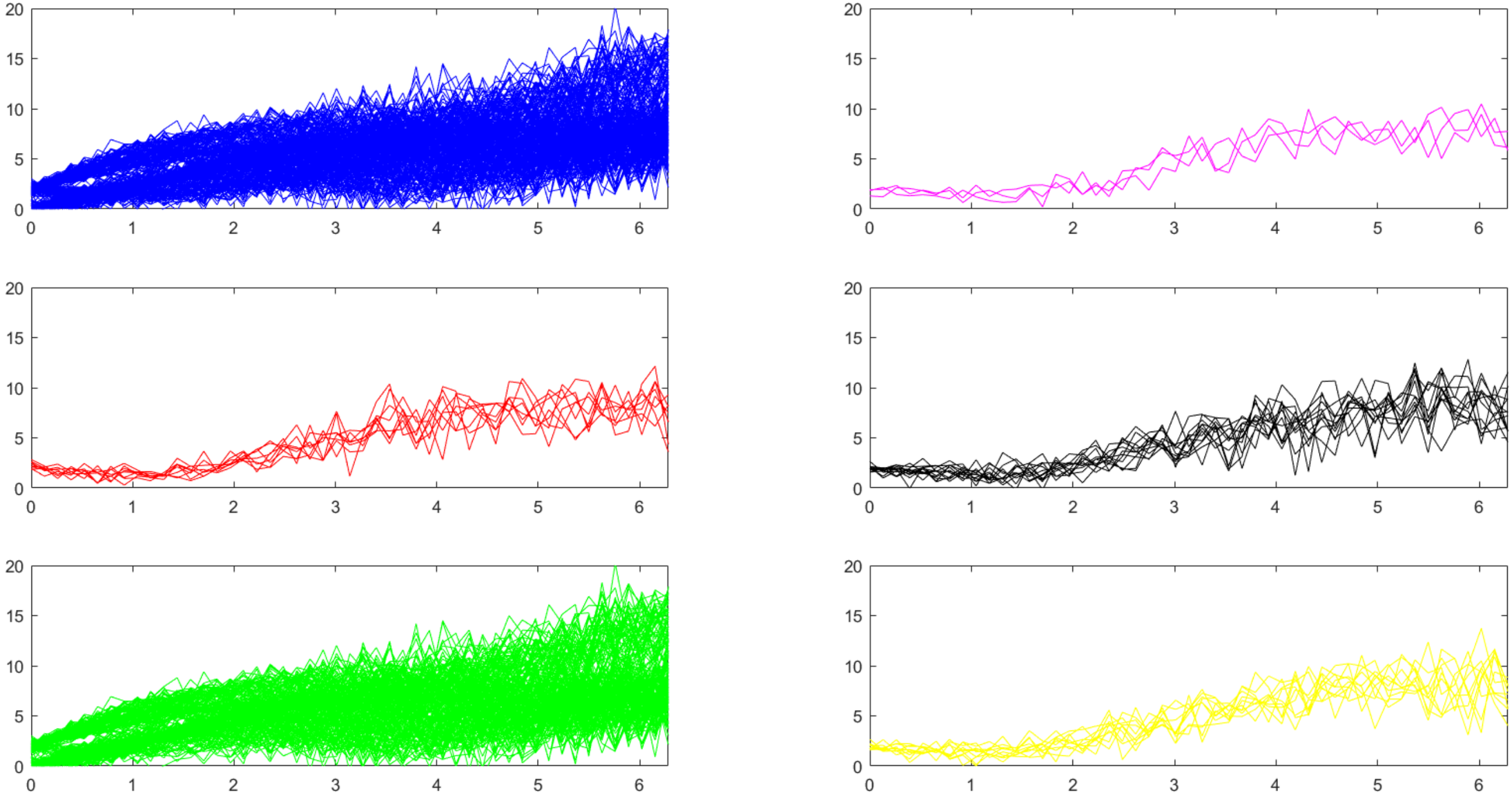
Examples and Case Studies

Latent Classes – Time Series



Examples and Case Studies

Latent Classes – Time Series



Examples and Case Studies

Signal Processing – Spectral Clustering for Speech Separation

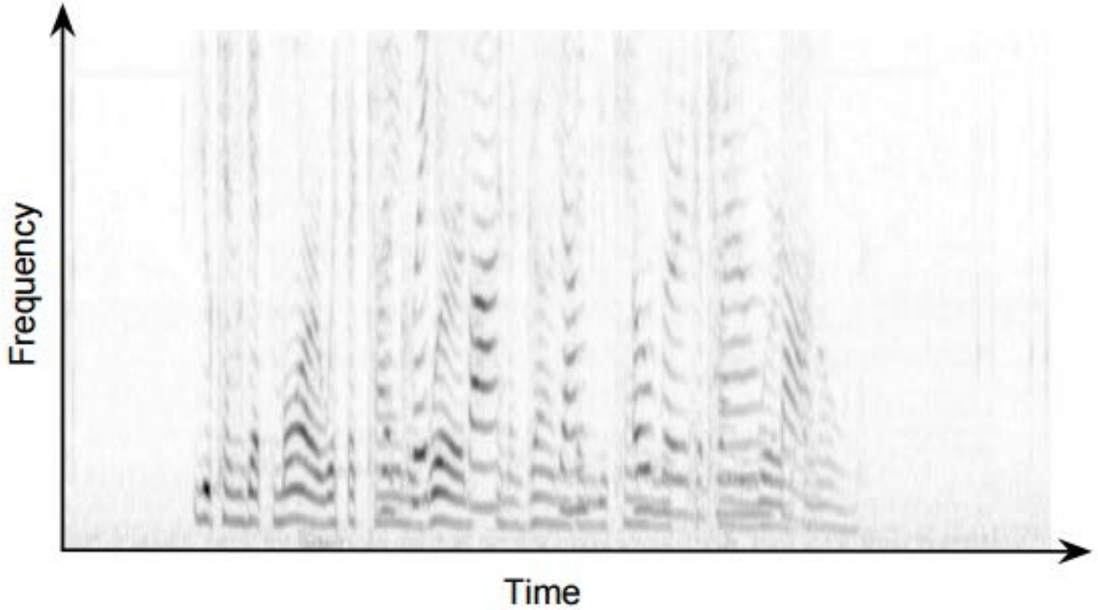
Project: Francis R. Bach and Michael I. Jordan combined prior relevant knowledge with learning similarity algorithm, to explain spectral clustering.

Goal: apply the algorithm to separate two speakers from a one-microphone blind source.

Data: Two speakers give speech and their voice signal is collected by a one-microphone blind source.

Spectrogram of speech (two simultaneous English speakers).

The gray intensity is proportional to the amplitude of the spectrogram.



Examples and Case Studies

Signal Processing – Spectral Clustering for Speech Separation

Method:

- Assume partitions are known in the given sample data.
- Perform spectral clustering on the similarity matrices
- Obtain the same partitions as assumed previously

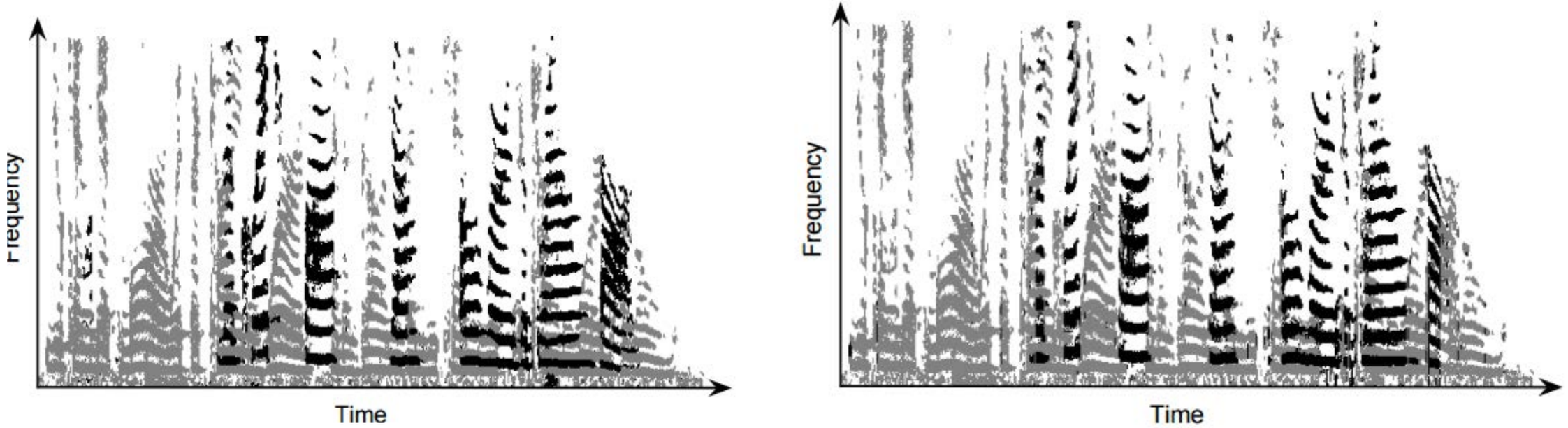
Algorithm: Similar to NJW.

Challenges:

- Limited to the setting of ideal acoustics and equal-strength mixing of two speakers
- Training examples can be created by mixing previously captured signals
- Spectral clustering needs to be robust to irrelevant features
- Computation challenge of spectral clustering applied to speech separation

Examples and Case Studies
Signal Processing – Spectral Clustering for Speech Separation

The result is an optimized segmenter for spectrograms of speech mixtures.



Selected result: (Left) Optimal segmentation for the spectrogram of English speakers, where the two speakers are “black” and “grey”; this segmentation is obtained from the known separated signals. (Right) The blind segmentation obtained with our algorithm.

F.R. Bach, M.I. Jordan, *Learning Spectral Clustering, With Application to Speech Separation*, Journal of Mach. Learn. Res. 7

Examples and Case Studies

Sensor Detection – A Spectral Clustering Approach to Validating Sensors via Their Peers in Distributed Sensor Networks

Project: H. T. Kung and Dario Vlah describe a spectral clustering approach to identify bad sensors, by using a simple model problem.

Motivation: current status and environment affect sensors performance and impractical to bring calibrate device to test each sensor

Goal: using peer sensors to detect badly performing sensors

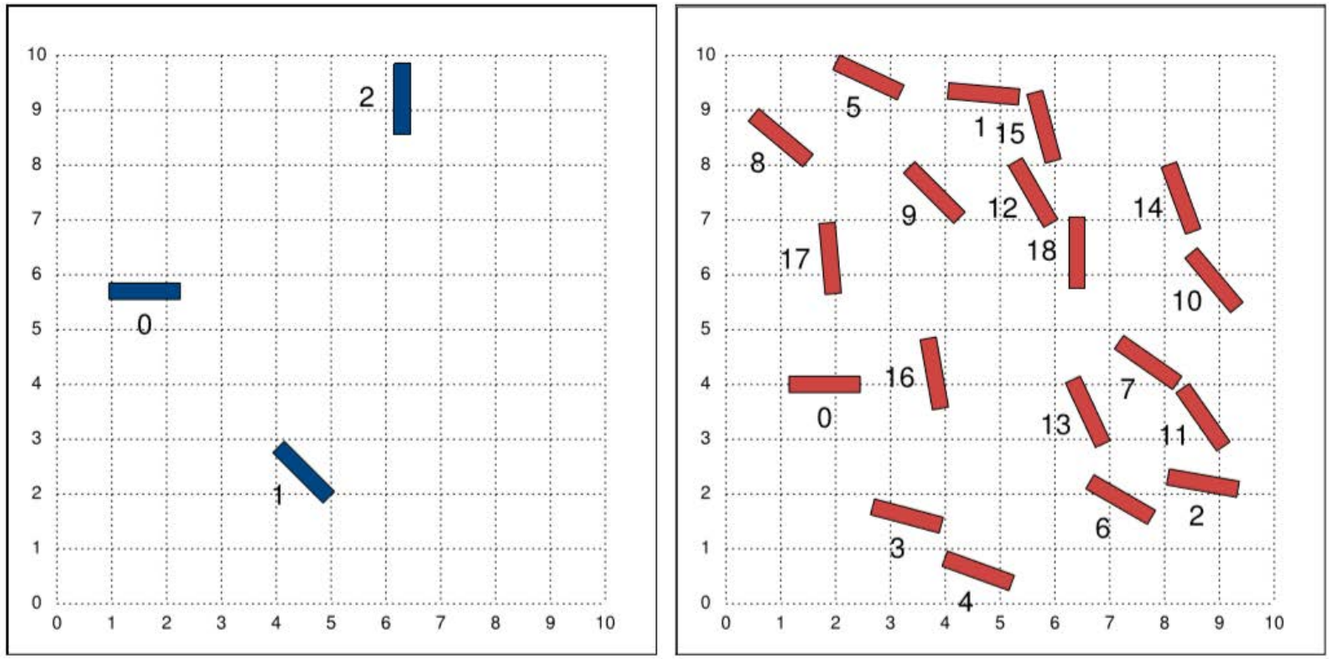
Method: simulation and spectral clustering

Examples and Case Studies

Sensor Detection – A Spectral Clustering Approach to Validating Sensors via Their Peers in Distributed Sensor Networks

Model design: sensors are indexed by their antenna orientations

- assume that the matching of a sensor and a target is based on the degree to which their antenna orientations match
- use of non-principal eigenvector with the principal one, to detect clustering structures



(a) 3 targets with 3 antenna orientations

(b) 19 sensors with 19 antenna orientations

Sensors and targets in the same region

Examples and Case Studies

Sensor Detection – A Spectral Clustering Approach to Validating Sensors via Their Peers in Distributed Sensor Networks

Simulation of large systems on the same model design

- Data: 100 sensors and 10 targets
- Assumption 1: sensors and targets are evenly partitioned into three groups, with antenna orientations of 0, 45 and 90 degrees
- Assumption 2: some randomly selected sensors are bad sensors in the sense that their measurements can be off by any amount from -100% to +100%

Results:

- When the number k of leading eigenvectors used increases, the accuracy performance improves
- The number of false positives decreases with the number of bad sensors input to the simulator.
- Spectral clustering achieves almost perfect performance in specific circumstances.

