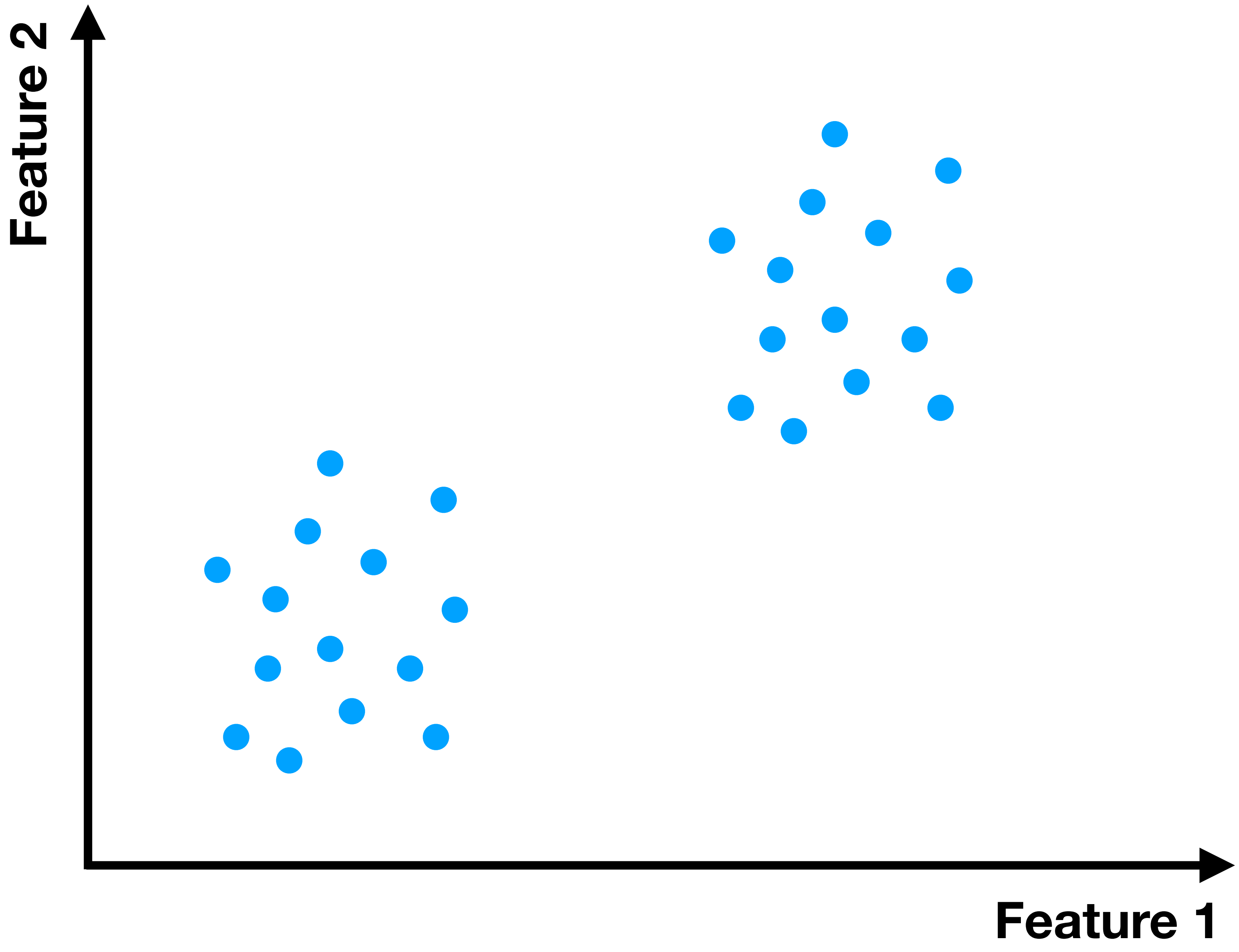
Clustering Algorithms

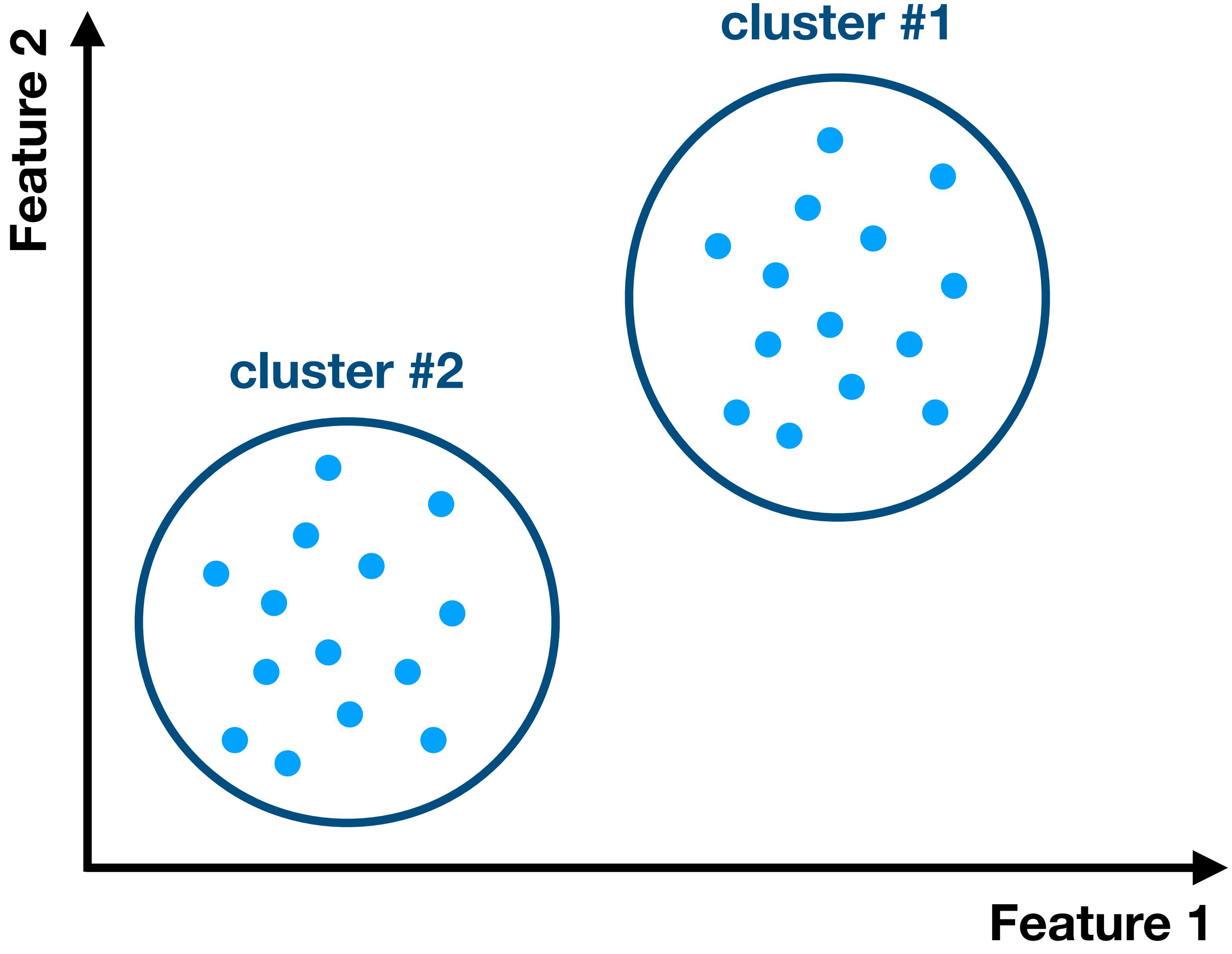
Dalya Baron (Tel Aviv University)

XXX Winter School, November 2018

Clustering

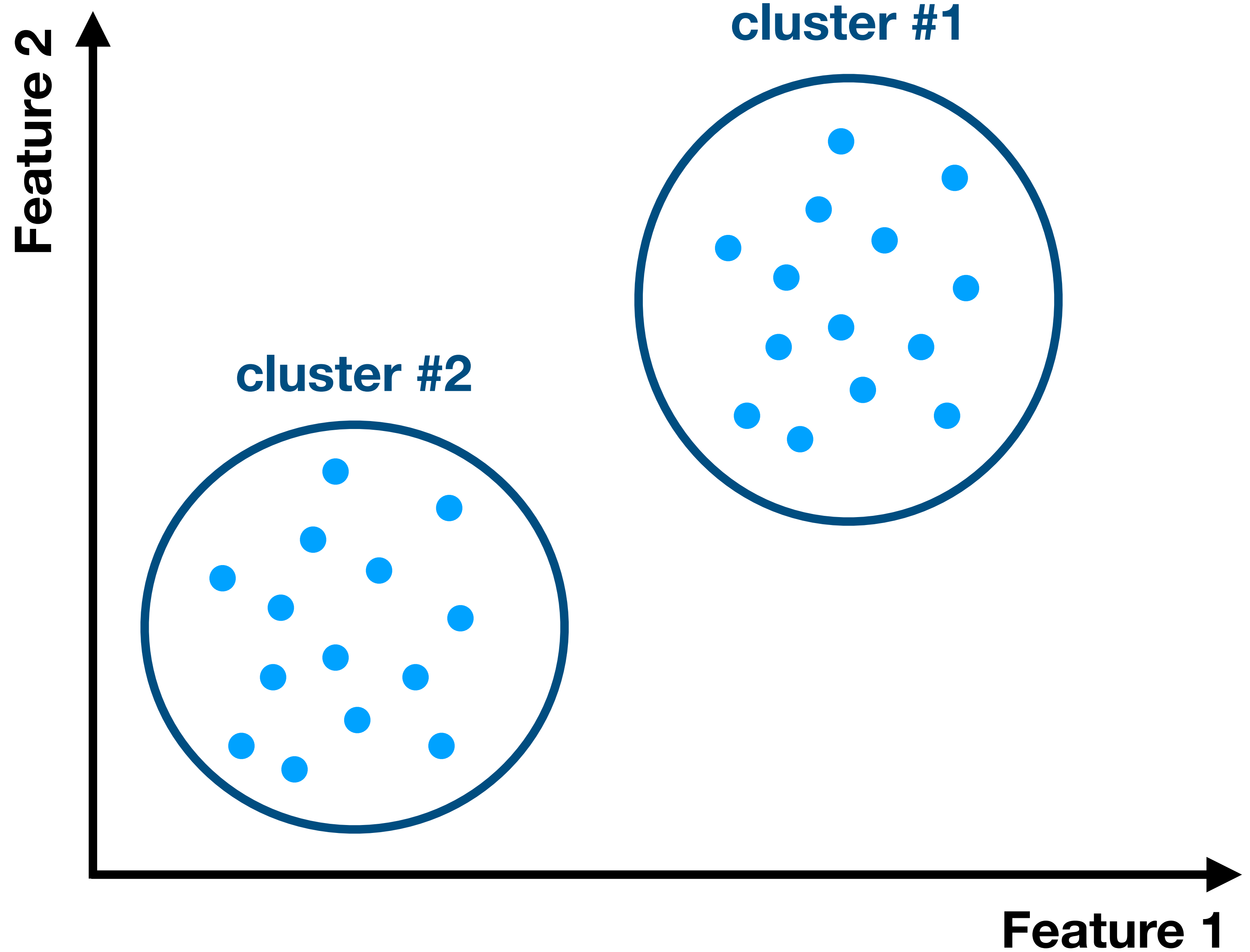


Clustering

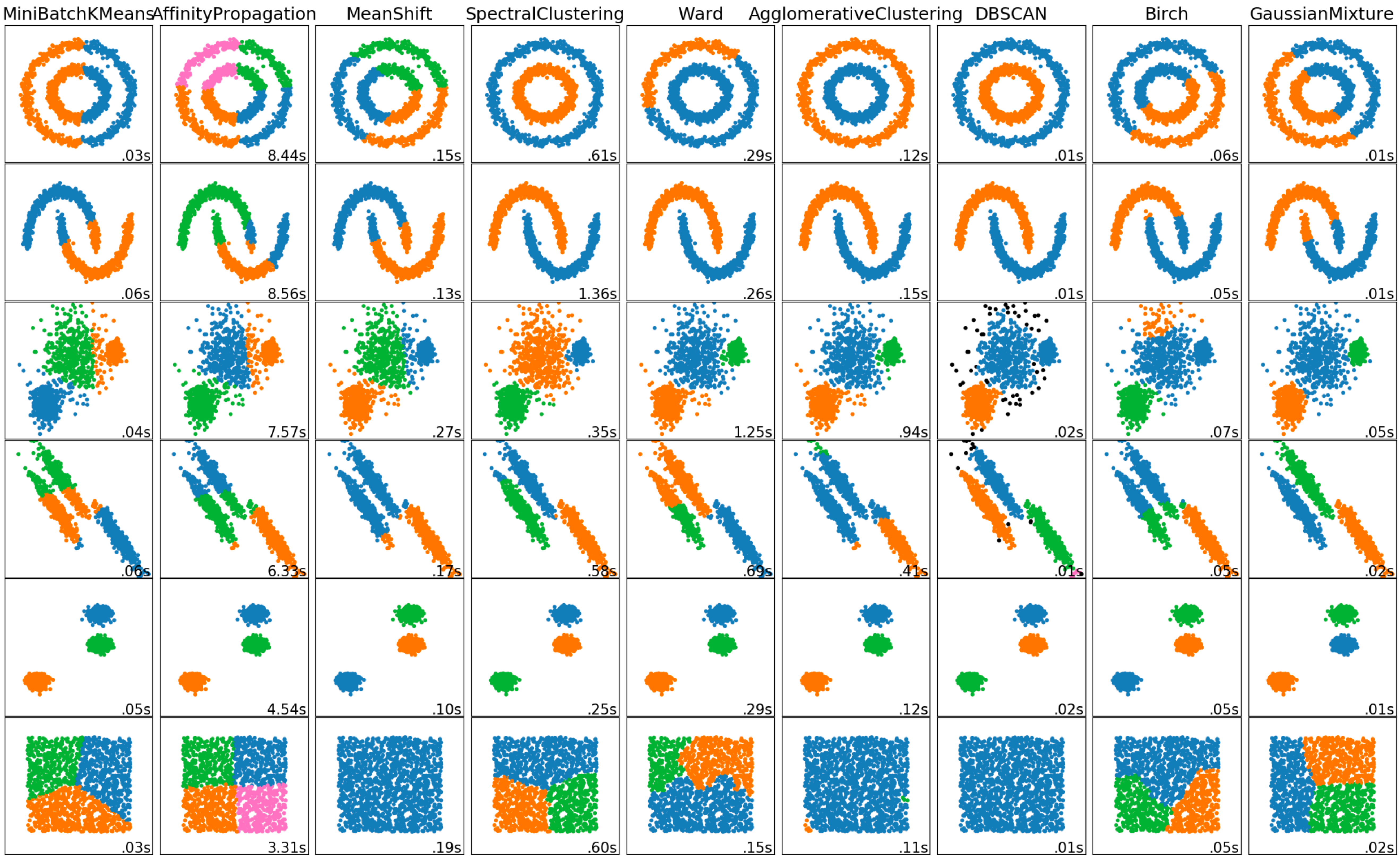


Clustering

Why should we look for clusters?

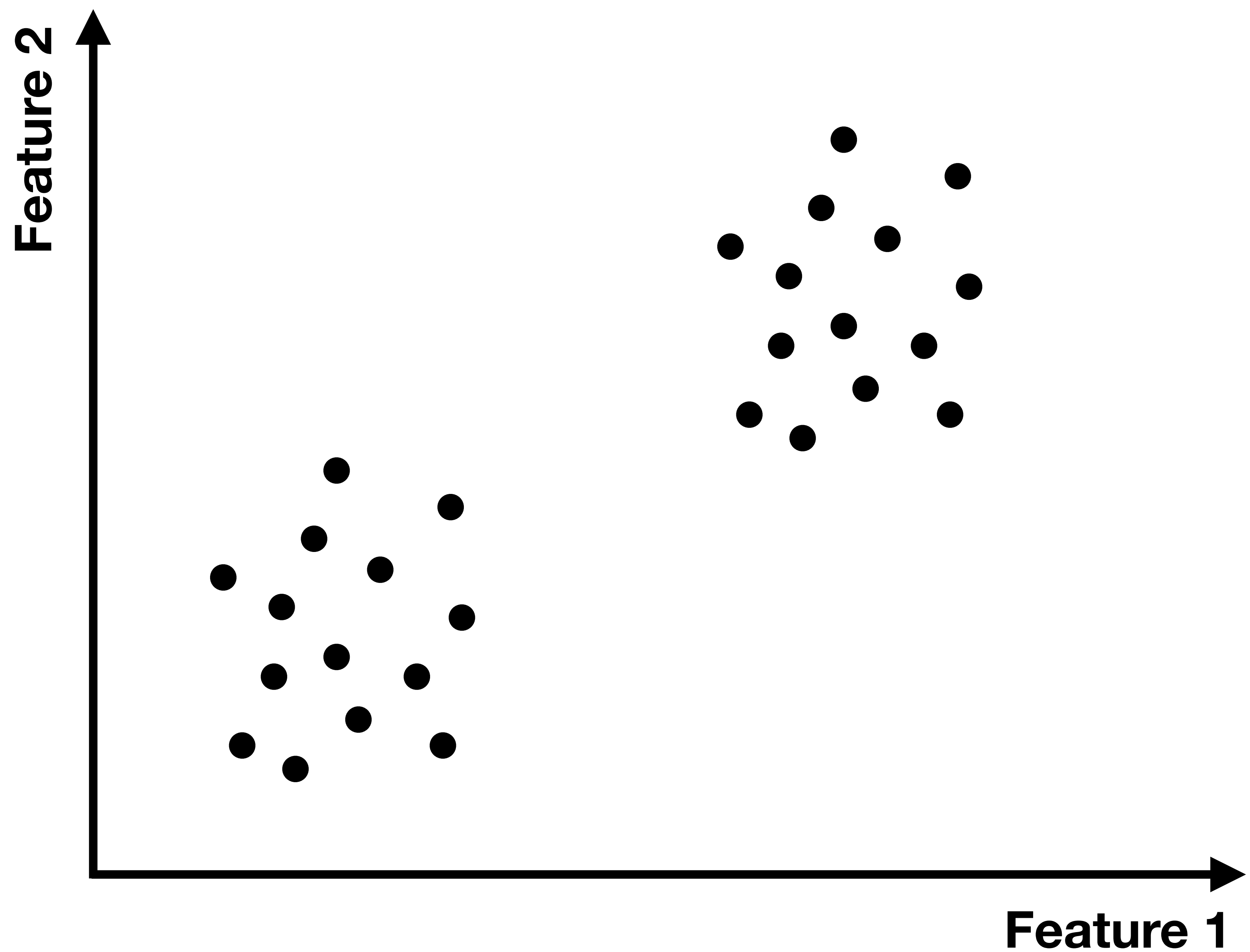


Clustering


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture
.03s
8.44s
.15s
.61s
.29s
.12s
.01s
.06s
.01s
.06s
8.56s
.13s
1.36s
.26s
.15s
.01s
.05s
.01s
.04s
7.57s
.27s
.35s
1.25s
.94s
.02s
.07s
.05s
.06s
6.33s
.17s
.58s
.69s
.41s
.01s
.05s
.02s
.05s
4.54s
.10s
.25s
.29s
.12s
.02s
.05s
.01s
.03s
3.31s
.19s
.60s
.15s
.11s
.01s
.05s
.02s

K-means

Input: measured features, and the number of clusters, **k**. The algorithm will classify **all** the objects in the sample into k clusters.



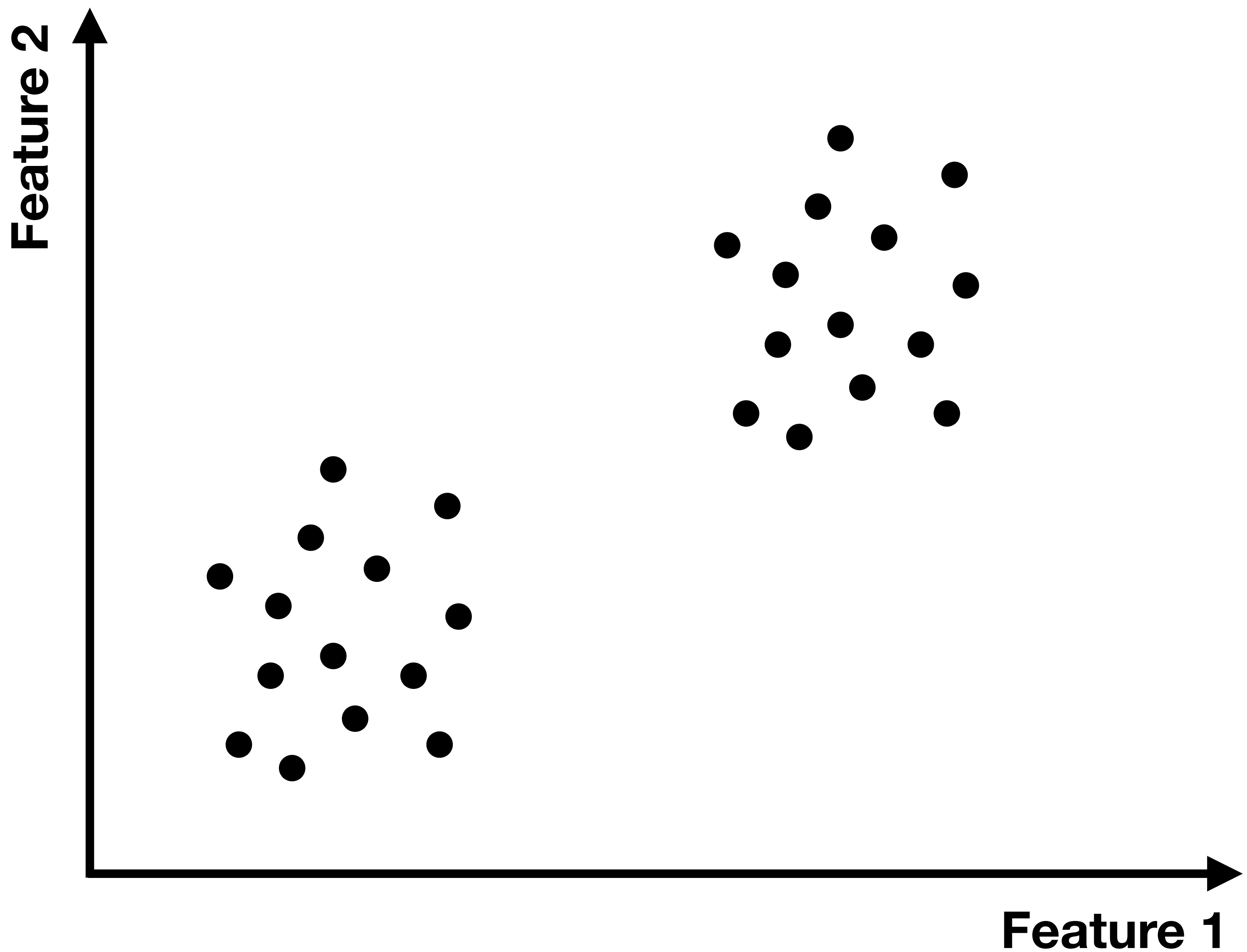
K-means

(I) The algorithm places randomly **k** points that represent the centroids of the clusters.

The algorithm performs several iterations, in each of them:

(II) The algorithm associates each object with a single cluster, according to its **distance** from the cluster centroid.

(III) The algorithm recalculates the cluster centroid according to the objects that are associated with it.



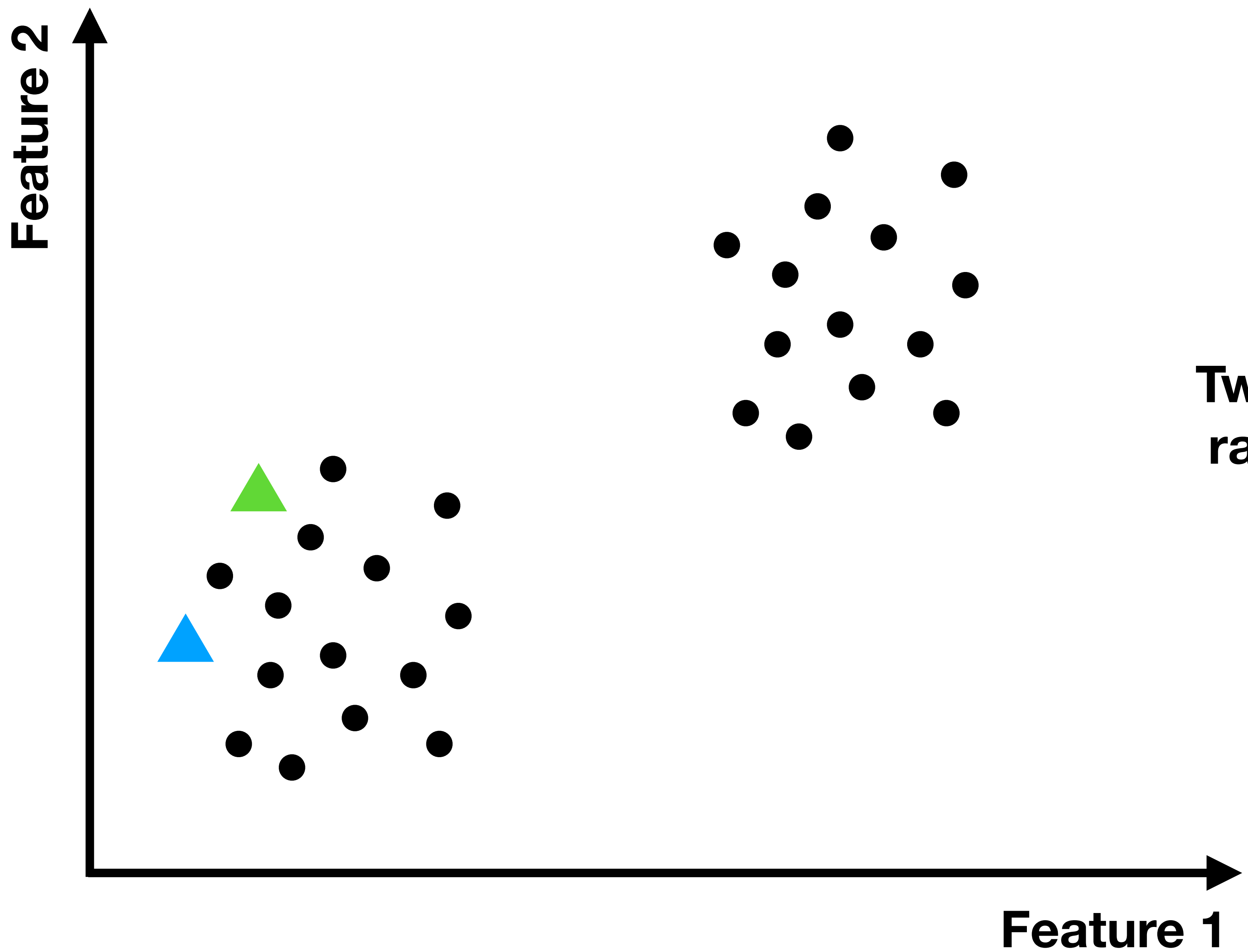
K-means

(I) The algorithm places randomly **k** points that represent the centroids of the clusters.

The algorithm performs several iterations, in each of them:

(II) The algorithm associates each object with a single cluster, according to its **distance** from the cluster centroid.

(III) The algorithm recalculates the cluster centroid according to the objects that are associated with it.