Clustering Validation
Is a Clustering Scheme Any Good?

There is **NO** optimal validation approach.

Possibilities include:

- comparing with the optimal clustering (external)
- comparing with other clustering methods (external)
- visualizing the clusters (external)
- Davies-Bouldin, Within-SS (internal) ← # of clusters
- repeated clusterings (internal) ← co-clustered items

Scenario 1: given data D, true clustering C, algorithm A produces C':

- **is C' "close" to C?**

Scenario 2: given data D, true clustering C, algorithm A produces C'; algorithm A^* produces C^*, and so forth.

- **are C', C^*, … , "close" to C? Which one is "closer"?**

Clustering Validation

Is a Clustering Scheme Any Good?

A distance measurement $d(C, C')$ between clusterings is needed…

Let $C = \{C_1, \dots, C_k\}$ be a **clustering** of a set of n data points $\{x_1, \dots, x_n\}$.

The **quadratic cost** is the function defined by

$$\Lambda(C) = -\mathrm{Trace}\big(Z^T(C) \cdot W \cdot Z(C)\big),$$

where Z is the matrix representation of C:

$$z_{ik} = \begin{cases} 1 & \text{if } x_i \in C_k \\ 0 & \text{if } x_i \notin C_k \end{cases}$$

In some sense, the clustering scheme for which $\Lambda(C)$ is minimized is **optimal** against quadratic cost. ⇦ For a given choice of similarity measure

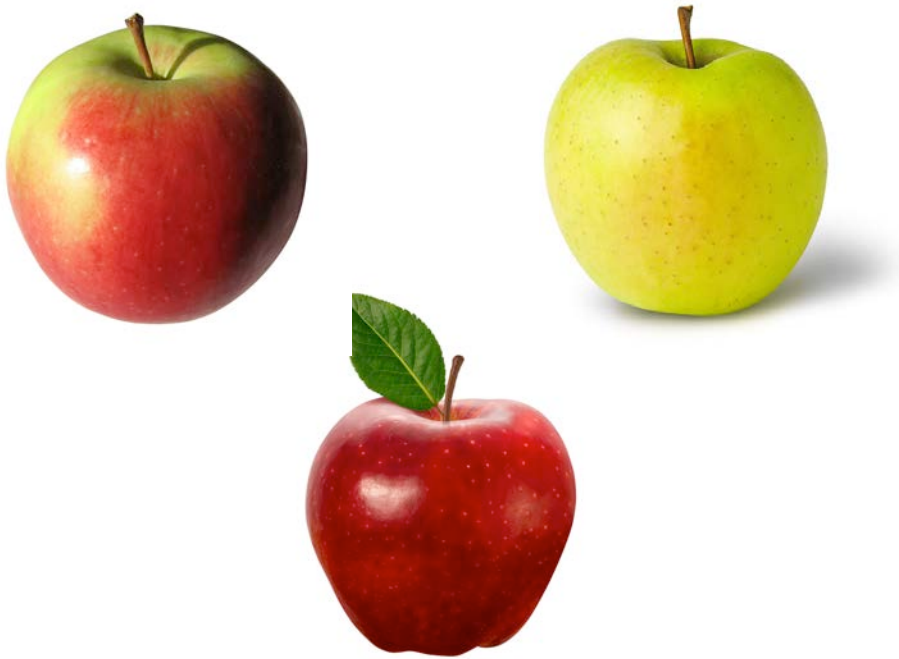
Clustering Validation

Cluster Validation – Part 1

INTRODUCTION

Clustering

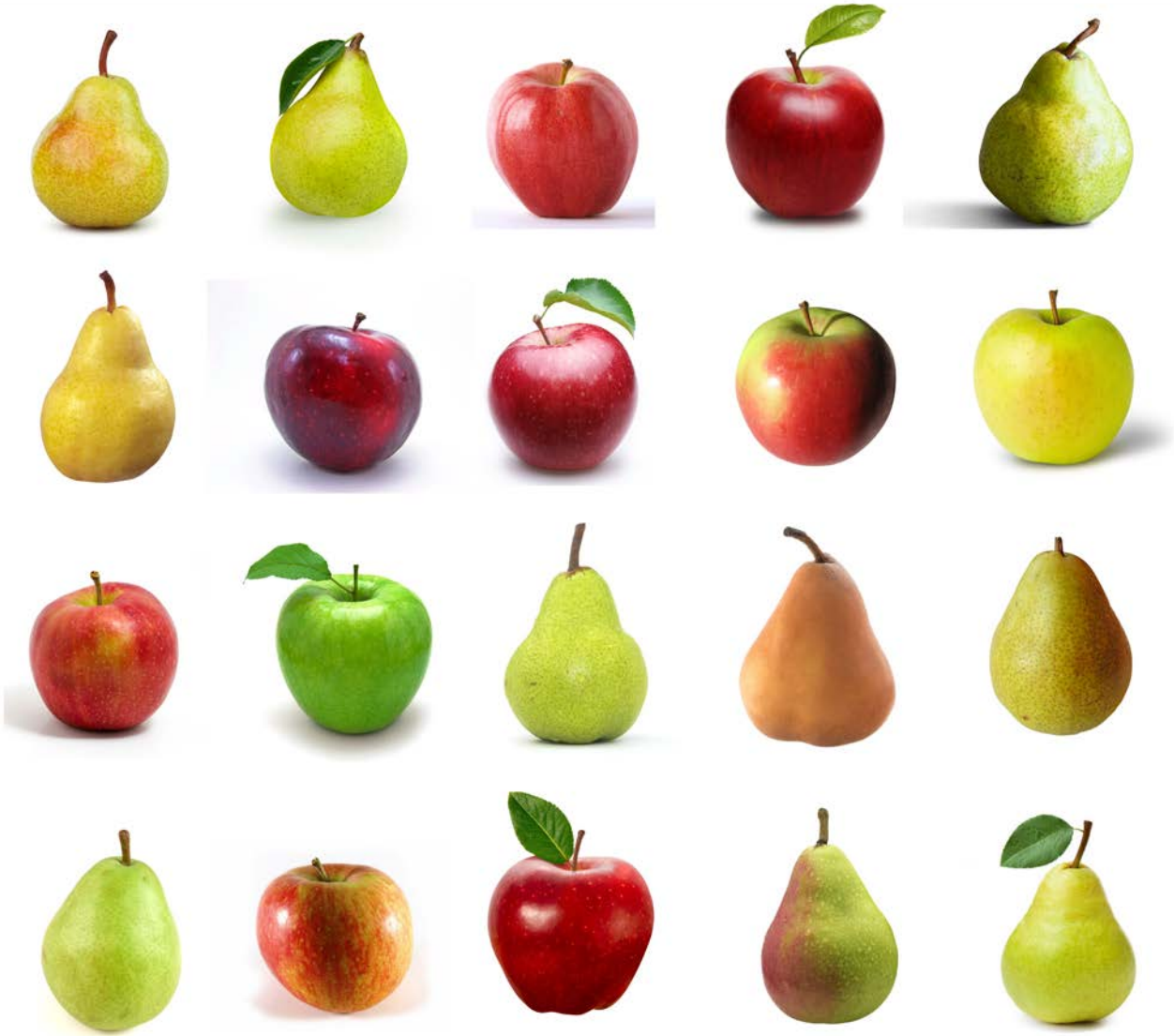
- In machine learning, ***clustering*** is defined as grouping objects <u>based on their **over-all** similarity</u> (or dissimilarity) to each other
- Note that each object has *multiple dimensions*, or attributes available for comparison
- It's tempting to focus on <u>just one or two attributes</u>, but that is typically **not** what we are doing in (machine learning) clustering!
- When we cluster, even if we were to focus on one particular attribute, <u>all of the other attributes would still come along for the ride</u>



What is the same about these objects?
What is different?
Do they belong in the same group?
How many groups? How many classes?

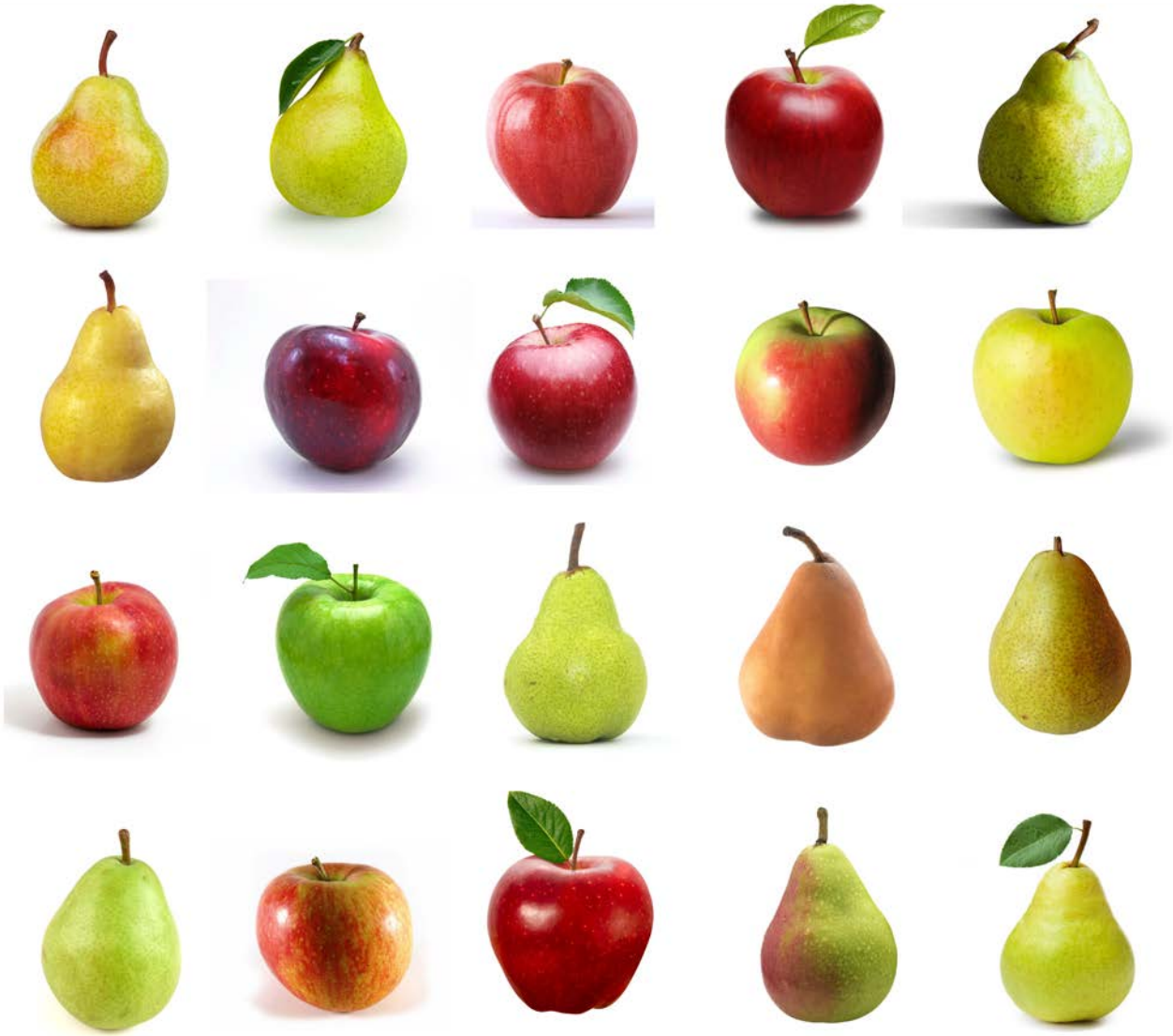
Fruit Image Dataset

- 20 images of fruit
- Are there right or wrong groupings of this dataset?
- Are there multiple possible ‘natural’ clusterings?
- Could different clusterings be used differently?
- Will some clusterings be of (objectively) higher ***quality*** than others?



Making Concepts Concrete

- To appreciate clustering validation, it helps to relate the concepts to something tangible
- In what follows, take the time to think about how the presented concepts can be related to the images from this small dataset

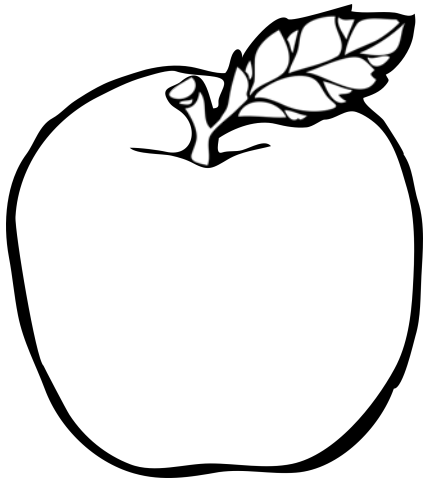


Clustering Validation – Part 2

KEY CONCEPTS ILLUSTRATED

Concept vs. Instance

- We group instances of objects into larger categories (clusters, classes, types)
- These larger categories can be represented by a concept, exemplar, representative or definition
- The concept (exemplar/definition) is a generalized representation - it captures something about all of the instances
- **For a given grouping – *can* we come up with a clear concept that captures the 'essence' of that grouping?**
- If yes, does that make it a good clustering?