Two centroids are randomly placed

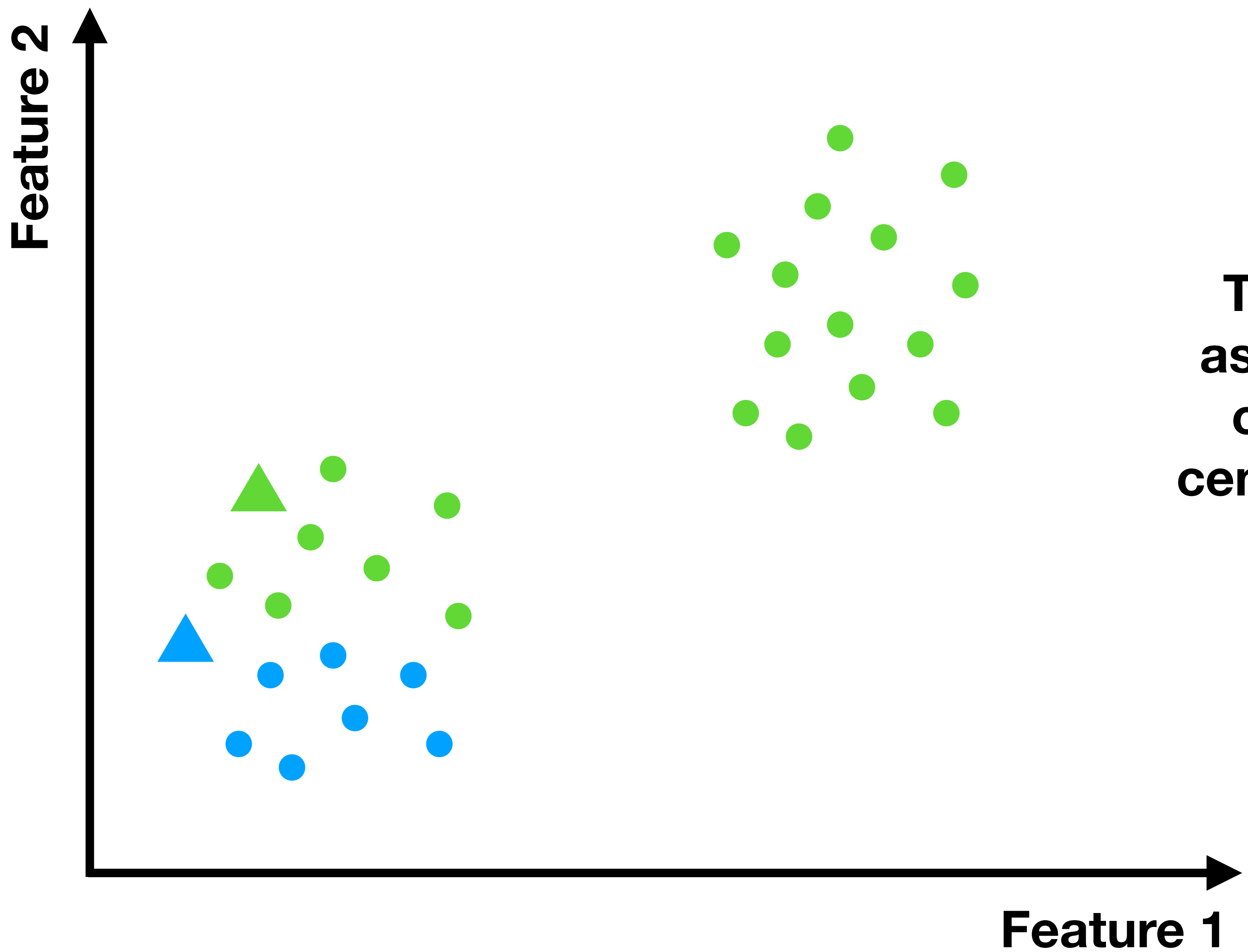
K-means

(I) The algorithm places randomly **k** points that represent the centroids of the clusters.

The algorithm performs several iterations, in each of them:

(II) The algorithm associates each object with a single cluster, according to its **distance** from the cluster centroid.

(III) The algorithm recalculates the cluster centroid according to the objects that are associated with it.



The objects are associated to the closest cluster centroid (Euclidean distance).

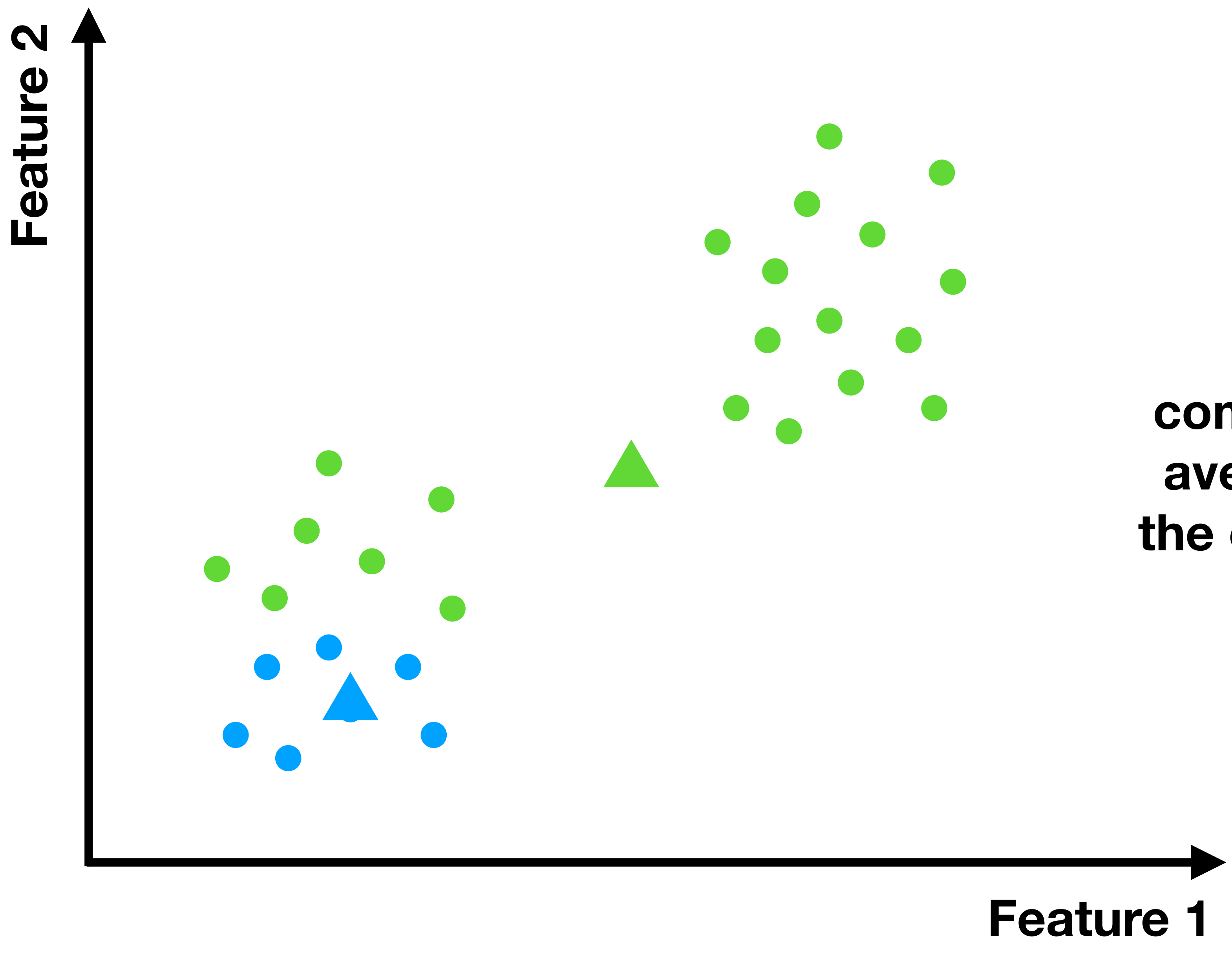
K-means

(I) The algorithm places randomly **k** points that represent the centroids of the clusters.

The algorithm performs several iterations, in each of them:

(II) The algorithm associates each object with a single cluster, according to its **distance** from the cluster centroid.

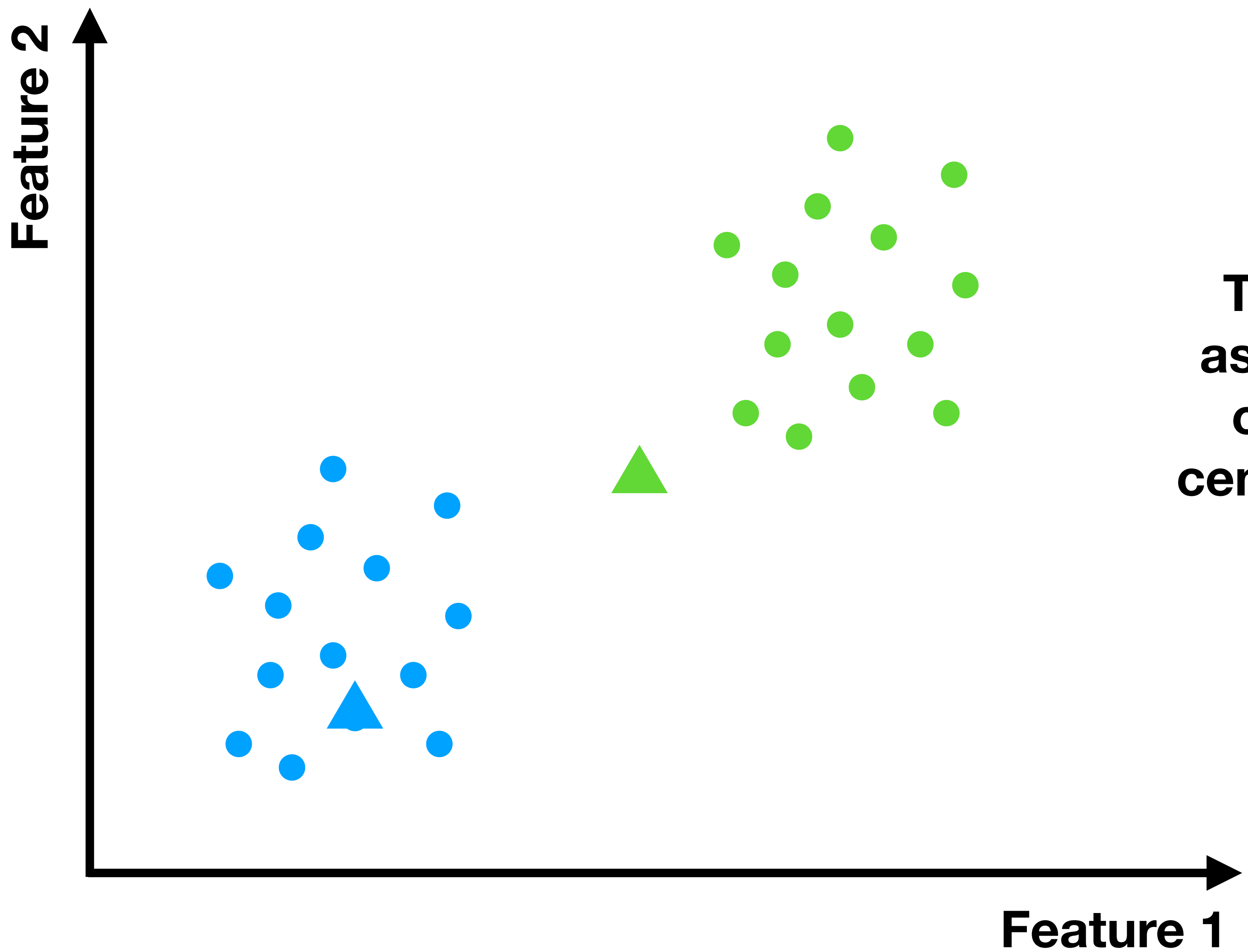
(III) The algorithm recalculates the cluster centroid according to the objects that are associated with it.



New cluster centroids are computed using the average location of the cluster members.

K-means

(I) The algorithm places randomly **k** points that represent the centroids of the clusters.
The algorithm performs several iterations, in each of them:
(II) The algorithm associates each object with a single cluster, according to its **distance** from the cluster centroid.
(III) The algorithm recalculates the cluster centroid according to the objects that are associated with it.



The objects are associated to the closest cluster centroid (Euclidean distance).

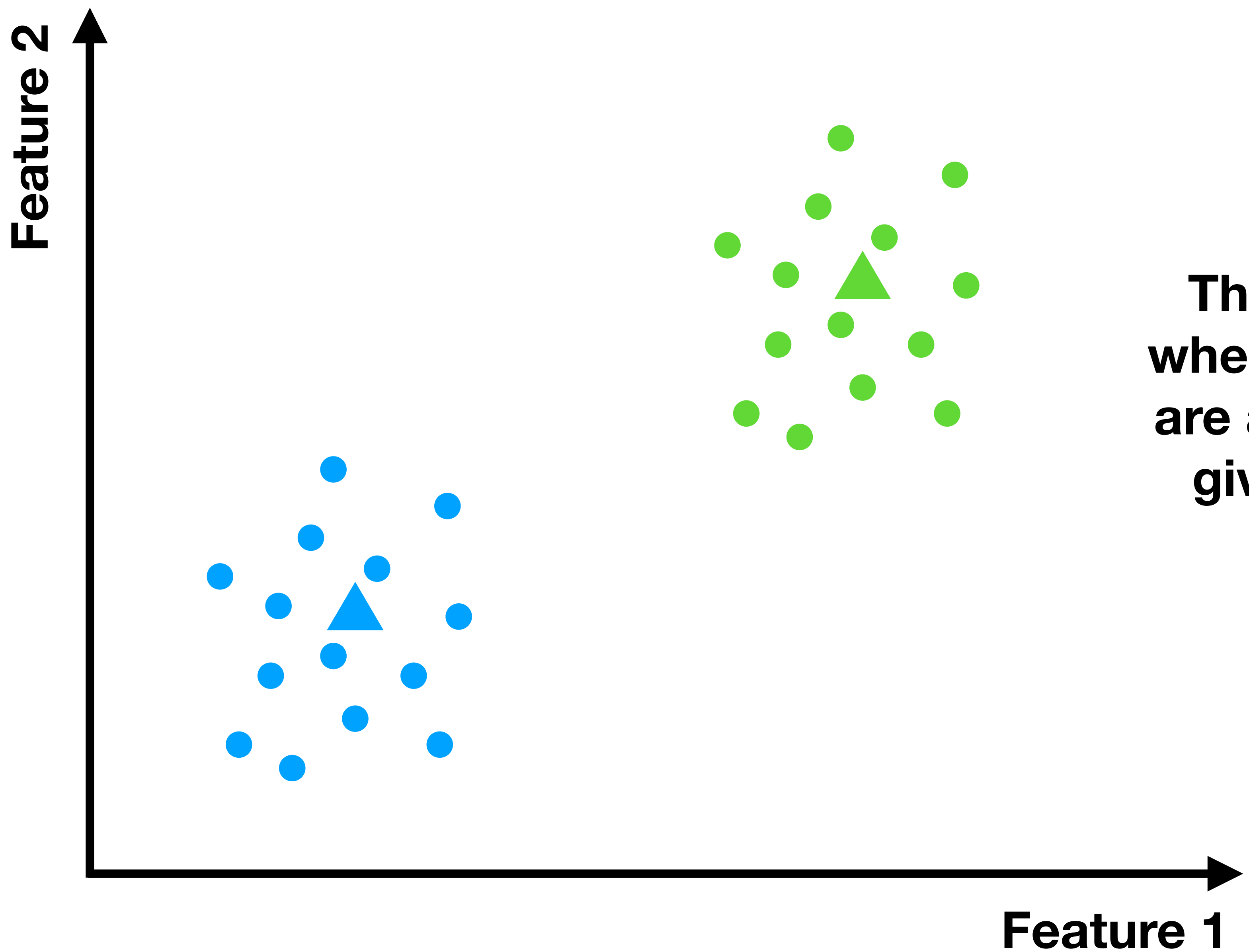
K-means

(I) The algorithm places randomly **k** points that represent the centroids of the clusters.