Exemplar, Concept, Representative

Definition: "the fleshy, usually rounded red, yellow, or green edible pome fruit of a usually cultivated tree (genus Malus) of the rose family" Mirriam-Webster

Instances

Instance Properties

- For machine learning purposes, we represent properties of object instances using vectors
- Each vector element represents an attribute of the object.
- The value of the vector element represents the value of that property (e.g. the colour) of that object
- Vector Properties:
  - Length
    - (= number of dimensions/attributes)
  - For each dimension
    - Continuous/Discrete
    - Numeric/Categorical
    - Range/Possible Values

[12, 9.12, round, golden delicious]

Does this vector sufficiently describe this object?

Instance-Instance Relationships

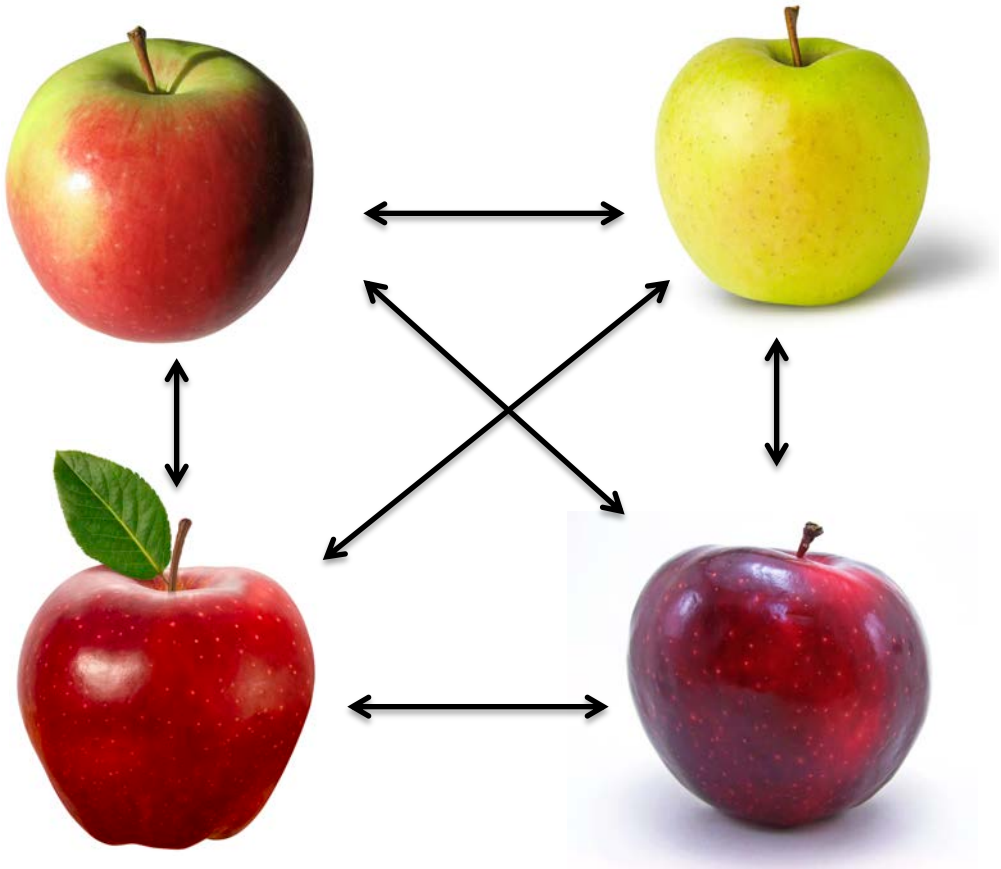
- Defined relationships between instances
- Comparison functions between instances:
  - Take as input vectors or parts of vectors
  - Might only take certain types of input (e.g. numeric)
  - Outputs a comparison result
- Similarity
  - Similarity as defined on a single dimension? Multiple dimensions?
  - Can we come up with functions that give us an overall similarity measure, across <u>all dimensions</u>?

[3, 10.43, round, macintosh]

[12, 9.12, round, golden delicious]

Distance

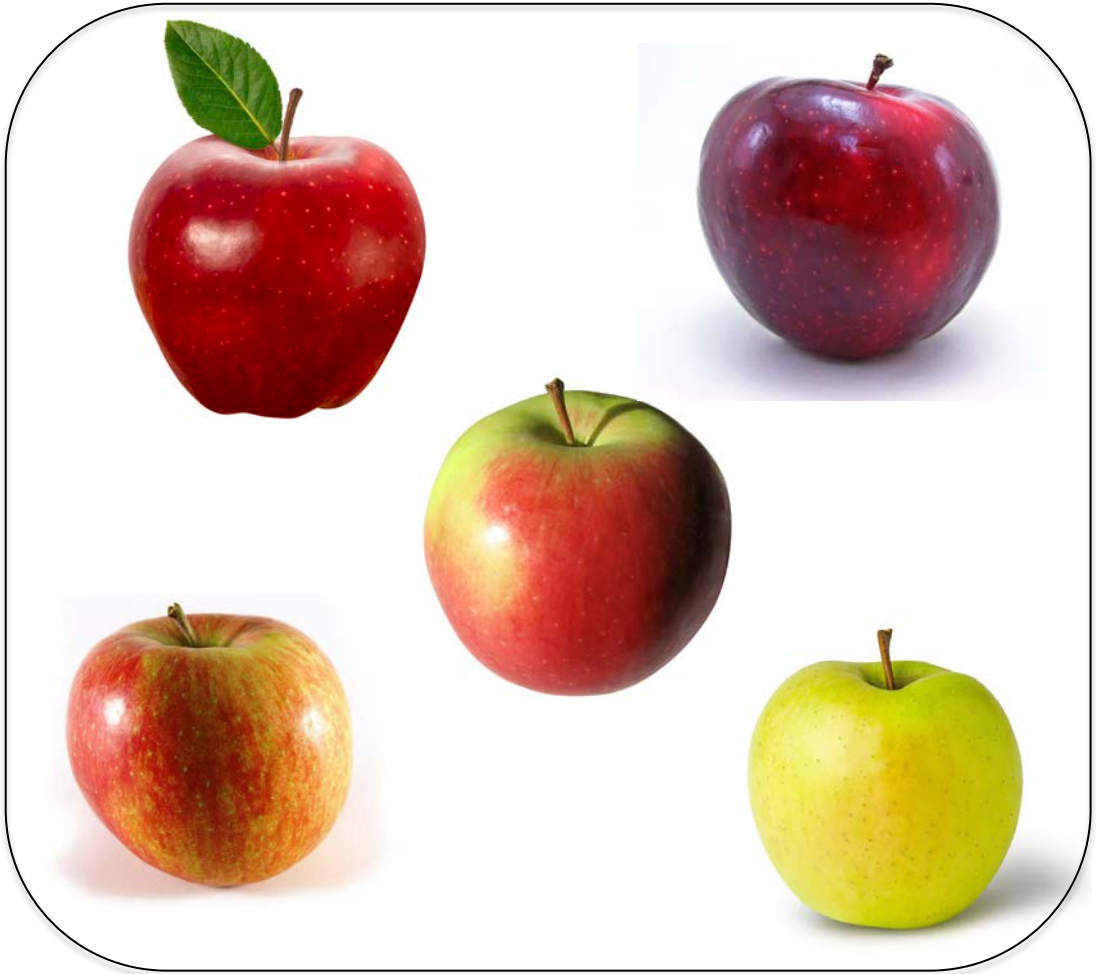
- Distance is a popular strategy for defining how similar to objects are to each other
- It is called distance because it is calculated in the same manner as Euclidean distance
- Importantly, distance takes into account *all* of the properties of the objects in question – it doesn't just focus on one or two
- Only numeric attributes are allowed as input, but it is technically possible to convert categorical attributes to numeric ones
- This only works as long as the categorical concepts are in some sense equidistant from each other, conceptually. Consider as an example where they are not - [apple, pear, vegetable].



How far apart are these apples?

Cluster Properties

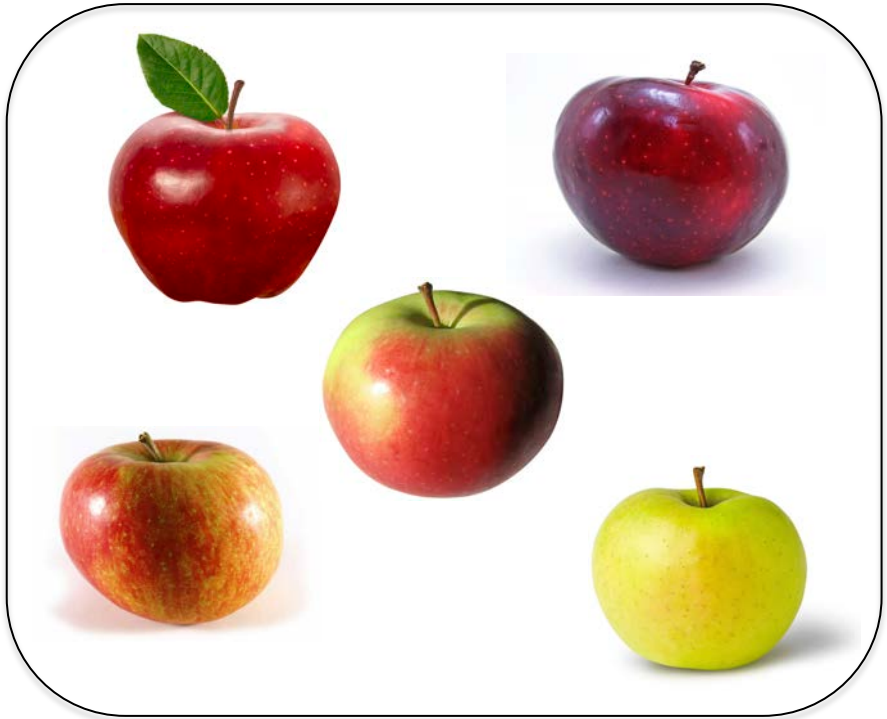
- Number of instances
- Similarity measures across instances within cluster
  - minimum similarity
  - maximum similarity
  - average similarity
- Cluster Representative:
  - may be an instance
  - may be an amalgamation of multiple instances (e.g. exemplar)



Which are the most similar? Which are the least? Which is the best representative?

Cluster – Instance Relationship

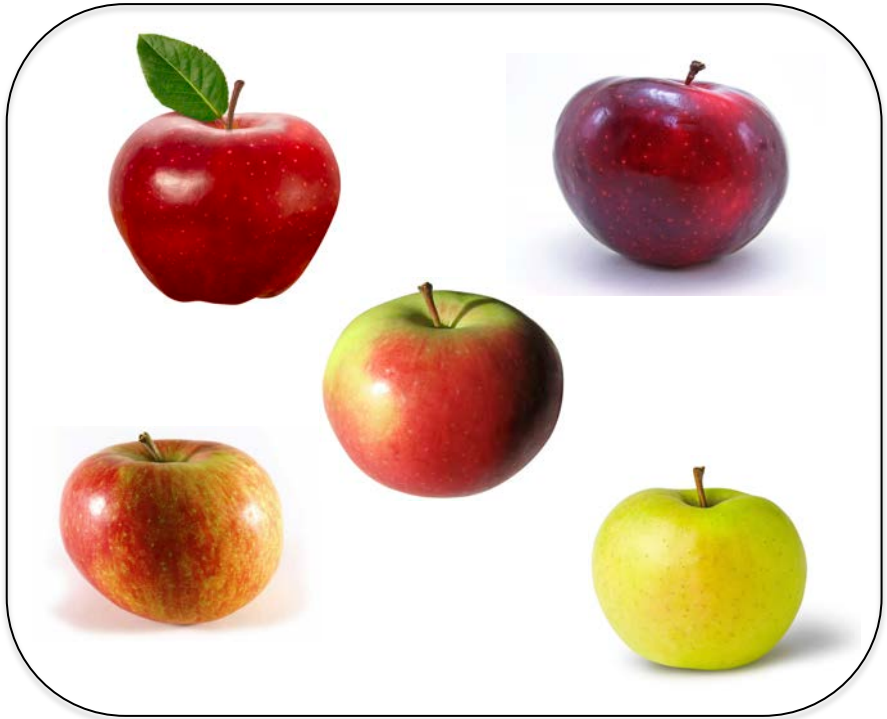
- Comparison of instance to cluster
- Might compare with representative instance
- See also instance instance relationships for comparison between the instance and specific instances within the cluster:
  - instance with cluster instance the greatest distance away from it
  - instance with cluster instance that is most similar



Is this instance similar to this cluster? Does it belong in this cluster?

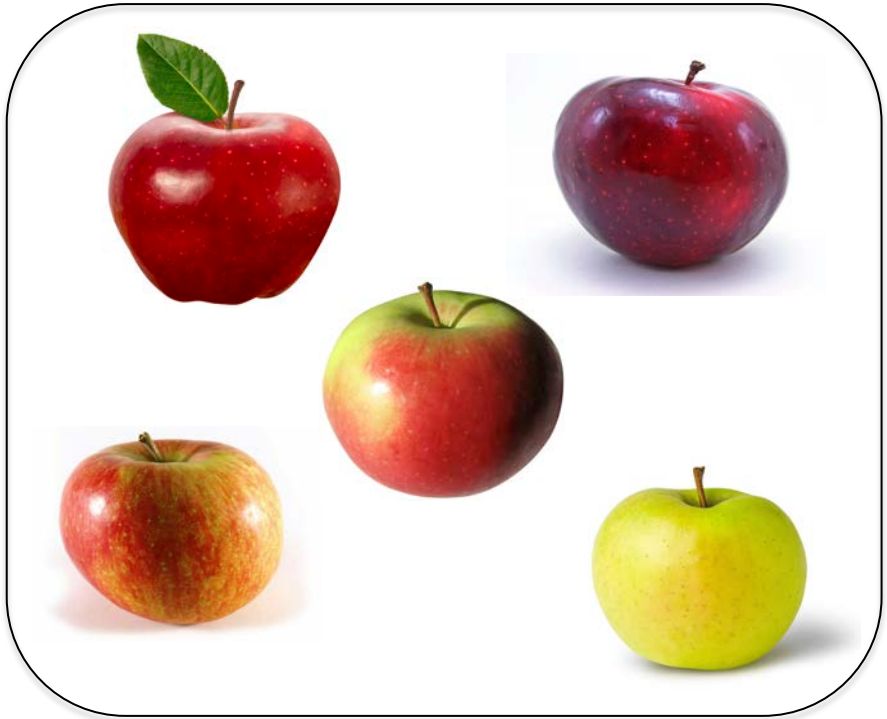
Cluster – Cluster Relationship

- Comparison of cluster level properties:
  - number of instances
  - max or min similarity
  - cluster representatives



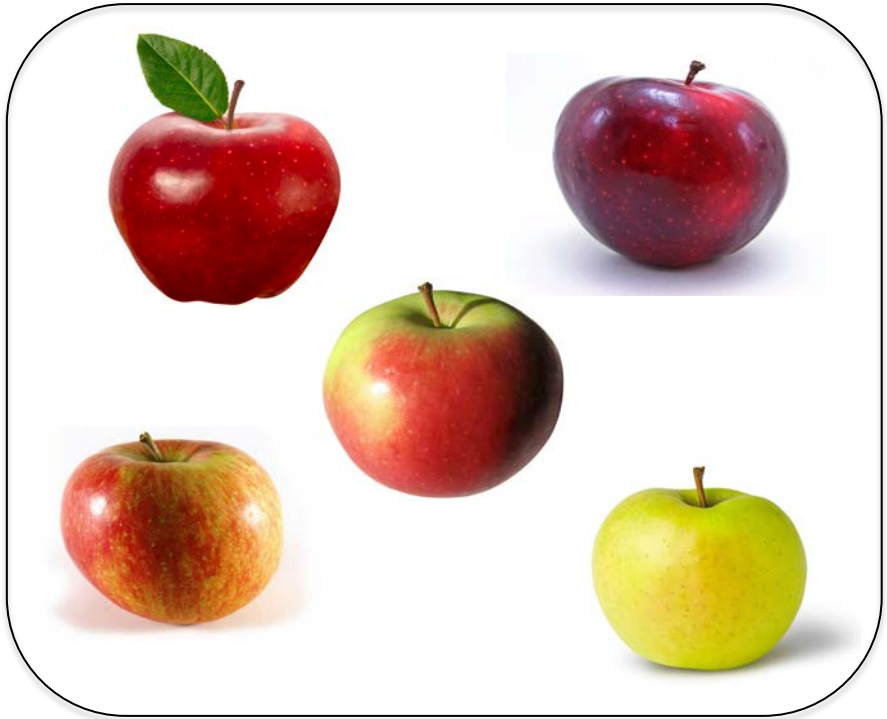
Comparisons Summary

- Comparison of cluster level properties
  - number of instances
  - max or min similarity
  - cluster representatives
- Comparison of cluster to instance properties
  - instance vector to cluster representative vector
- Comparison of instance to instance properties
  - similarity measures
- Comparisons may occur both within cluster and across clusters



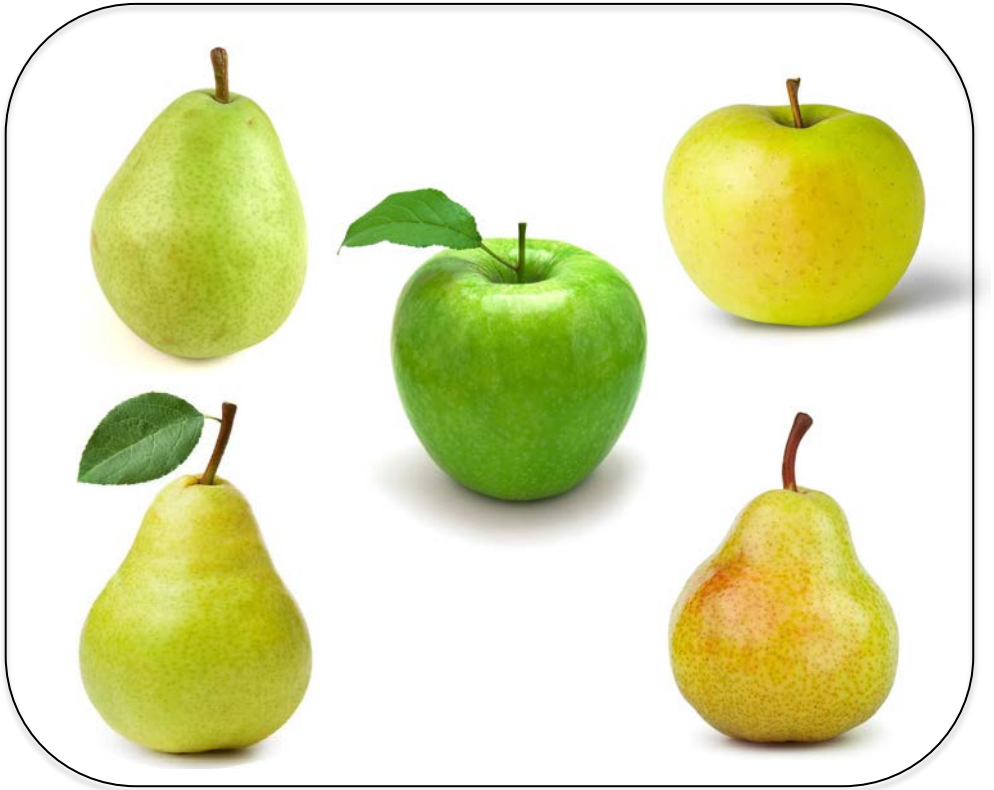
Getting to quality clusters

- Cluster and instance comparisons can be combined in many different ways.
- These can be used to generate a <u>vast number of different cluster validation functions</u>
- What do these tell us about the **quality** of a particular clustering:
  - relative to some objective criteria about good clustering schemes
  - relative to another clustering option
  - relative to external information (e.g. functionality, natural classes)



?

A Quality Clustering? Natural?



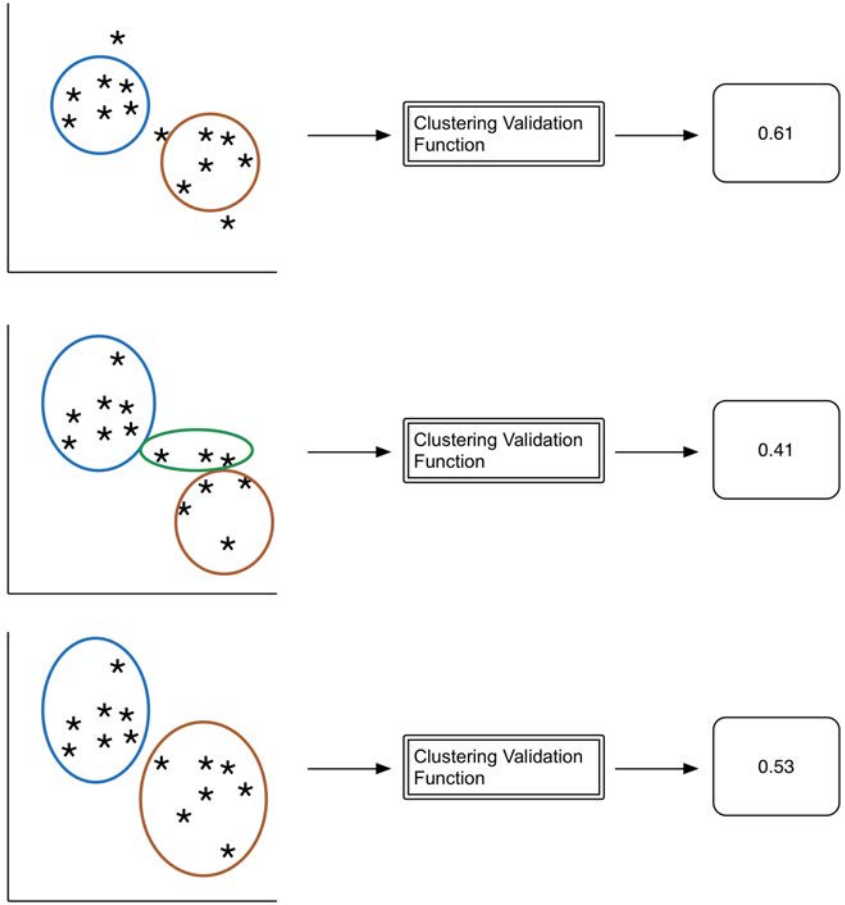
What level of quality is this clustering? Are there higher quality clusterings? Lower? How would you quantify this? Use some of the introduced concepts?

Clustering Validation – Part 3

TYPES OF CLUSTERING VALIDATION

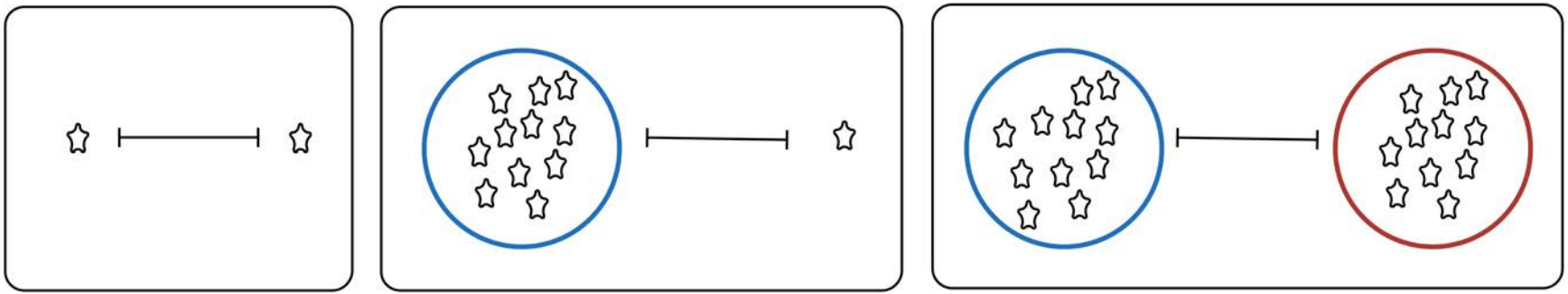
Clustering Operations

- Clustering involves two main activities
  - Creating clusters
  - Assessing cluster quality
- We create functions to carry out both of these activities
- Clustering functions
  - Input: Instances (vectors)
  - Output: Cluster assignment to each instance
- Assessing cluster quality
  - Input: Instances + Cluster Assignments (+ similarity matrix, usually)
  - Output: A numeric value



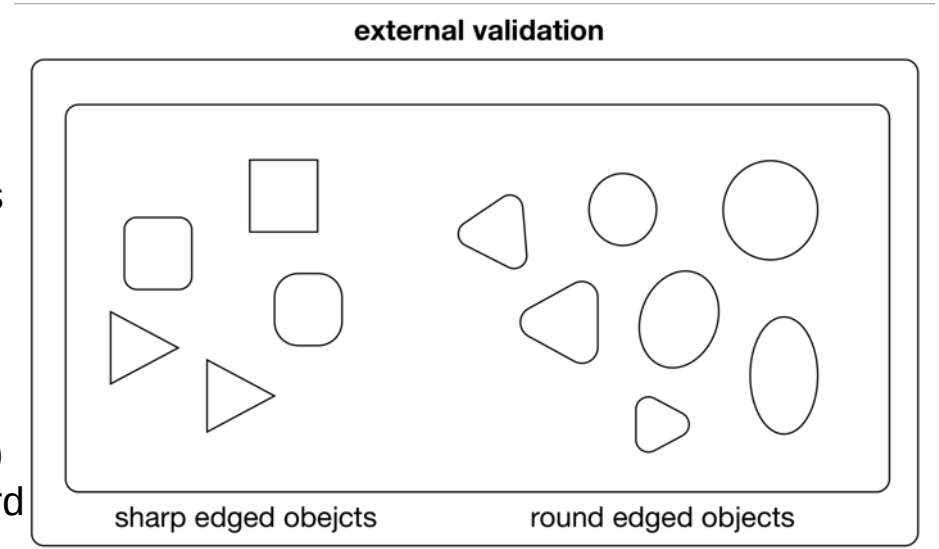
Clustering Validation Function Components