The algorithm performs several iterations, in each of them:

(II) The algorithm associates each object with a single cluster, according to its **distance** from the cluster centroid.

(III) The algorithm recalculates the cluster centroid according to the objects that are associated with it.



The process stops when the objects that are associated with a given class do not change.

The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, \ldots\}) = \overrightarrow{y}$$

Internal choices and/or internal cost function:

(I) Initial centroids are randomly selected from the set of examples.

(II) The global cost function that is minimized by K-means:

$$J = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2$$

cluster centroids

cluster members

Euclidean distance

The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Internal choices and/or internal cost function:

(I) Initial centroids are randomly selected from the set of examples.

(II) The global cost function that is minimized by K-means:

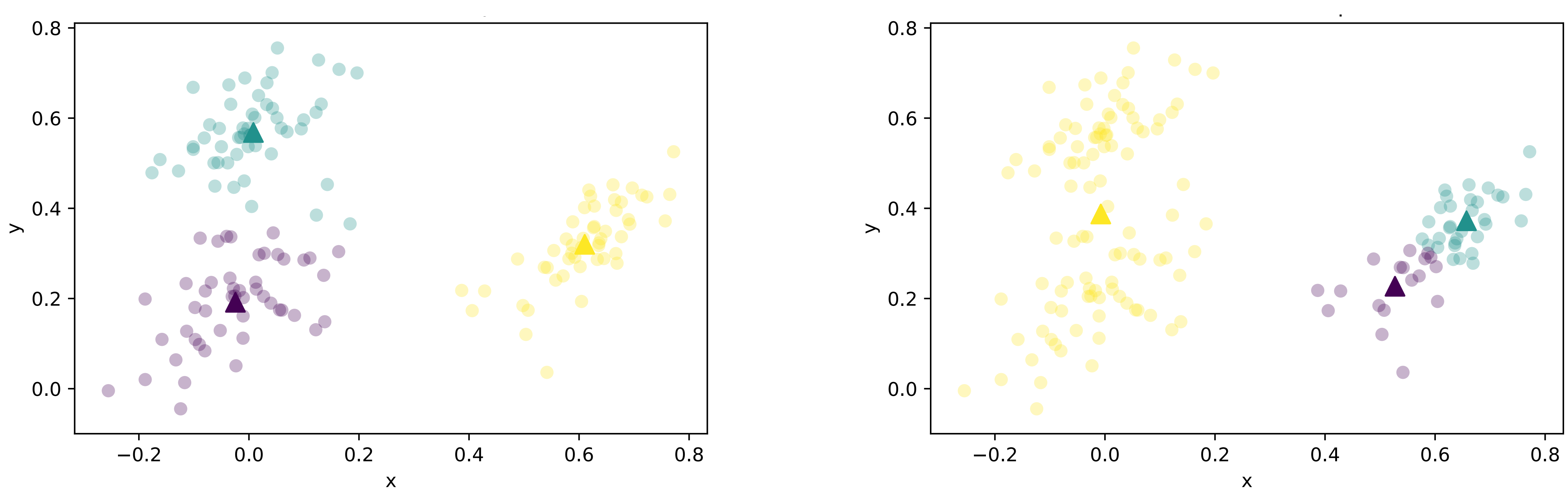
$$J = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2$$

cluster centroids (μ_k)

cluster members (x_i)

Euclidean distance ($||\cdot||^2$)

k=3, and two different random placements of centroids

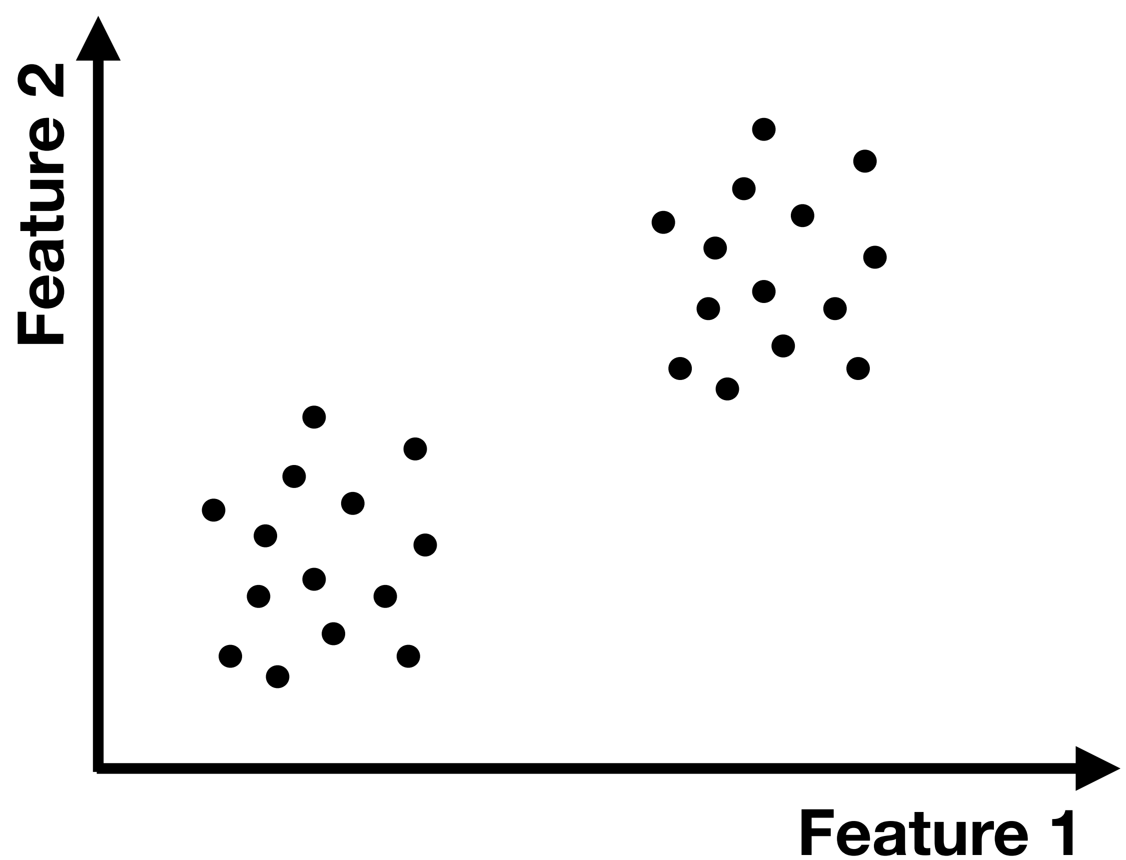


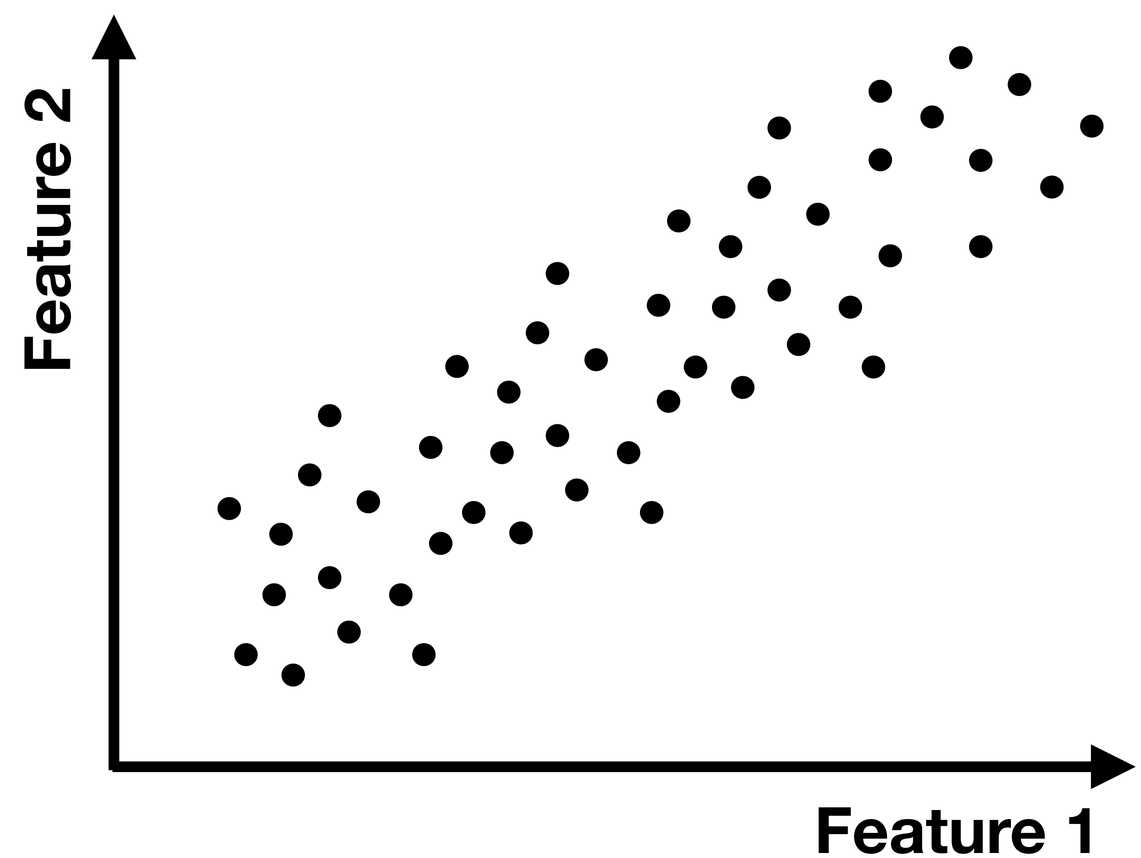
The anatomy of K-means

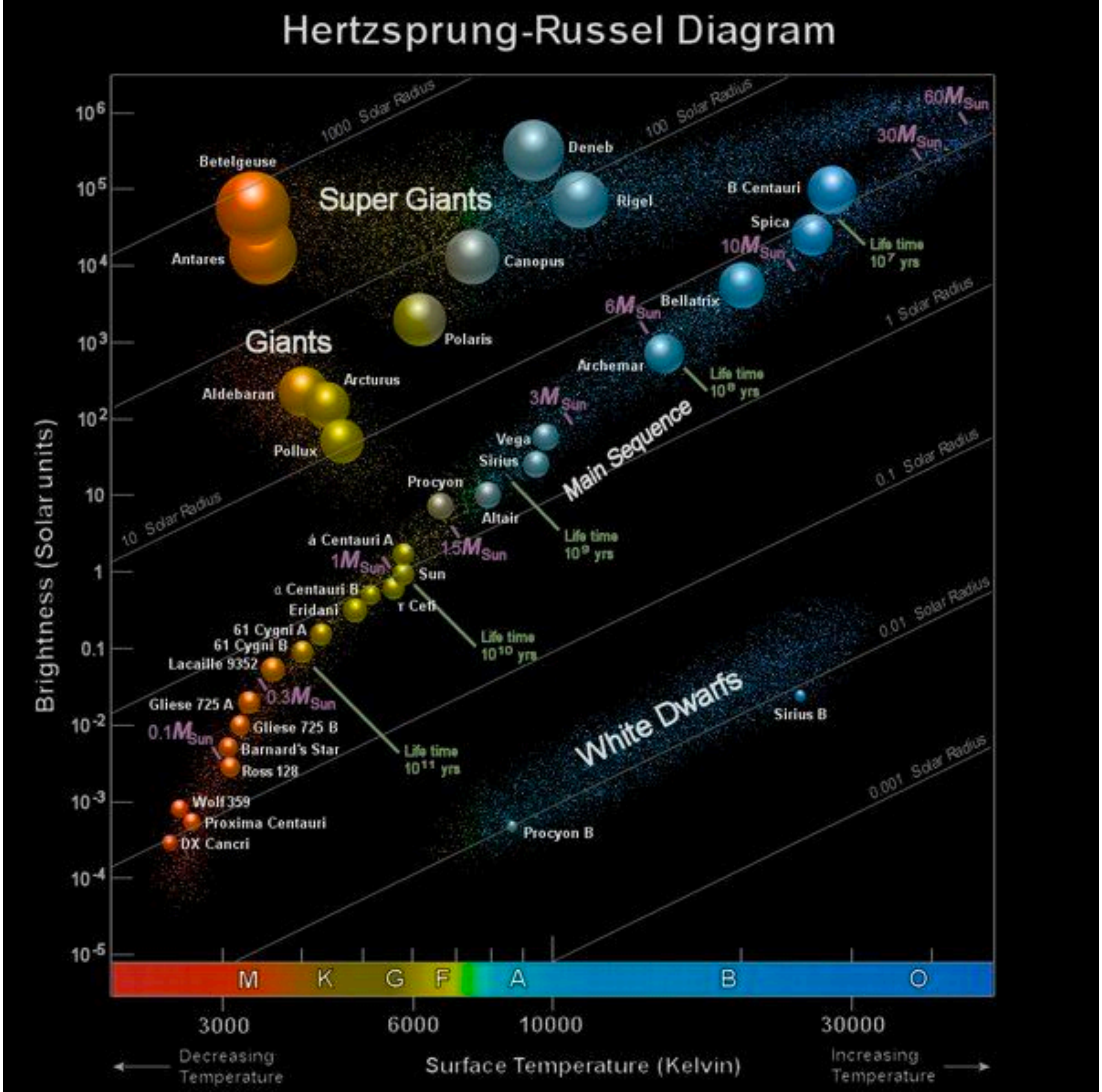
$$f(\overrightarrow{X}, \{a_1, a_2, \ldots\}) = \overrightarrow{y}$$

Input dataset: a list of objects with measured features.