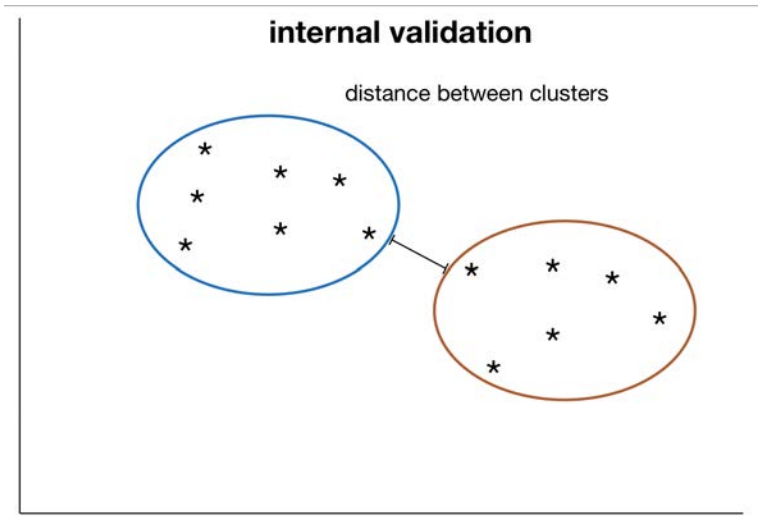
- There are a huge number of both of clustering and cluster validation functions
- However, all are built up out of the basic measures relating to instance or cluster properties we have already reviewed:
  - **Instance Properties**
  - **Cluster Properties**
  - **Instance – Instance relationship properties**
  - **Cluster – Instance Relationship Properties**
  - **Cluster – Cluster Relationship Properties**

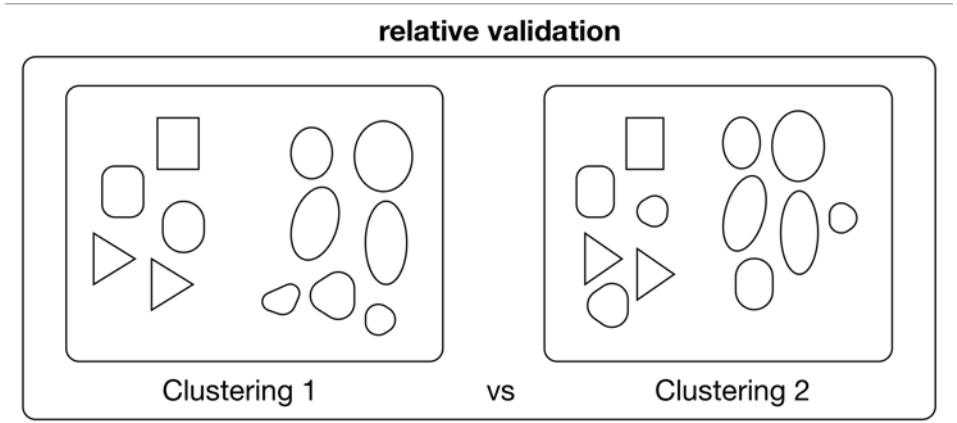


Three types of validation

- **Internal Validation**: Based only on properties available within a single clustering result (note that this comprises multiple clusters)
- **Relative Validation**: Comparison of one (entire) clustering result with another
- **External Validation**: Comparison of a (single) clustering result with some external standard





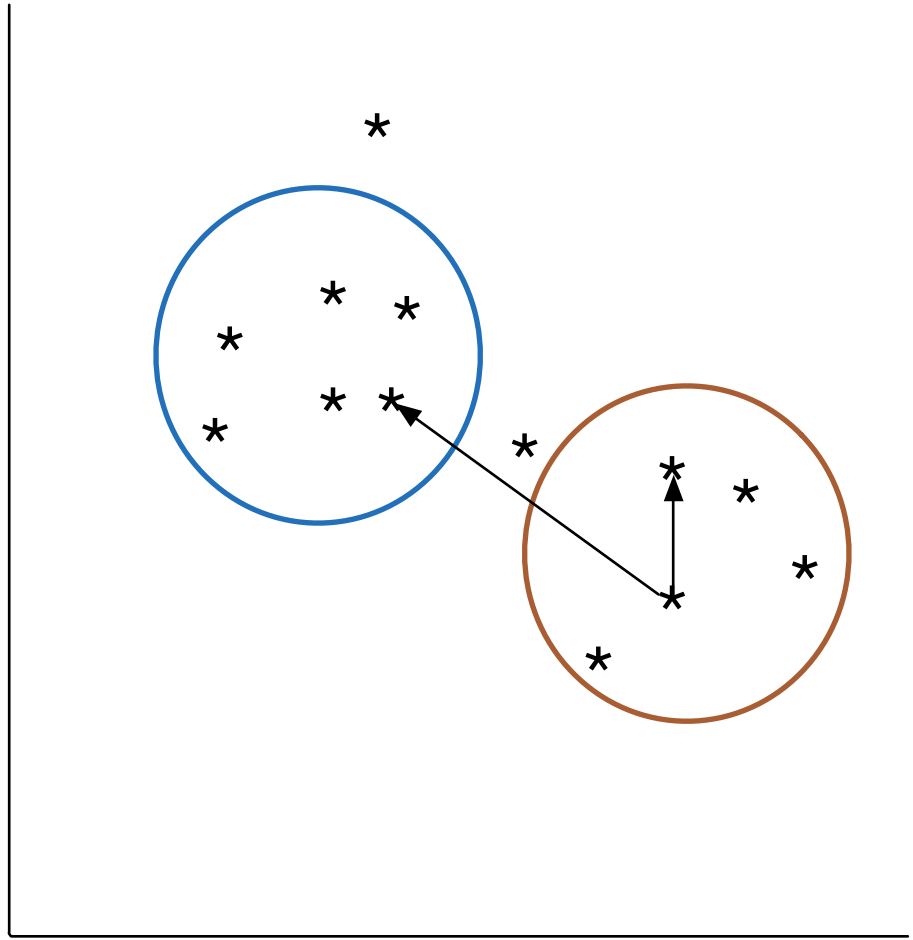


Clustering Validation – Part 4

INTERNAL VALIDATION

Validity vs. Quality

- Context is very relevant to the quality of a given clustering
- BUT what if we have no context?
- Is there a way to <u>objectively measure</u> cluster quality without any specific context?
- The term ‘validity’ suggests there is a **correct** clustering, and all we need to do is see how close we are to that
- Alternatively Lewis, Ackerman and de Sa (2012) use the term **Clustering Quality Measures** (CQM) instead



A (Small?) Sample of Internal CQMs

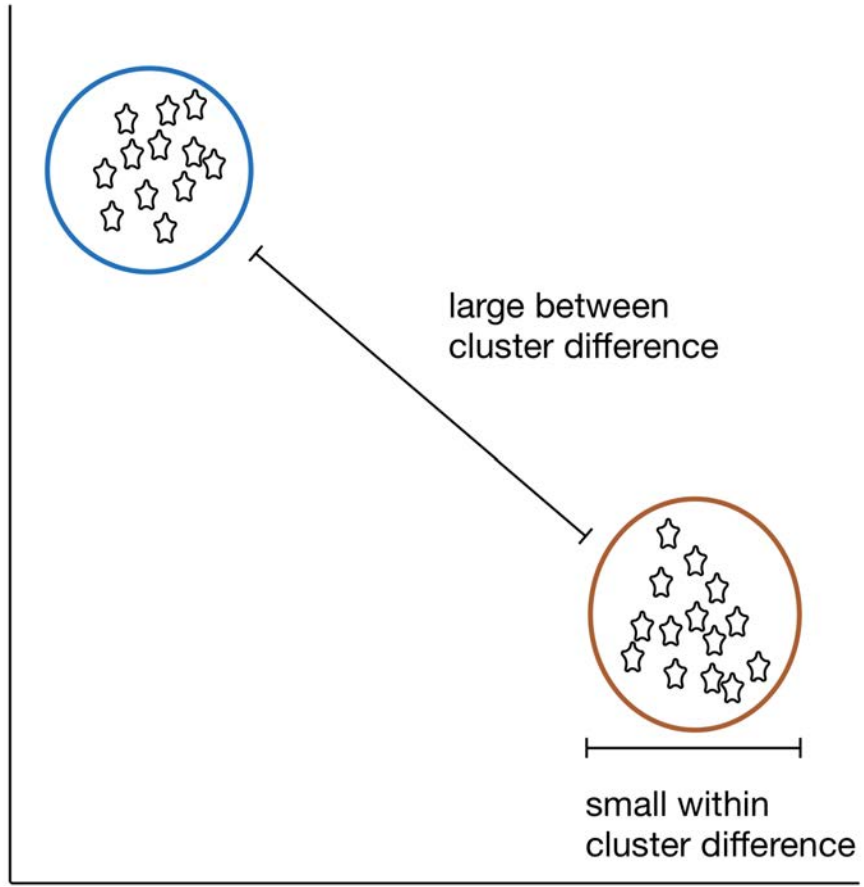
- The Ball-Hall index
- The Banfeld-Raftery index
- The C index
- The Calinski-Harabasz index
- The Davies-Bouldin index
- The Det Ratio index
- The Dunn index
- The Baker-Hubert Gamma index
- The GDI index
- The Gplus index
- The KsqDetW index
- The LogDetRatio index
- The LogSSRatio index
- The McClain-Rao index
- The PBM index
- The Point-Biserial index
- The Ratkowsky-Lance index
- The Ray-Turi index
- The Scott-Symons index
- The SD index
- The SDbw index
- The Silhouette index
- The Tau index
- The TraceW index
- The TraceWiB index
- The Wemmert-Gancˇarski index
- The Xie-Beni index

(These are all defined and available in the clusterCrit package for R)

What are we to make of all these different, <u>supposedly context free</u> measures of clustering quality?

Very Broad Goals

- Within clusters, everything is very similar.
- Between clusters, there is a lot of difference.
- The problem: there are many ways for clusters to deviate from this ideal.
- In specific clustering cases, how do we weigh the good aspects (e.g. high within cluster similarity) relative to the bad (e.g. low between cluster separation)
- Thus the large number of CQMs
- Question: is this trade-off (and the resulting CQMs) really context independent?
- Maybe different weightings are more relevant in different contexts?



Comparing measures across datasets

Vendramin et al 2010 used a number of benchmark tests to compared a large number of intrinsic validation measures