For which datasets should we use K-means?







Hertzsprung-Russel Diagram
Brightness (Solar units)
Surface Temperature (Kelvin)
Decreasing Temperature
Increasing Temperature
3000
6000
10000
30000
M
K
G
F
A
B
O
1000 Solar Radius
100 Solar Radius
10 Solar Radius
1 Solar Radius
0.1 Solar Radius
0.01 Solar Radius
0.001 Solar Radius
Super Giants
Giants
Main Sequence
White Dwarfs
Betelgeuse
Antares
Deneb
Rigel
Canopus
Polaris
Arcturus
Aldebaran
Pollux
B Centauri
Spica
Bellatrix
Archemar
Vega
Sirius
Procyon
Altair
á Centauri A
Sun
α Centauri B
τ Ceti
Eridani
61 Cygni A
61 Cygni B
Lacaille 9352
Gliese 725 A
Gliese 725 B
Barnard's Star
Ross 128
Wolf 359
Proxima Centauri
DX Cancri
Sirius B
Procyon B
60M_{Sun}
30M_{Sun}
10M_{Sun}
6M_{Sun}
3M_{Sun}
1.5M_{Sun}
1M_{Sun}
0.3M_{Sun}
0.1M_{Sun}
Life time 10^7 yrs
Life time 10^8 yrs
Life time 10^9 yrs
Life time 10^{10} yrs
Life time 10^{11} yrs

The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

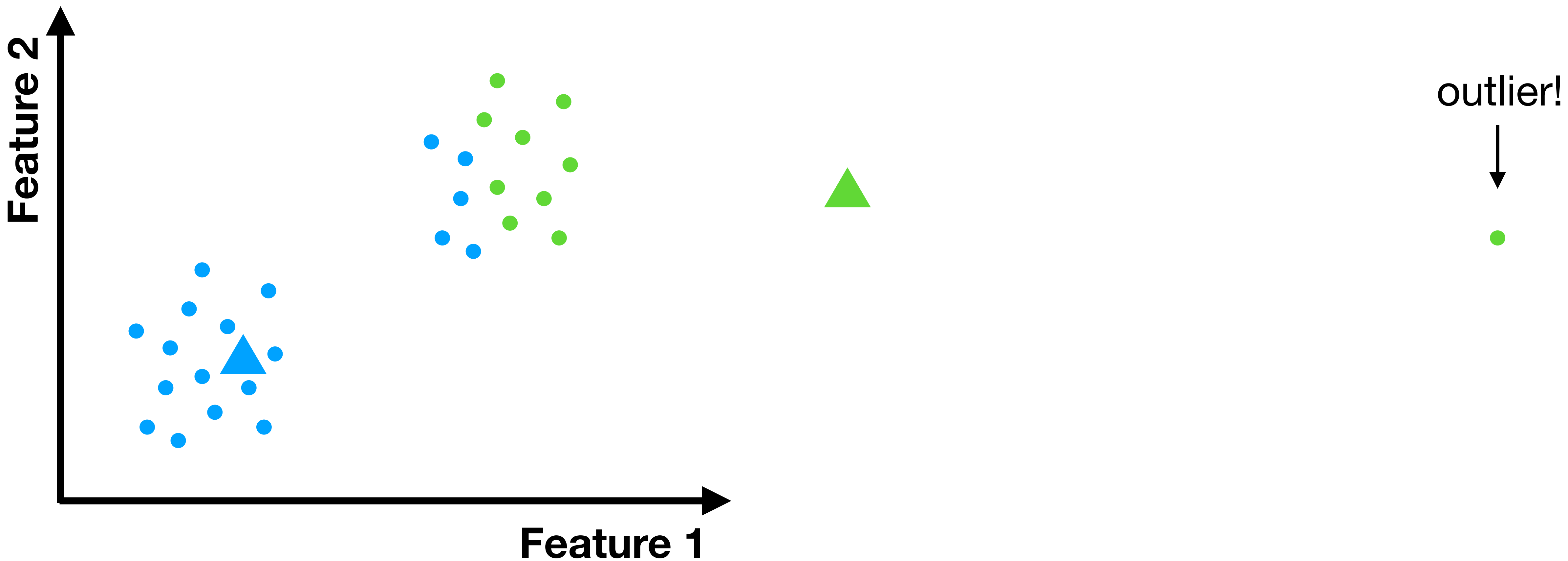
Input dataset: a list of objects with measured features.
What happens when we have an outlier in the dataset?

The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, \ldots\}) = \overrightarrow{y}$$

Input dataset: a list of objects with measured features.

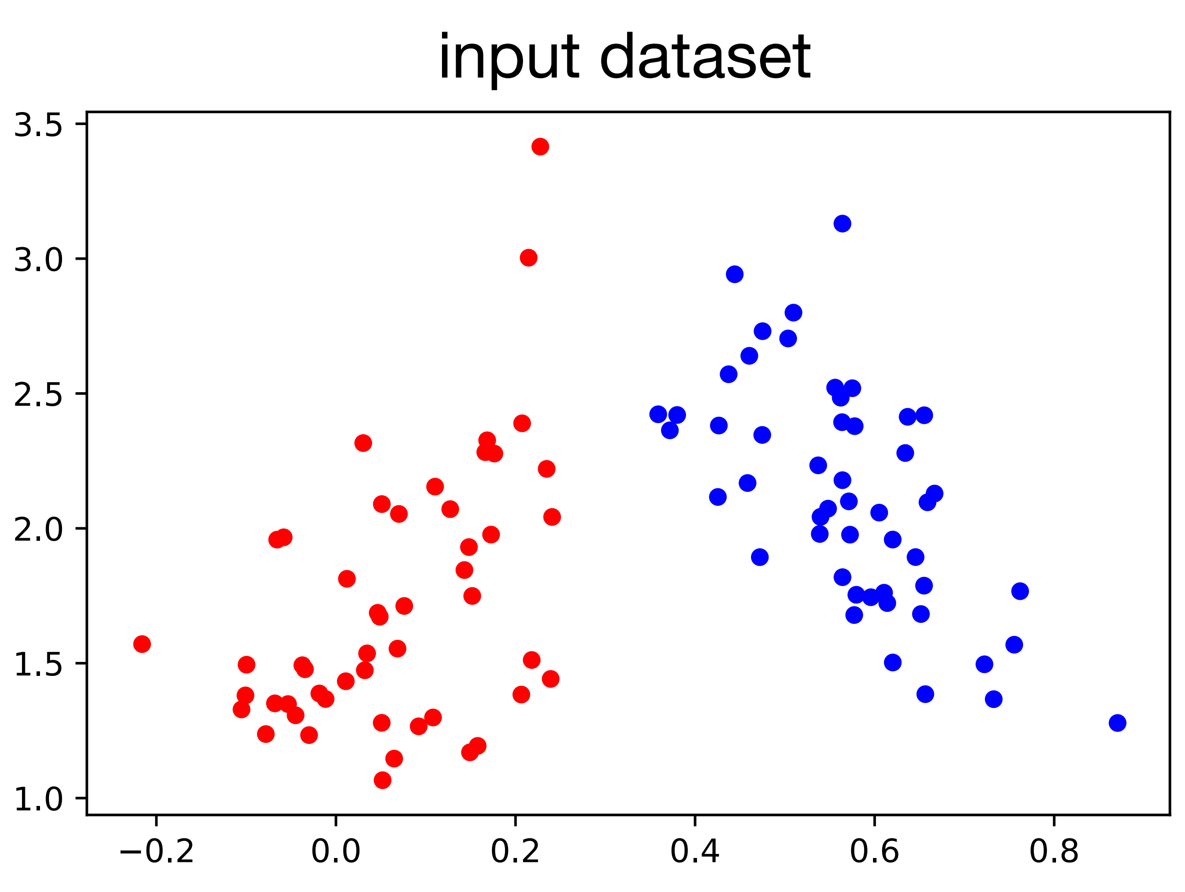
What happens when we have an outlier in the dataset?

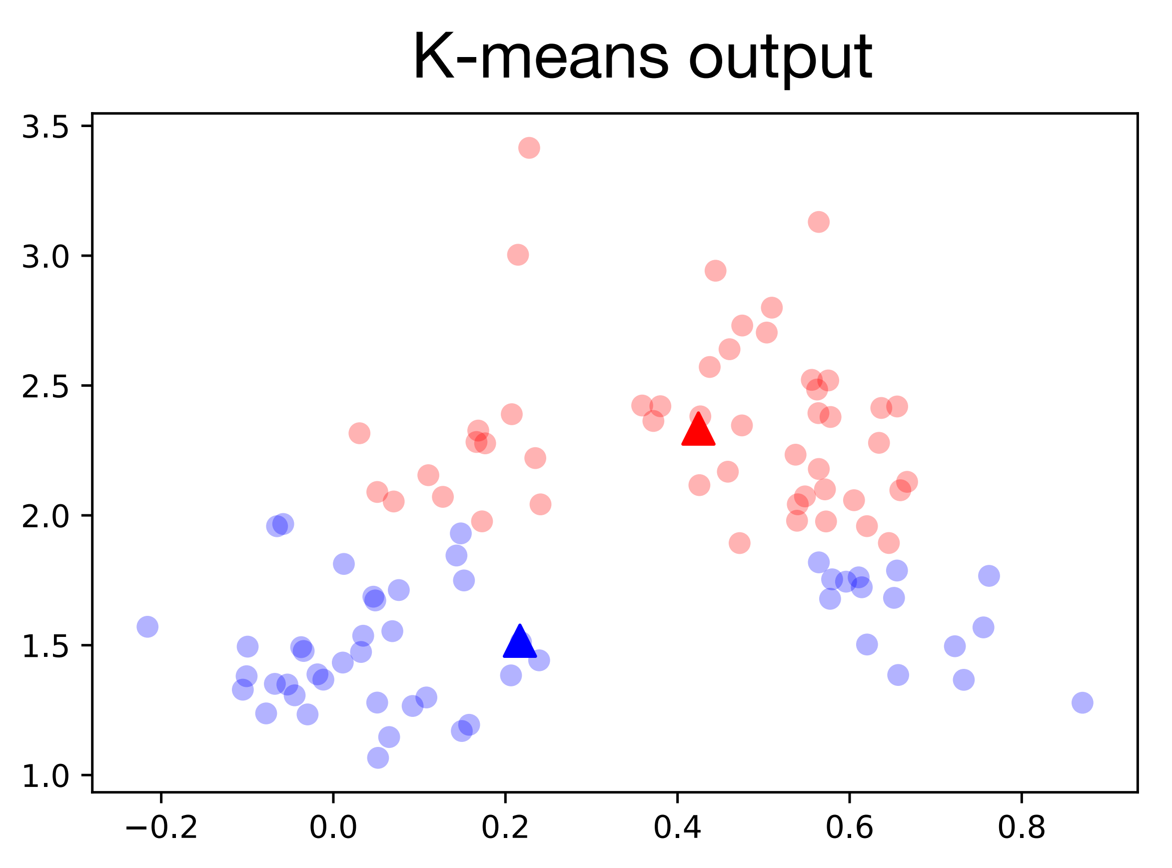


The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Input dataset: a list of objects with measured features.
What happens when the features have different physical units?