Broad conclusion: variants of Silhouette performed well across tests

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Point-Biserial	A	0.000	0.046	0.226	0.247	0.262	0.289	0.306	0.373	0.390	0.408	0.488	0.555	0.566	0.571	0.584	0.636	0.642	0.645	0.694	0.705	0.729	0.736	0.768	0.822	0.837	1.107
Tau	B	-0.046	0.000	0.180	0.201	0.216	0.243	0.260	0.327	0.344	0.362	0.442	0.509	0.520	0.525	0.538	0.590	0.597	0.599	0.649	0.659	0.683	0.690	0.722	0.776	0.791	1.061
$C/k^{1/2}$	C	-0.226	-0.180	0.000	0.021	0.036	0.063	0.080	0.147	0.164	0.182	0.263	0.329	0.340	0.345	0.358	0.410	0.417	0.419	0.469	0.479	0.504	0.510	0.542	0.596	0.611	0.881
ASWC	D	-0.247	-0.201	-0.021	0.000	0.015	0.042	0.060	0.127	0.143	0.161	0.242	0.308	0.319	0.324	0.338	0.390	0.396	0.398	0.448	0.458	0.483	0.489	0.521	0.575	0.590	0.860
ASSWC	E	-0.262	-0.216	-0.036	-0.015	0.000	0.027	0.045	0.112	0.128	0.146	0.227	0.293	0.304	0.309	0.323	0.375	0.381	0.384	0.433	0.443	0.468	0.474	0.506	0.560	0.575	0.846
PBM	F	-0.289	-0.243	-0.063	-0.042	-0.027	0.000	0.017	0.084	0.101	0.119	0.199	0.266	0.277	0.282	0.295	0.347	0.353	0.356	0.406	0.416	0.440	0.447	0.479	0.533	0.548	0.818
SWC	G	-0.306	-0.260	-0.080	-0.060	-0.045	-0.017	0.000	0.067	0.083	0.102	0.182	0.249	0.260	0.265	0.278	0.330	0.336	0.339	0.388	0.399	0.423	0.430	0.462	0.516	0.530	0.801
SSWC	H	-0.373	-0.327	-0.147	-0.127	-0.112	-0.084	-0.067	0.000	0.016	0.035	0.115	0.181	0.193	0.198	0.211	0.263	0.269	0.272	0.321	0.332	0.356	0.363	0.395	0.449	0.463	0.734
Dunn12	I	-0.390	-0.344	-0.164	-0.143	-0.128	-0.101	-0.083	-0.016	0.000	0.018	0.099	0.165	0.176	0.181	0.195	0.247	0.253	0.255	0.305	0.315	0.340	0.346	0.378	0.432	0.447	0.717
Dunn62	J	-0.408	-0.362	-0.182	-0.161	-0.146	-0.119	-0.102	-0.035	-0.018	0.000	0.080	0.147	0.158	0.163	0.176	0.228	0.234	0.237	0.287	0.297	0.321	0.328	0.360	0.414	0.429	0.699
Dunn13	K	-0.488	-0.442	-0.263	-0.242	-0.227	-0.199	-0.182	-0.115	-0.099	-0.080	0.000	0.066	0.078	0.082	0.096	0.148	0.154	0.157	0.206	0.217	0.241	0.248	0.280	0.334	0.348	0.619
VRC	L	-0.555	-0.509	-0.329	-0.308	-0.293	-0.266	-0.249	-0.181	-0.165	-0.147	-0.066	0.000	0.011	0.016	0.030	0.082	0.088	0.090	0.140	0.150	0.175	0.181	0.213	0.267	0.282	0.552
Ball and Hall	M	-0.566	-0.520	-0.340	-0.319	-0.304	-0.277	-0.260	-0.193	-0.176	-0.158	-0.078	-0.011	0.000	0.005	0.018	0.070	0.076	0.079	0.129	0.139	0.163	0.170	0.202	0.256	0.271	0.541
Trace(W)	N	-0.571	-0.525	-0.345	-0.324	-0.309	-0.282	-0.265	-0.198	-0.181	-0.163	-0.082	-0.016	-0.005	0.000	0.013	0.065	0.072	0.074	0.124	0.134	0.159	0.165	0.197	0.251	0.266	0.536
DB	O	-0.584	-0.538	-0.358	-0.338	-0.323	-0.295	-0.278	-0.211	-0.195	-0.176	-0.096	-0.030	-0.018	-0.013	0.000	0.052	0.058	0.061	0.110	0.121	0.145	0.152	0.184	0.238	0.252	0.523
Nlog(|T|/|W|)	P	-0.636	-0.590	-0.410	-0.390	-0.375	-0.347	-0.330	-0.263	-0.247	-0.228	-0.148	-0.082	-0.070	-0.065	-0.052	0.000	0.006	0.009	0.058	0.069	0.093	0.100	0.132	0.186	0.200	0.471
Trace(CovW)	Q	-0.642	-0.597	-0.417	-0.396	-0.381	-0.353	-0.336	-0.269	-0.253	-0.234	-0.154	-0.088	-0.076	-0.072	-0.058	-0.006	0.000	0.003	0.052	0.063	0.087	0.094	0.126	0.180	0.194	0.465
$k^2\|W\|$	R	-0.645	-0.599	-0.419	-0.398	-0.384	-0.356	-0.339	-0.272	-0.255	-0.237	-0.157	-0.090	-0.079	-0.074	-0.061	-0.009	-0.003	0.000	0.049	0.060	0.084	0.091	0.123	0.177	0.192	0.462
log(SSB/SSW)	S	-0.694	-0.649	-0.469	-0.448	-0.433	-0.406	-0.388	-0.321	-0.305	-0.287	-0.206	-0.140	-0.129	-0.124	-0.110	-0.058	-0.052	-0.049	0.000	0.010	0.035	0.041	0.074	0.128	0.142	0.413
Dunn11	T	-0.705	-0.659	-0.479	-0.458	-0.443	-0.416	-0.399	-0.332	-0.315	-0.297	-0.217	-0.150	-0.139	-0.134	-0.121	-0.069	-0.063	-0.060	-0.010	0.000	0.024	0.031	0.063	0.117	0.132	0.402
Gamma	U	-0.729	-0.683	-0.504	-0.483	-0.468	-0.440	-0.423	-0.356	-0.340	-0.321	-0.241	-0.175	-0.163	-0.159	-0.145	-0.093	-0.087	-0.084	-0.035	-0.024	0.000	0.007	0.039	0.093	0.107	0.378
McClain and Rao	V	-0.736	-0.690	-0.510	-0.489	-0.474	-0.447	-0.430	-0.363	-0.346	-0.328	-0.248	-0.181	-0.170	-0.165	-0.152	-0.100	-0.094	-0.091	-0.041	-0.031	-0.007	0.000	0.032	0.086	0.101	0.371
C-Index	W	-0.768	-0.722	-0.542	-0.521	-0.506	-0.479	-0.462	-0.395	-0.378	-0.360	-0.280	-0.213	-0.202	-0.197	-0.184	-0.132	-0.126	-0.123	-0.074	-0.063	-0.039	-0.032	0.000	0.054	0.069	0.339
|T|/|W|	X	-0.822	-0.776	-0.596	-0.575	-0.560	-0.533	-0.516	-0.449	-0.432	-0.414	-0.334	-0.267	-0.256	-0.251	-0.238	-0.186	-0.180	-0.177	-0.128	-0.117	-0.093	-0.086	-0.054	0.000	0.015	0.285
$Trace(W^{-1}B)$	Y	-0.837	-0.791	-0.611	-0.590	-0.575	-0.548	-0.530	-0.463	-0.447	-0.429	-0.348	-0.282	-0.271	-0.266	-0.252	-0.200	-0.194	-0.192	-0.142	-0.132	-0.107	-0.101	-0.069	-0.015	0.000	0.270
G(+)	Z	-1.107	-1.061	-0.881	-0.860	-0.846	-0.818	-0.801	-0.734	-0.717	-0.699	-0.619	-0.552	-0.541	-0.536	-0.523	-0.471	-0.465	-0.462	-0.413	-0.402	-0.378	-0.371	-0.339	-0.285	-0.270	0.000
	Mean	0.959	0.913	0.733	0.712	0.697	0.670	0.653	0.586	0.569	0.551	0.471	0.404	0.393	0.388	0.375	0.323	0.316	0.314	0.264	0.254	0.230	0.223	0.191	0.137	0.122	-0.148

Fig. 10 Mean values (bottom bar) and their differences (cells) for Pearson correlation between relative and external (Jaccard) criteria: $k_{\max} = 25$.

Machine Learning vs Human Learning

- Lewis et al compare 6 common CQMs with human evaluation of clustering results
- Main finding: Human clustering evaluation was most similar to Silhouette and Calinski-Harabasz
- Maybe internal validation/CQM is saying something about clustering across all contexts?
- Maybe easier to identify the clearly bad than all the variations of good?

Table 1: Correlation coefficients between human responses and CQMs with k factored out (except for the k column). Text in bold (excluding k column) if $p < .0025$ after Bonferroni correction for $n = 20$ comparisons per subject group and $\alpha = .05$.

ρ	Expert Positive	Expert Negative	Novice Positive	Novice Negative	Gamma	Silhouette	Dunn	Avg Within	Avg Btw	CH	W-Inter/Intra	k
Expert Pos	1	**-.35**	**.56**	-.19	-.15	**.46**	**.40**	**-.39**	**.34**	**.44**	.19	-.43
Expert Neg		1	-.13	**.44**	.09	**-.27**	-.12	**.44**	-.18	**-.36**	**-.30**	.32
Novice Pos			1	-.04	-.13	**.39**	**.40**	-.20	.23	**.30**	.04	-.73
Novice Neg				1	.08	**-.27**	.01	**.30**	-.07	**-.25**	**-.27**	.71

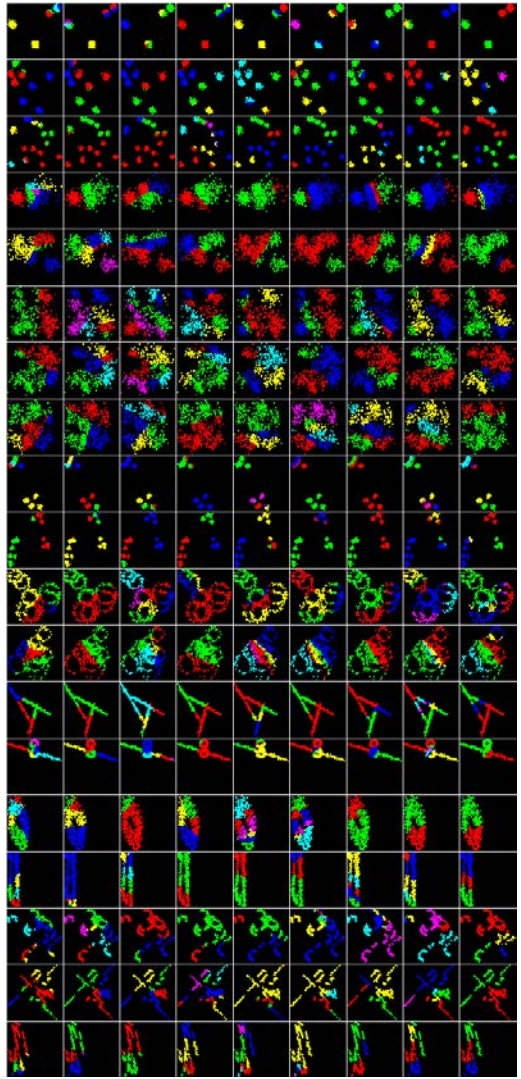
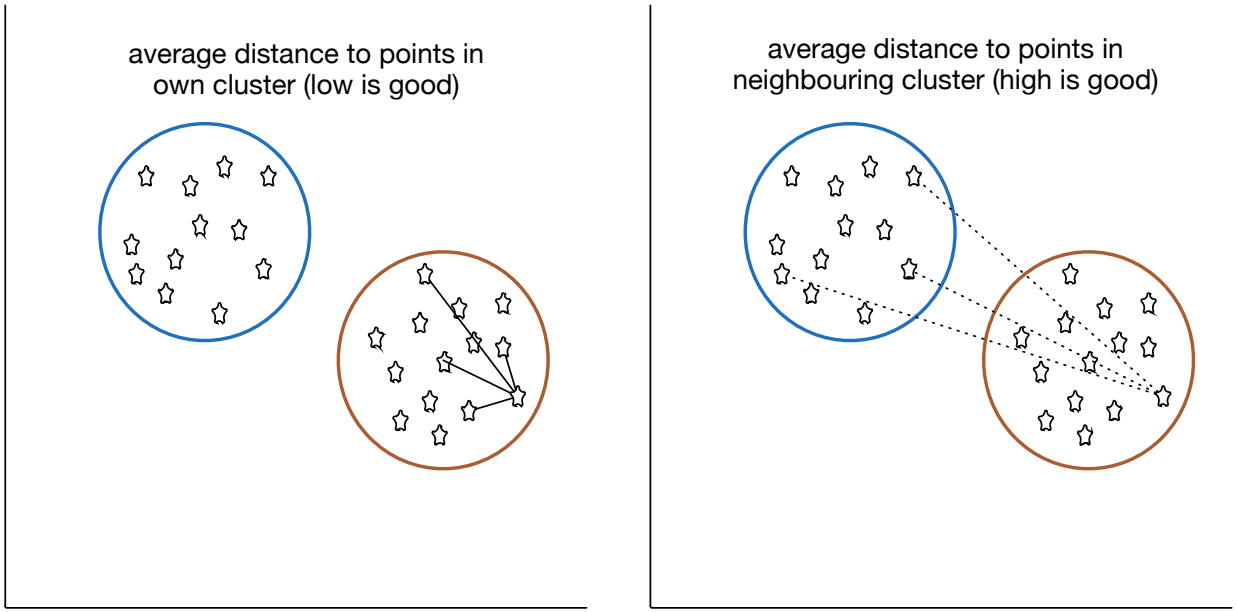


Figure 1: All stimuli. Datasets are in rows; partitions are in columns.

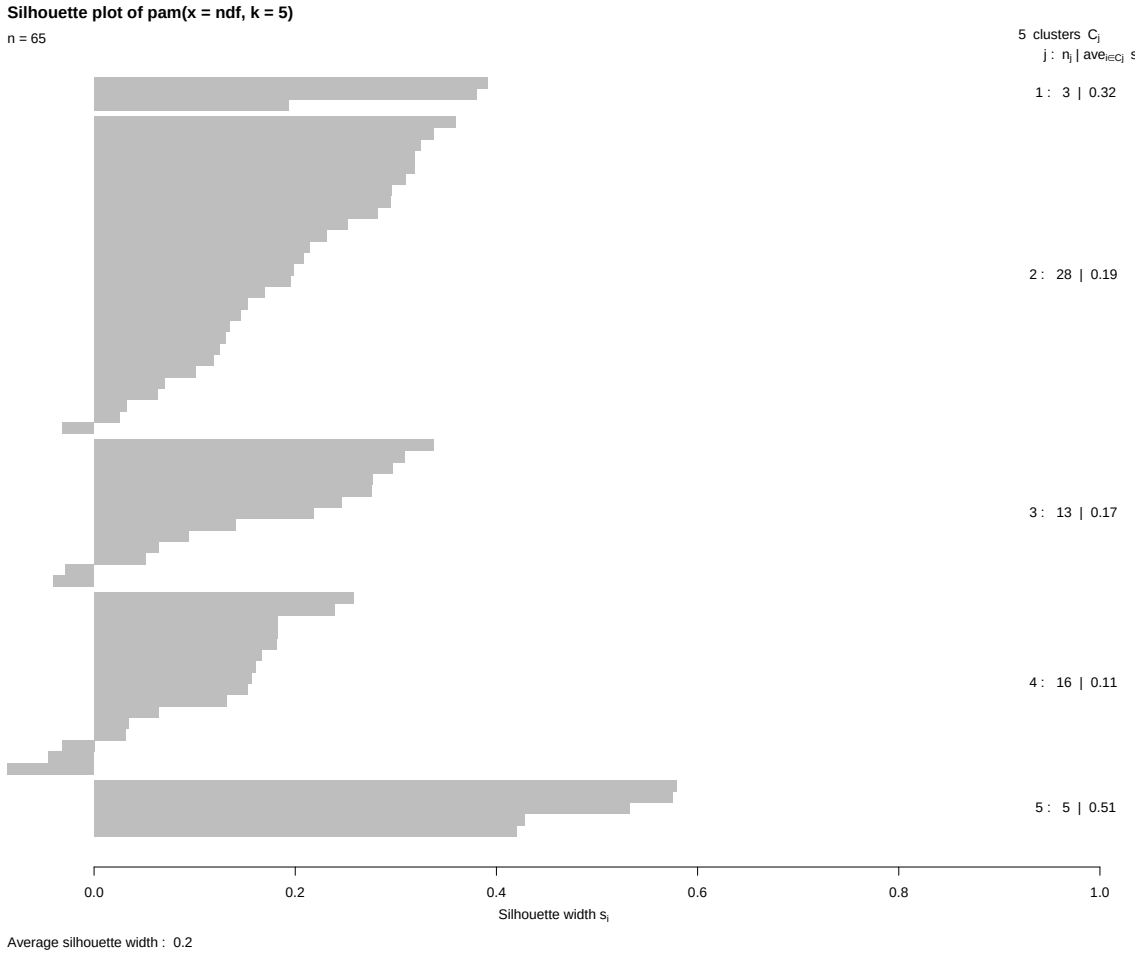
Silhouette Index: Algorithm



$$\text{silhouette metric for a point} = \frac{\text{(average dissimiliarity with neighbouring cluster - average dissimilarity with own cluster)}}{\text{maximum dissimilarity value (own or neighbour)}}$$

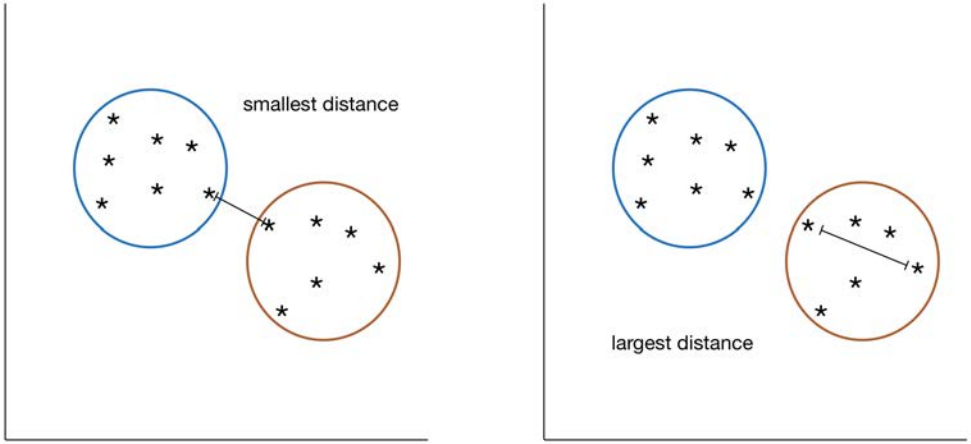
A strong internal validation metric that incorporates a number of measures.