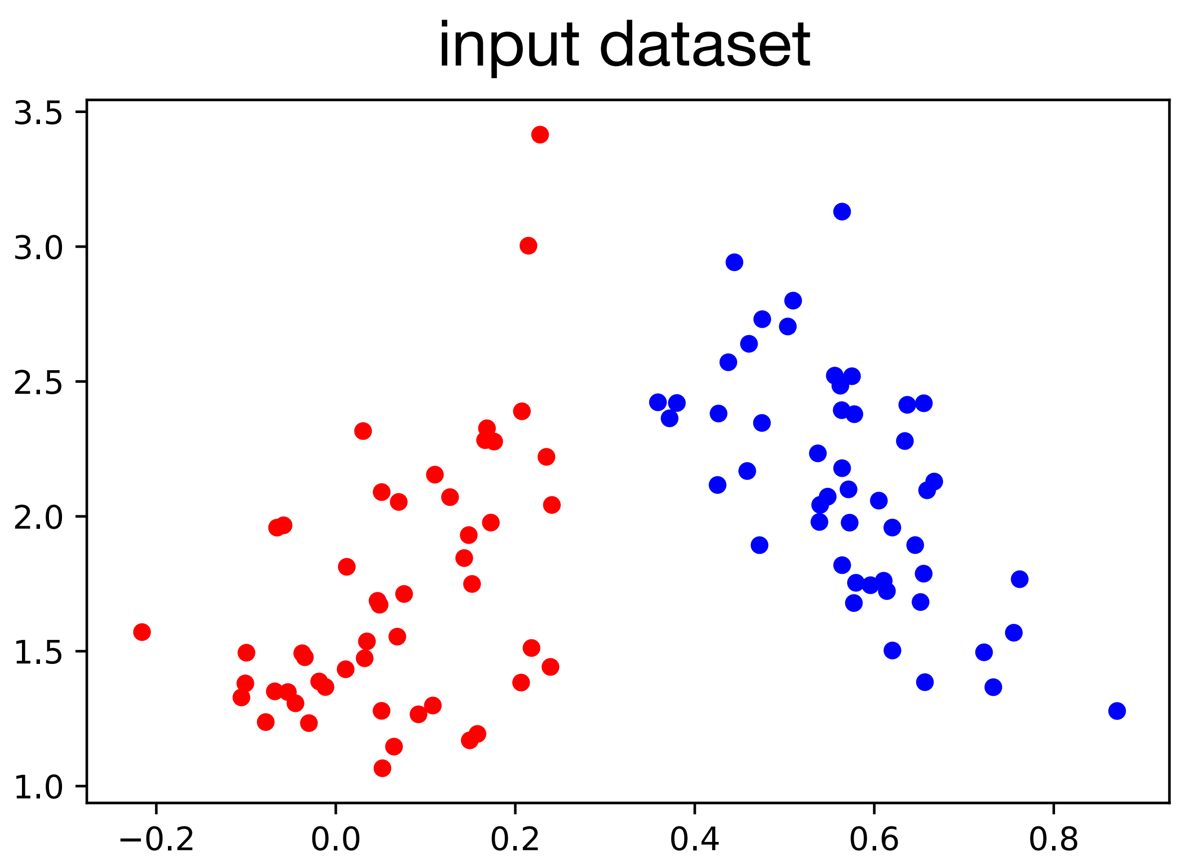


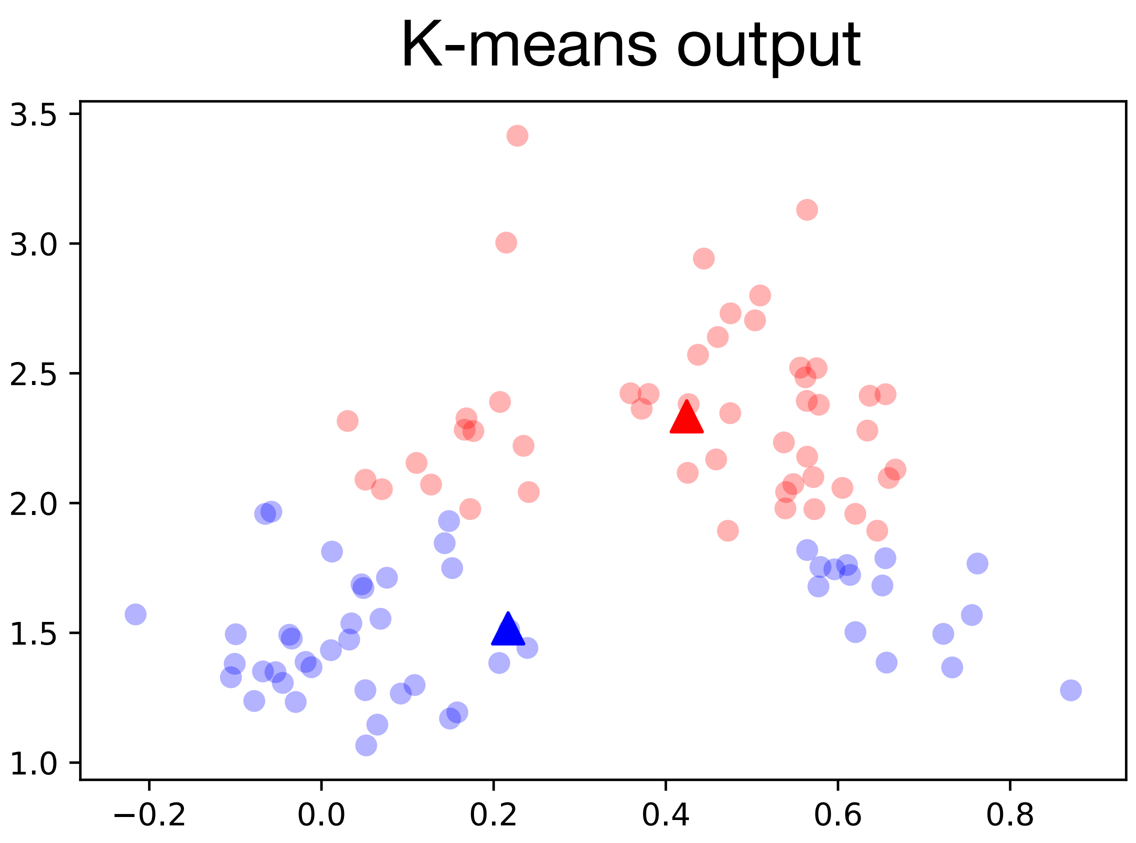
The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, \ldots\}) = \overrightarrow{y}$$

Input dataset: a list of objects with measured features.
What happens when the features have different physical units?

How can we avoid this?

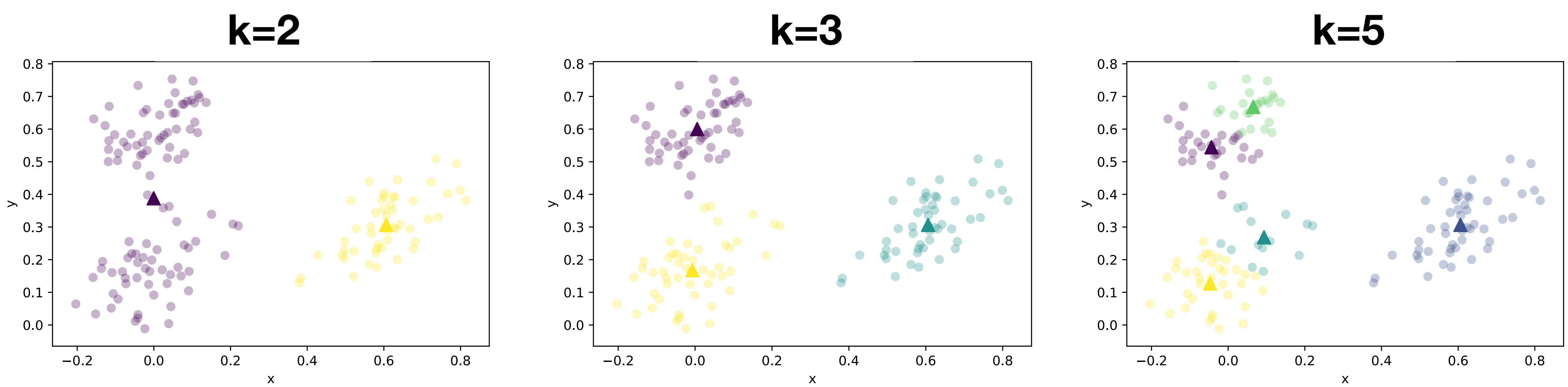




The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

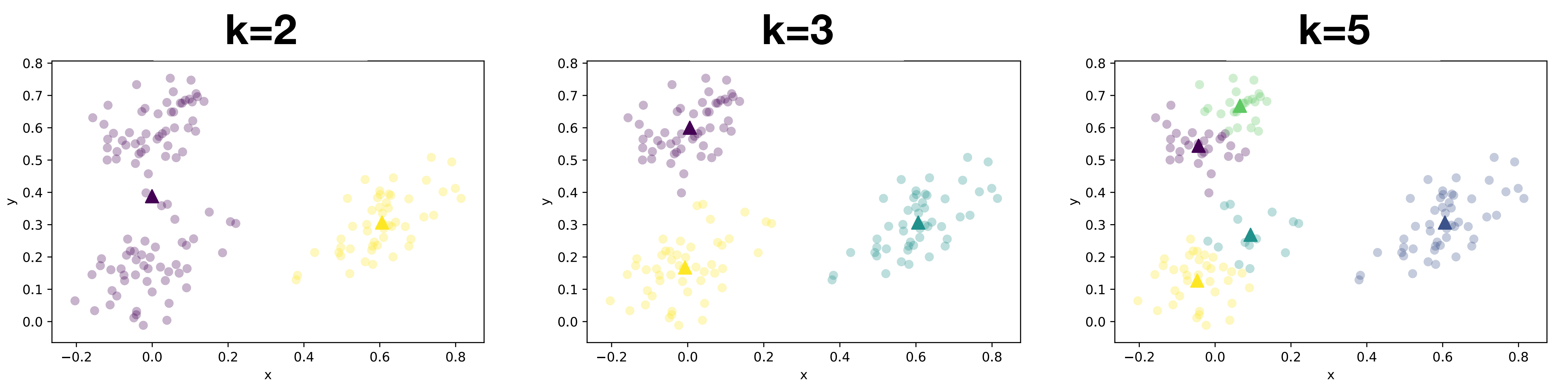
Hyper-parameters: the number of clusters, k.
Can we find the optimal k using the cost function?

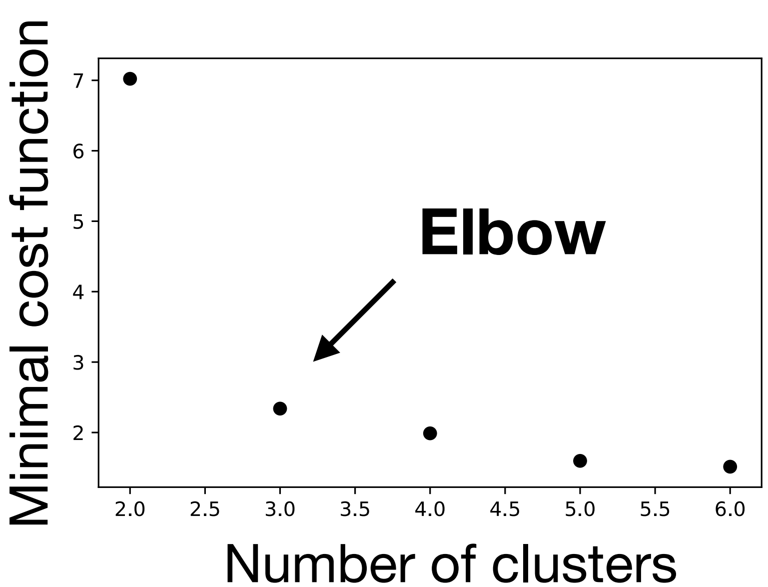


The anatomy of K-means

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Hyper-parameters: the number of clusters, k.
Can we find the optimal k using the cost function?

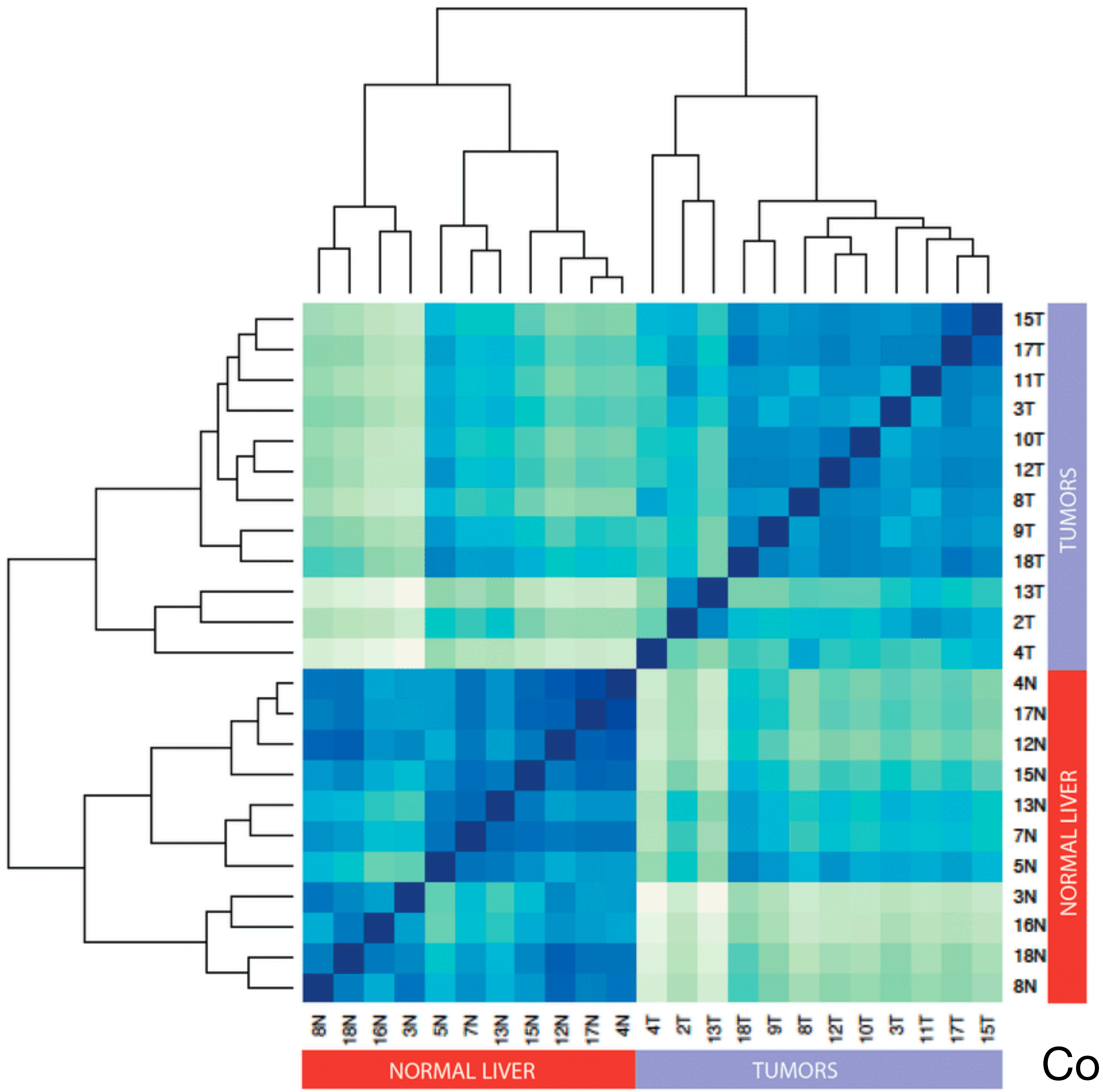




Questions?

Hierarchal Clustering

or, how to visualize complicated similarity measures

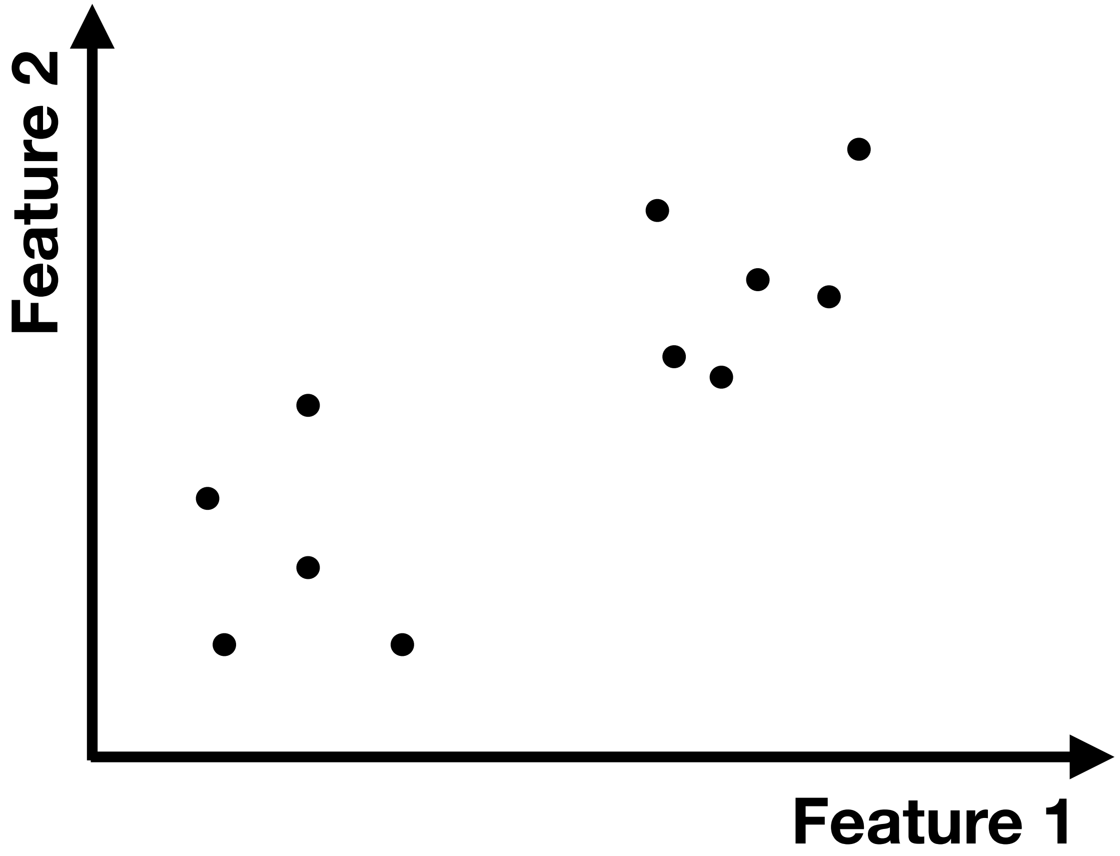


Correa-Gallego+ 2016

Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

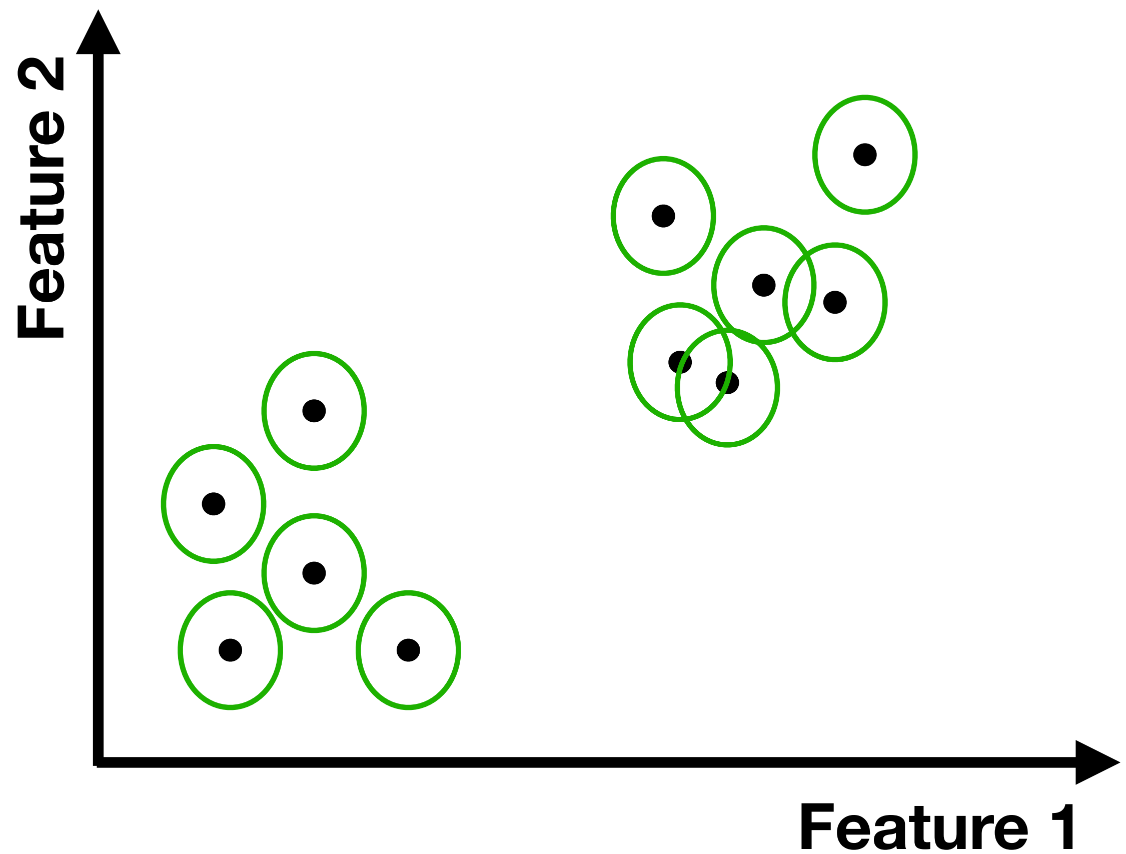
Initialization: each object is a cluster of size 1.



Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.



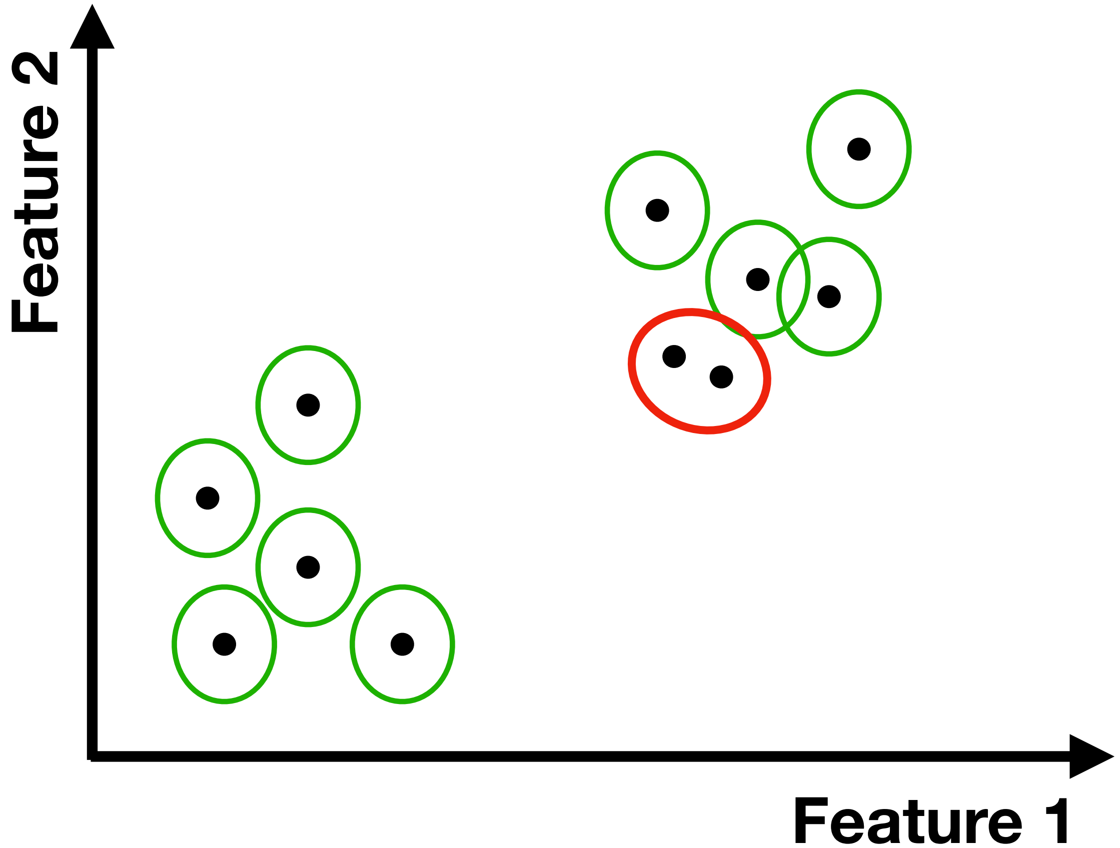
Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.

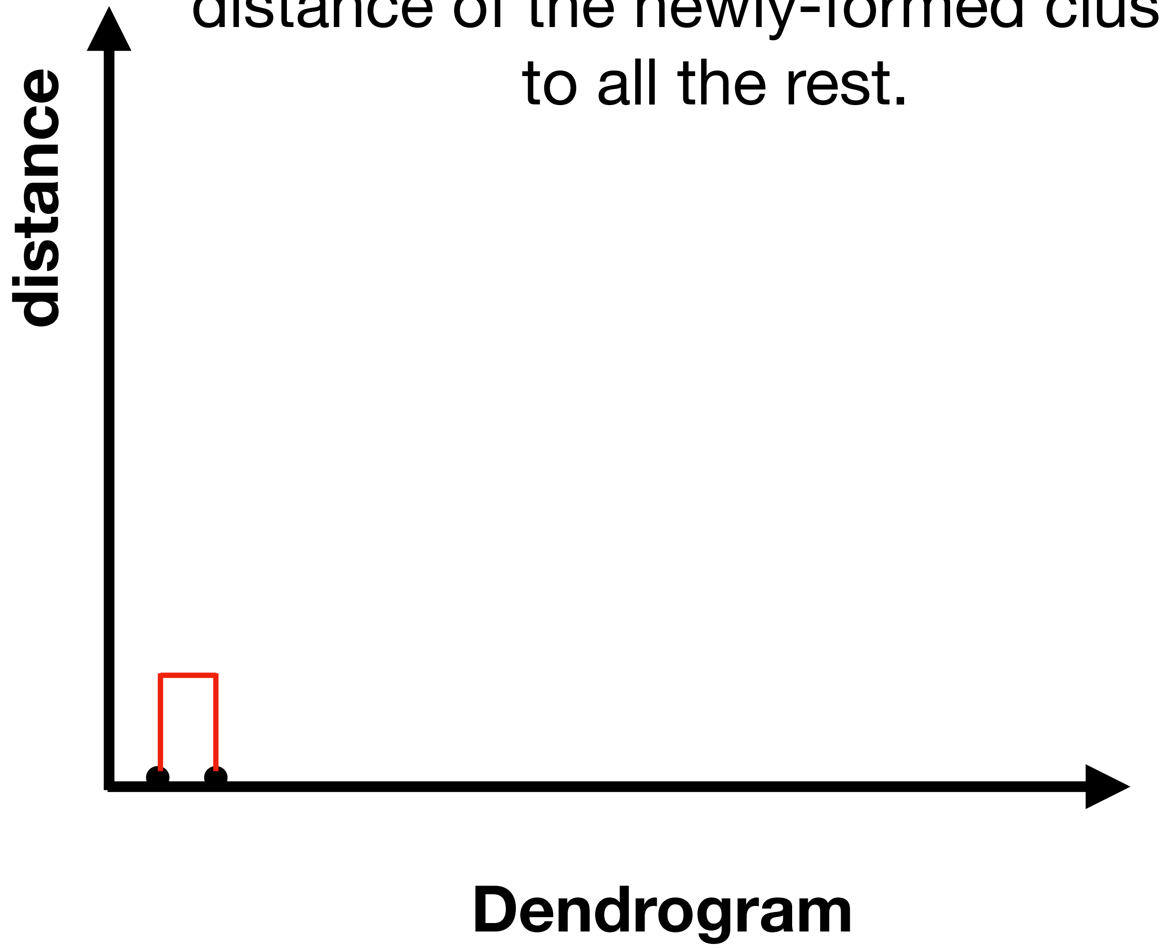
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