Silhouette Metric Sample Results (I)



Dunn's Index: Algorithm

- Within a cluster, the size of the cluster (e.g. greatest distance between points)
- Between two clusters, the distance between the clusters (e.g. minimum distance between points)
- Ratio: The minimum intercluster distance across all pairs of clusters / maximum intracluster distance across all clusters
- A number of possible ways to define inter cluster distance and cluster size.



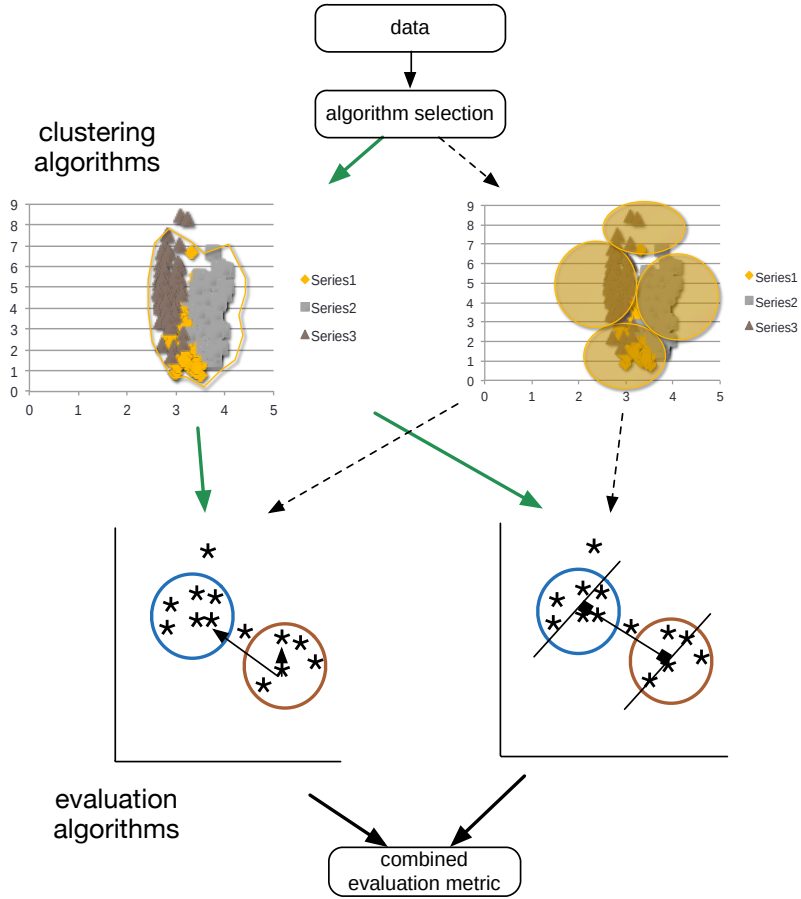
Comparison with Silhouette Index: In a sense, a simpler measure. More of a whole cluster measure, rather than a point by point measure. Evaluates based on extremes (max and min).

Clustering Validation – Part 5

RELATIVE VALIDATION

More is better?

- Getting a single validation measure for a single clustering is not that useful – could the results be better? Is this the best we can hope for?
- How about comparing results across runs or parameter settings?
- Main emphasis with relative validation is how to compare results of individual runs.



Correlation Measures

- Look at correlation between clustering assignments
- Rand, Jaccard, Gamma
- Perfect correlation gives maximum value of the measure

	P1	P2	P3	P4	P5	P6
P1	1					
P2	0	1				
P3	1	0	1			
P4	1	0	1	1		
P5	0	1	0	0	1	
P6	0	0	0	0	0	1

	P1	P2	P3	P4	P5	P6
P1	1					
P2	0	1				
P3	1	0	1			
P4	1	0	1	1		
P5	0	1	0	0	1	
P6	0	1	0	0	1	1

Two very similar clustering results (but notice they vary in the number of clusters)

Rand's Index

- SS- the number of pairs of items belonging to the same cluster in both clusterings (C1 = 1 and C2 = 1)
- SD- the number of pairs together in one clustering but not the other (C1 = 1 and C2 = 0)
- DS- the number of pairs not together in one clustering but together in the other (C1 = 0 and C2 = 1)
- DD- the number of pairs not together in either cluster (C1 = 0 and C2 = 0)
- Note that SS and DD are good, and DS, SD are bad.
- Rand Index is the ratio of SS + DD to the total number of pairs. If Rand Index = 1, the clustering perfectly matches the gold standard.

Clustering 1 (C1)

	P1	P2	P3	P4	P5	P6
P1	1					
P2	0	1				
P3	1	0	1			
P4	1	0	1	1		
P5	0	1	0	0	1	
P6	0	0	0	0	0	1

Clustering 2 (C2)

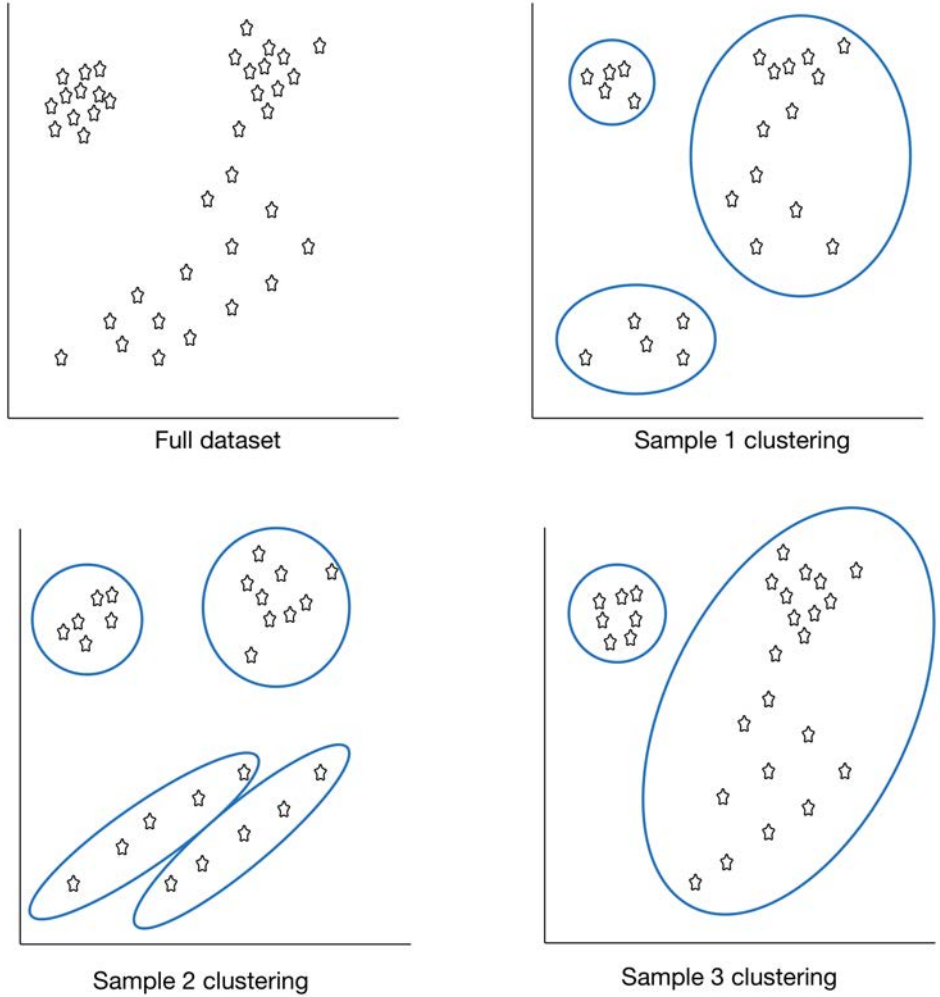
	P1	P2	P3	P4	P5	P6
P1	1					
P2	0	1				
P3	1	0	1			
P4	1	0	1	1		
P5	0	1	0	0	1	
P6	0	1	0	0	1	1

$$(SS + DD)/(SS + DD + SD + DS)$$

$$(4 + 9)/(4 + 9 + 0 + 2) = 0.87$$

Stability

- Some options:
  - multiple datasets sampled from same source
  - different columns used to generate clusters (i.e. drop a different column each time)
- Similarity of results is measured
- If results are not stable across clustering schemes, further investigation required

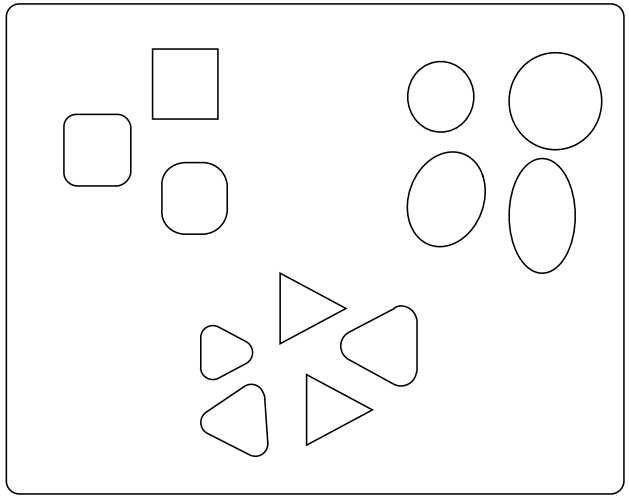


Clustering Validation – Part 6

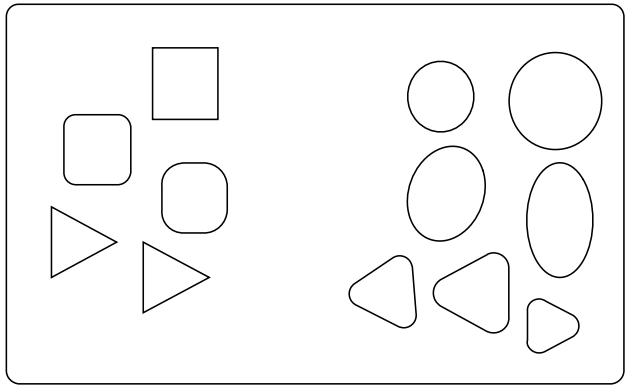
EXTERNAL VALIDATION

Back to Context

- Brings in outside information to evaluate the clusters
- Outside information is typically the ‘correct’ class
- How is this different from classification then?
- Often used to build confidence in the <u>overall approach</u>, based on preliminary or sample results



Natural Groupings

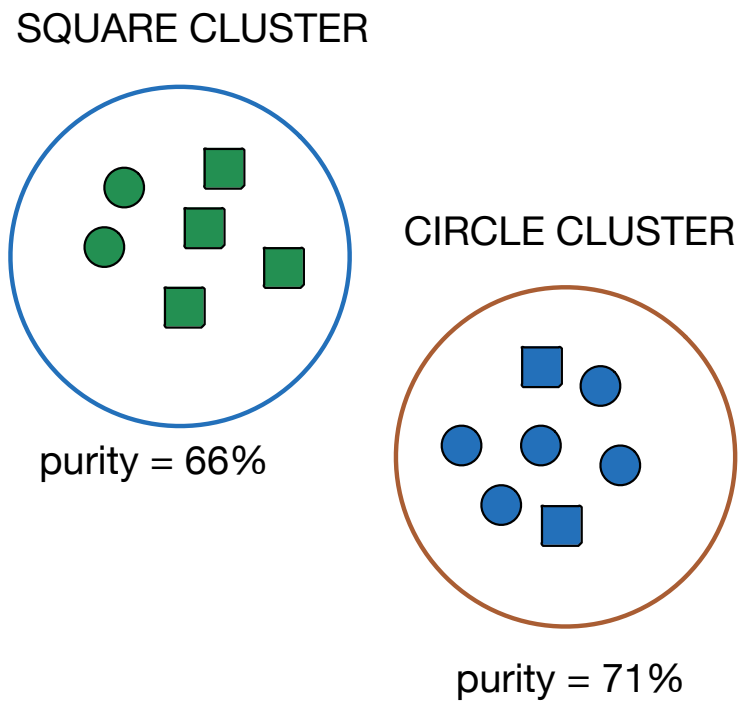


Clustering Results

Example Metric: Purity

- For this metric each cluster is assigned to the class which is most frequent in the cluster
- To calculate the purity: number of correctly assigned points / number of points in the cluster
- Some other options: precision, recall

Assuming we are interested in shape…



Types of External Validation

- Amigó et al (2009) provide a number of constraints for external validation measures
- They suggest external evaluation strategies can be based on:
  - set matching
  - counting pairs
  - entropy measures
  - edit distance
- Similar to strategies used to evaluate classification
- They recommend using a particular version (Bcubed) of precision and recall for external validation, as these best take into consideration the 4 constraints

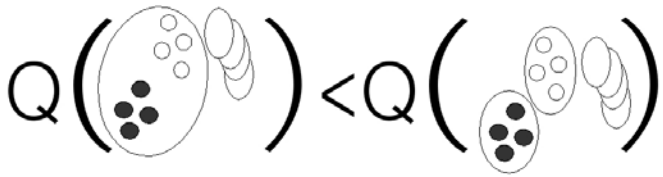


Figure 1: Constraint 1: Cluster Homogeneity

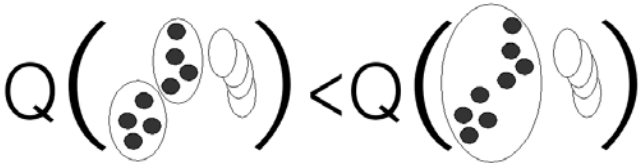


Figure 2: Constraint 2: cluster completeness

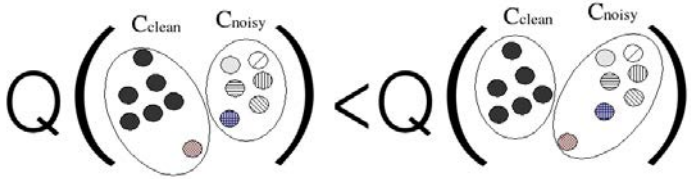


Figure 3: Constraint 3: Rag Bag

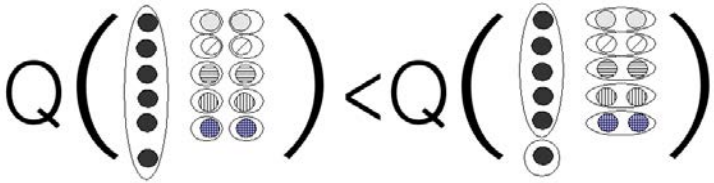


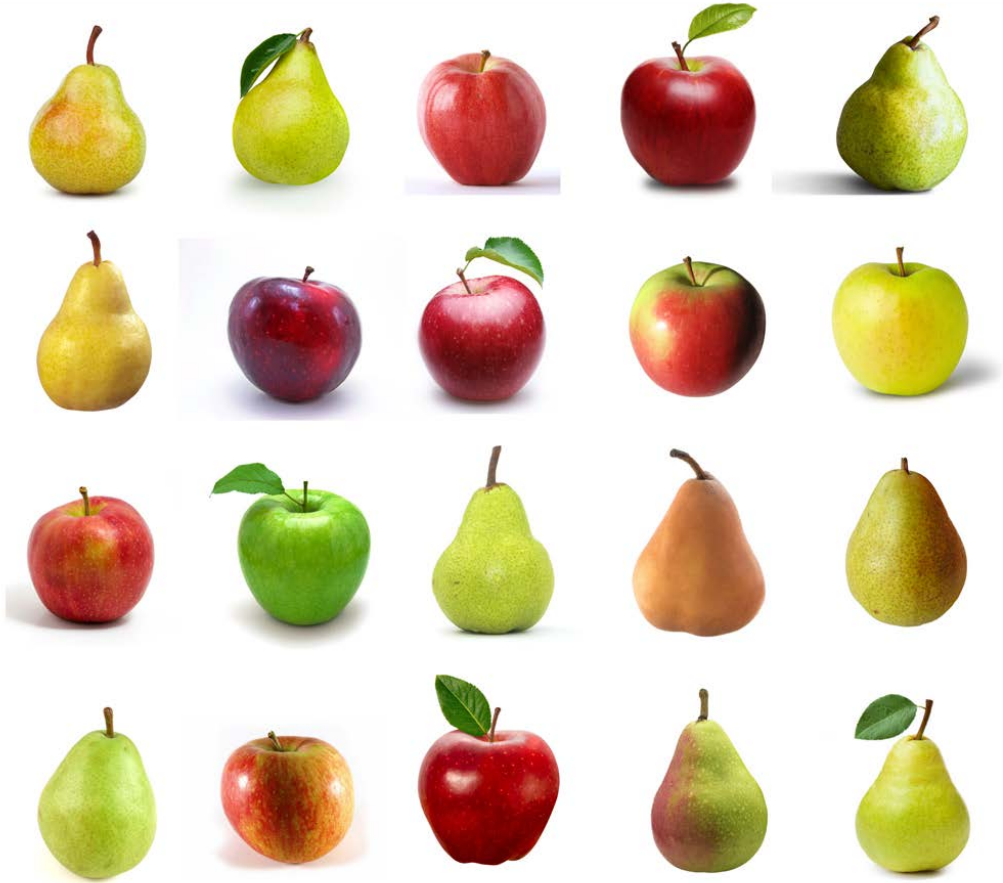
Figure 4: Clusters Size vs. Quantity

Clustering Validation – Part 7

CONCLUDING THOUGHTS

Try and Try Again

- A large amount of diversity in clustering validation techniques
- Be aware of the types of validation, and variations within types
- Seek agreement across techniques, ok to compare
- There are many ways for a clustering to be ‘ok’ – you need to decide what is important and what can be ignored
- A lot depends on context



References

References

Hierarchical Clustering

- *hclust {stats},* R Documentation: Hierarchical Clustering, from Package stats version 3.3.0., by R Core Team and contributors worldwide. Retrieved 2016.10.11
- Wikipedia: *Hierarchical clustering*. Last edited 2016-09-16. Retrieved 2016.10.11.
- *Hierarchical agglomerative clustering*, in Introduction to Information Retrieval, by C.D. Manning, P. Raghavan and H. Schütze. Published (online version) 2009.04.07
- *Hierarchical Clustering with R (feat. D3.js and Shiny)*, by R. Vogler. Published 2014-12-14. Retrieved 2016.10.11.
- *Introduction to Visualizing Hierarchies*, by R. Mazza, D. Brodbeck, M. Lanza and R. Wettel. Retrieved 2016.10.11.
- *Hierarchical cluster analysis, in Multivariate Analysis of Ecological Data*, by M. Greenacre and R. Primicerio. Published 2013, by Fundación BBVA, 2013, Plaza de San Nicolás, 4 48005 Bilbao.
- *TIBCO Spotfire Documentation: What is a Treemap?* Last edited: 2015-02-12. Retrieved 2016.10.11.

References

Clustering Evaluation

- Wikipedia: *Silhouette (clustering)*. Last edited 2016-07-15. Retrieved 2016.10.11.
- *silhouette {cluster} Compute or Extract Silhouette Information from Clustering*, from Package cluster version 2.0.3, by M. Maechler, P. Rousseeuw, A. Struyf and M. Hubert. Published 2016-10-08.
- *Relative Clustering Validity Criteria: A Comparative Overview* by L. Vendramin, R.J.G.B. Campello and E.R. Hruschka. Published online 2010-06-30, by Wiley InterScience.

Similarity and Dissimilarity Metrics

- *A General Coefficient of Similarity and Some of Its Properties*, by J. C. Gower. Biometrics, 1971.
- *Dissimilarity Measures*, in A Guide to Statistical Analysis in Microbial Ecology: a community-focused, living review of multivariate data analyses. by P.L. Buttigieg and A. Ramette.
- *Cosine similarity, Pearson correlation, and OLS coefficients*, by B. O'Connor.
- *Data Mining Portfolio Similarity and Dissimilarity Measures*, by G. Benoît.
- *Measures of distance and correlation between variables*, in Multivariate Analysis of Ecological Data, by M. Greenacre and R. Primicerio. Published 2013.

References

Hierarchical Clustering Examples

- *Scientists Trace Society's Myths to Primordial Origins*, by J. d'Huy. In Scientific American (Online). Published 2016-09-29. Retrieved 2016-10-11.
- *Complex building's energy system operation patterns analysis using bag of words representation with hierarchical clustering*, by U. Habib, K. Hayat and G. Zucker. Complex Adaptive Systems Modeling, 2016, 4:8.
- *A Comparison of Antioxidant, Antibacterial, and Anticancer Activity of the Selected Thyme Species by Means of Hierarchical Clustering and Principal Component Analysis*, by M. Orłowska, K. Pytlakowskae, A. Mrozek-Wilczkiewicz, R. Musioł, M. Waksmundzka-Hajnos, M. Sajewicz, T. Kowalska.Acta Chromatographica, 2016, 28.
- *Use of hierarchical cluster analysis to classify prisons in Ireland into mutually exclusive drug-use risk categories*, by M. Codd, J. Mehegan, C. Kelleher, A. Drummond.
- *Divisive Analysis (DIANA) of hierarchical clustering and GPS data for level of service criteria of urban streets*, by A.K. Patnai, P.K. Bhuyan, K.V.K. Rao.

References

DBSCAN

- Clusters and DBScan, by Jesse Johnson.Published 2013-08-20. Retrieved 2016.10.11.
- Scikit Documentation: Comparing different clustering algorithms on toy datasets. By scikit-learn developers. Retrieved 2016.10.11.
- Data Mining TNM033 Notes: DBSCAN A Density-Based Spatial Clustering of Application with Noise, by Henrik Bäcklund, Anders Hedblom and Niklas Neijman. Published 2011-11-30. Retrieved 2016.10.11.
- Scatterplot3d: an R package for Visualizing Multivariate Data, by Uwe Ligges and Martin Mächler. Published 2003. Retrieved 2016.10.11.
- *Wikipedia* article for DBSCAN
- Schubert, Erich; Sander, Jörg; Ester, Martin; Kriegel, Hans Peter; Xu, Xiaowei (July 2017). "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN", *ACM Trans. Database Syst.* **42** (3): 19:1–19:21. doi:10.1145/3068335, ISSN 0362-5915

References

Big O Notation

- *Big O Notation*, by Parker Phinney. Retrieved 2016.10.11.

DBSCAN Case Studies

- Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease, by C. Plant, S.J. Teipel, A. Oswald, C. Böhm, T. Meindl, J. Mourao-Miranda, A.W. Bokde, H. Hampel, M. Ewers.
- A Novel Approach for Predicting the Length of Hospital Stay With DBSCAN and Supervised Classification Algorithms, by Panchami V.U., N. Radhika, A.V. Vidyapeetham.
- Simulation of DNA damage clustering after proton irradiation using an adapted DBSCAN algorithm, by Z. Francis, C. Villagrasa, I. Clairand.
- Where traffic meets DNA: mobility mining using biological sequence analysis revisited, by A. Jawad, K. Kersting, N.V. Andrienko.
- Individual movements and geographical data mining. clustering algorithms for highlighting hotspots in personal navigation routes, by G. Schoier, G. Borruso.

References

Spectral Clustering

- Ulrike von Luxburg, *A Tutorial on Spectral Clustering*, Max Planck Institute for Biological Cybernetics
- Aarti Singh, *Spectral Clustering*.
- Andrew Y. Ng, Michael I. Jordan, Yair Weiss, *On Spectral Clustering: Analysis and an Algorithm.*
- Jing Wang, *An Introduction to Support Vector Machine and Spectral Clustering.*
- Lihi Zelnik-Manor, Pietro Perona, *Self-Tuning Spectral Clustering.*
- Denis Hamad, Philippe Biela, *Introduction to spectral clustering.*
- Marina Meila, *Classic and Modern Data Clustering*
- H.T. Kung, Dario Vlah, *A Spectral Clustering Approach to Validating Sensors via Their Peers in Distributed Sensor Networks*

References

Spectral Clustering

- Morteza Chehreghani, Alberto Busetto, Joachim Buhmann, *Information Theoretic Model Validation for Spectral Clustering*
- Sepandar D. Kamvar, Dan Klein, Christopher Manning, *Spectral Clustering*

Clustering Validation

- Vendramin, L. & J. G. B. Campello, R. & Hruschka, E. (2010). Relative clustering validity criteria: A comparative overview. Statistical Analysis and Data Mining. 3. 209-235. 10.1002/sam.10080.
- Amigó, E. & Gonzalo, J. & Artiles, J. & Verdejo, M. (2009). Comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval. 12. 461-486.
- M Lewis, J. & Ackerman, M. & de Sa, V. (2012). Human Cluster Evaluation and Formal Quality Measures: A Comparative Study. Proc. 34th Conf. of the Cognitive Science Society.
- Bernard Desgraupes (2013). Clustering Indices. Lab Modal'X, University Paris Ouest.

References

Spectral Clustering Case Studies

- Justin Cranshaw, Raz Schwartz, Jason I. Hong, and Norman Sadeh, “The Livehoods Project: Utilizing Social Media to Understand the Dynamics of A City,” *Proceedings of the the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM–12),* Dublin, Ireland, pp. 1–8, 2012.

- Kung H. T., Vlah D.A, “Spectral clustering approach to validating sensors via their peers in distributed sensor networks”, *Proceedings of the 18th IEEE International Conference on Computer Communications and Networks (ICCCN '09),* 2009.

- F. R. Bach and M. I. Jordan, “Spectral clustering for speech separation”, in J. Keshet and S. Bengio (Eds.), *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods.* New York: John Wiley, 2008.