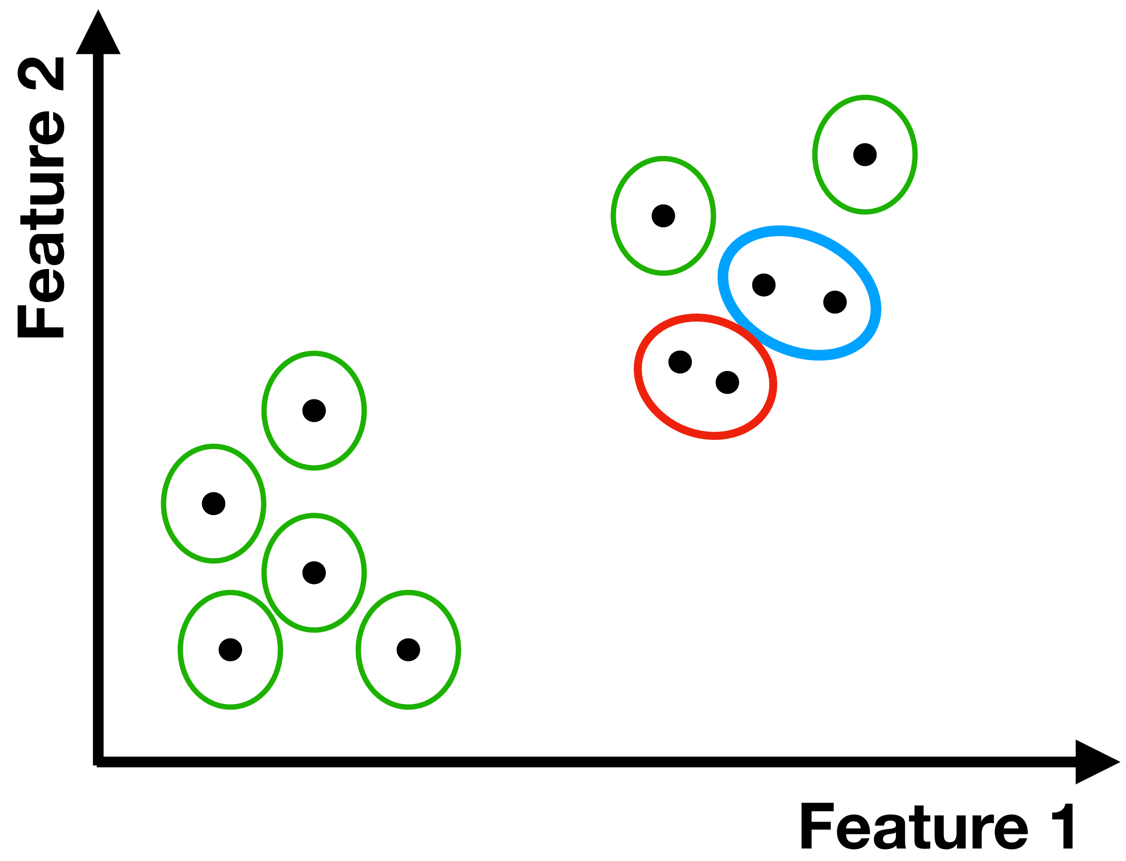


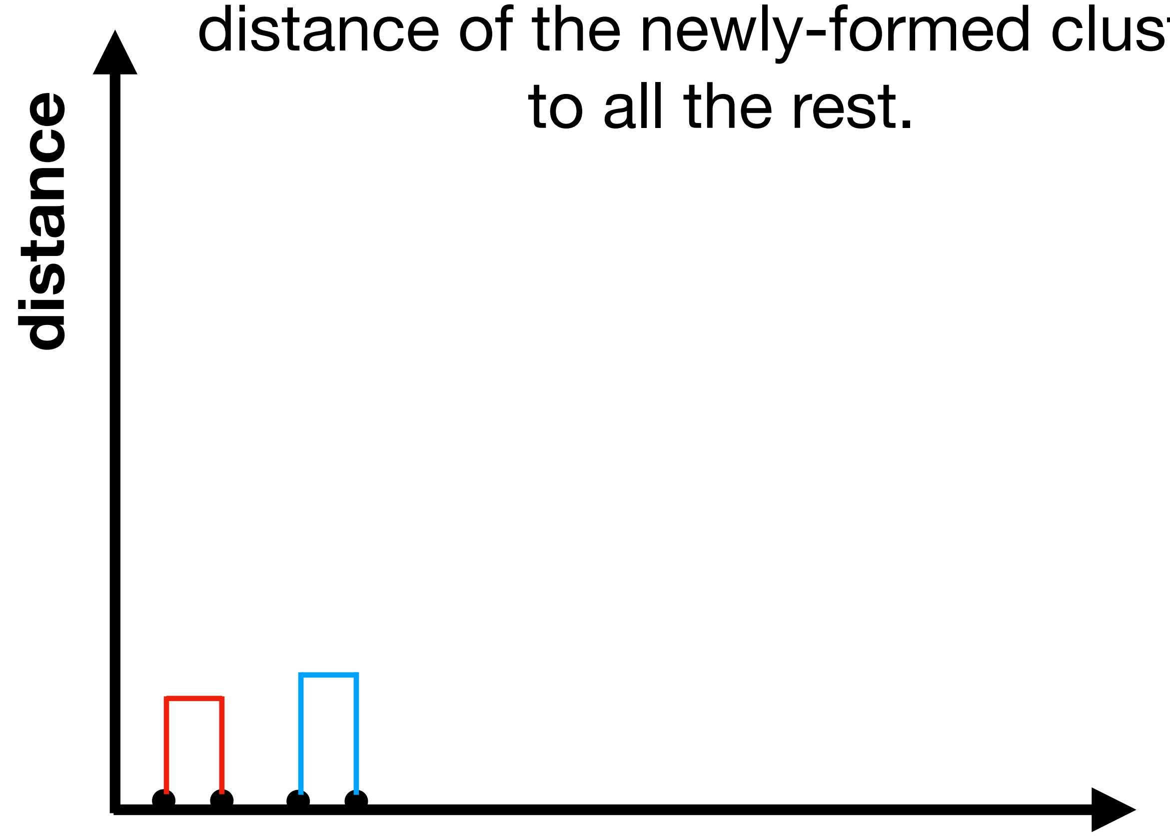
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.





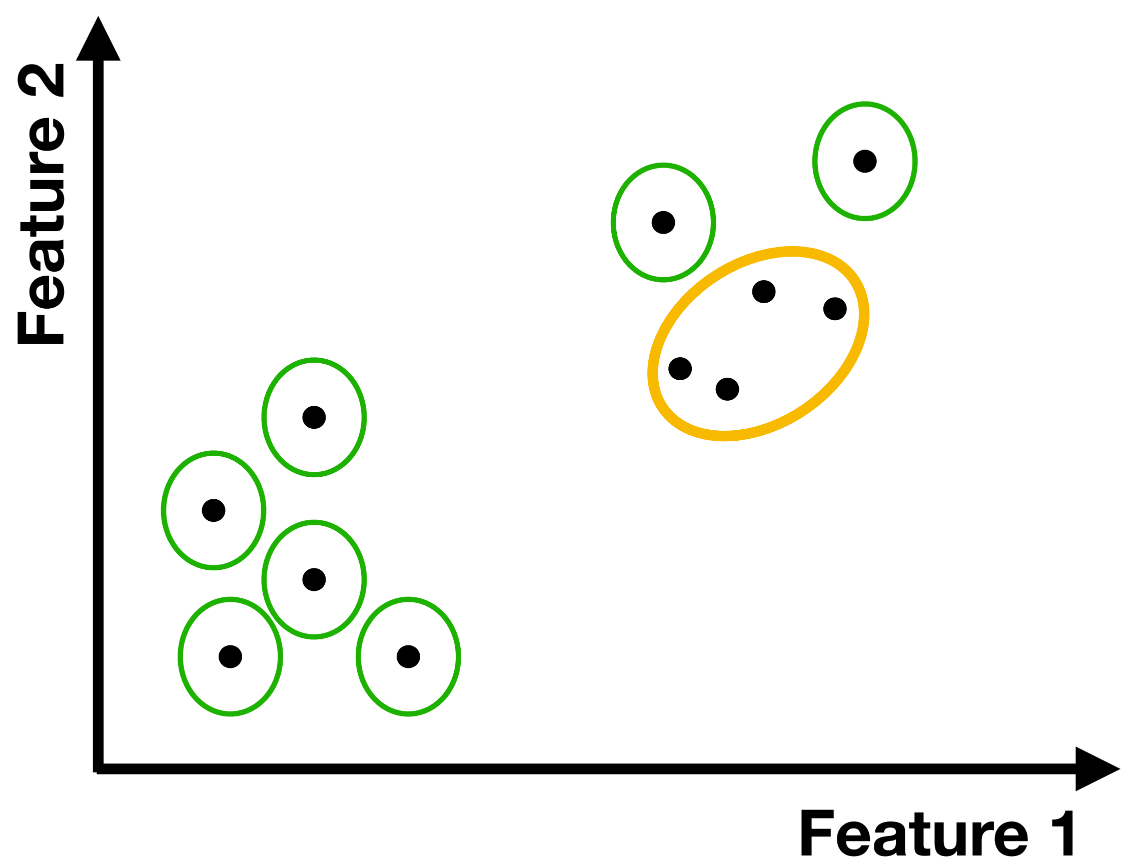
Dendrogram

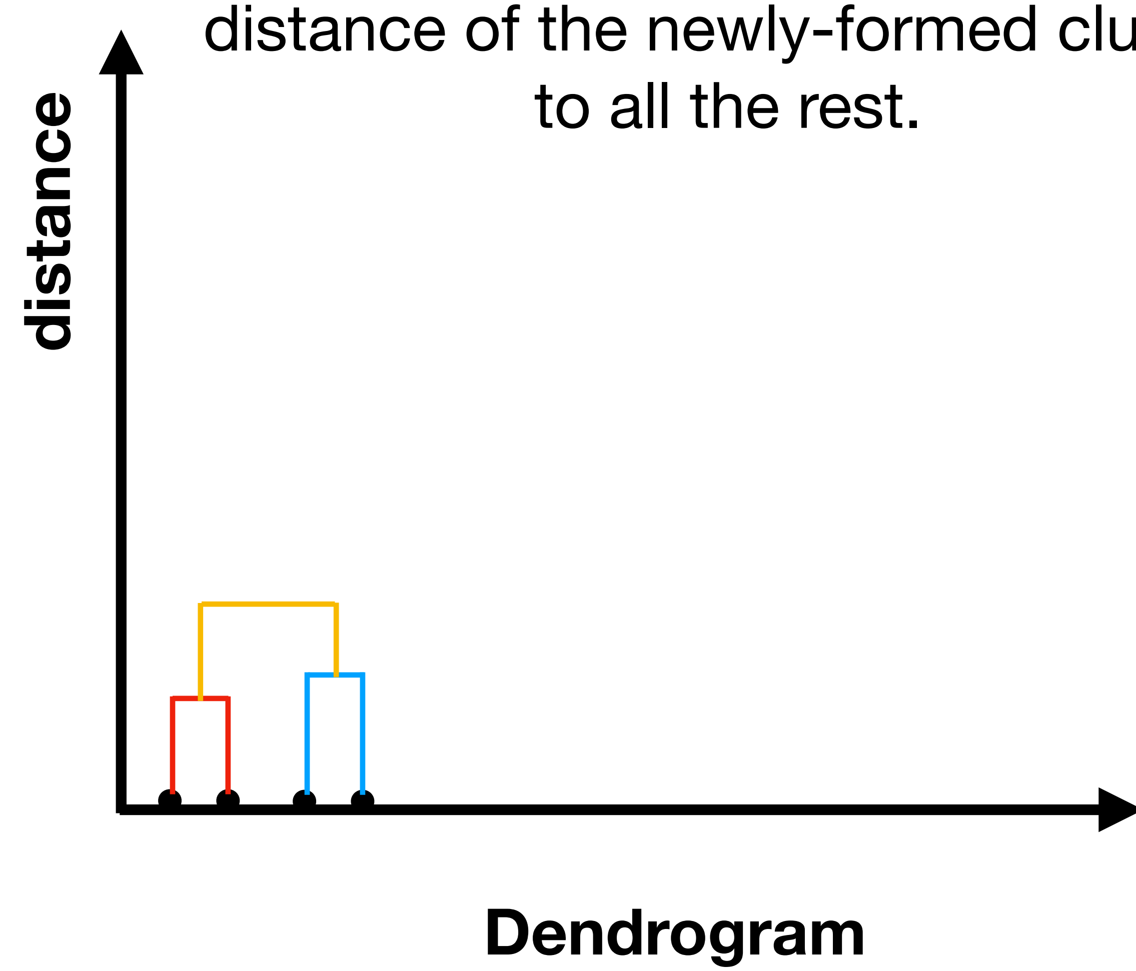
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



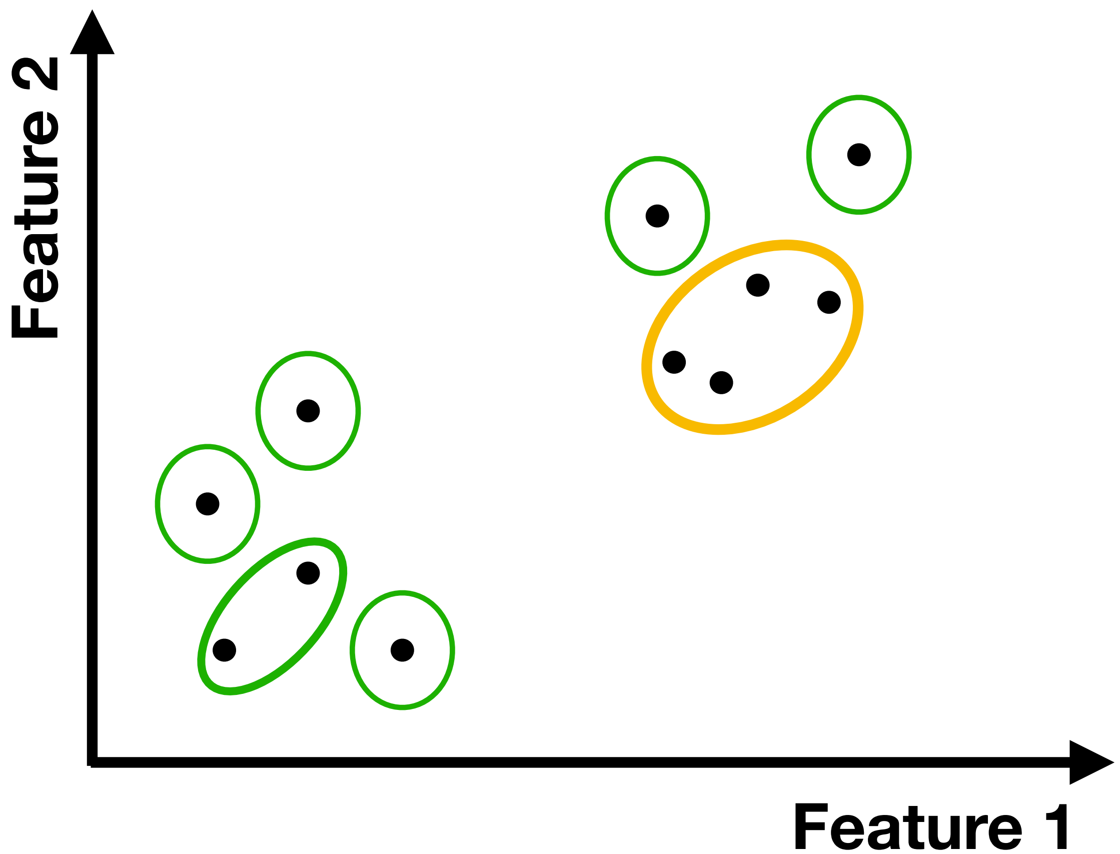


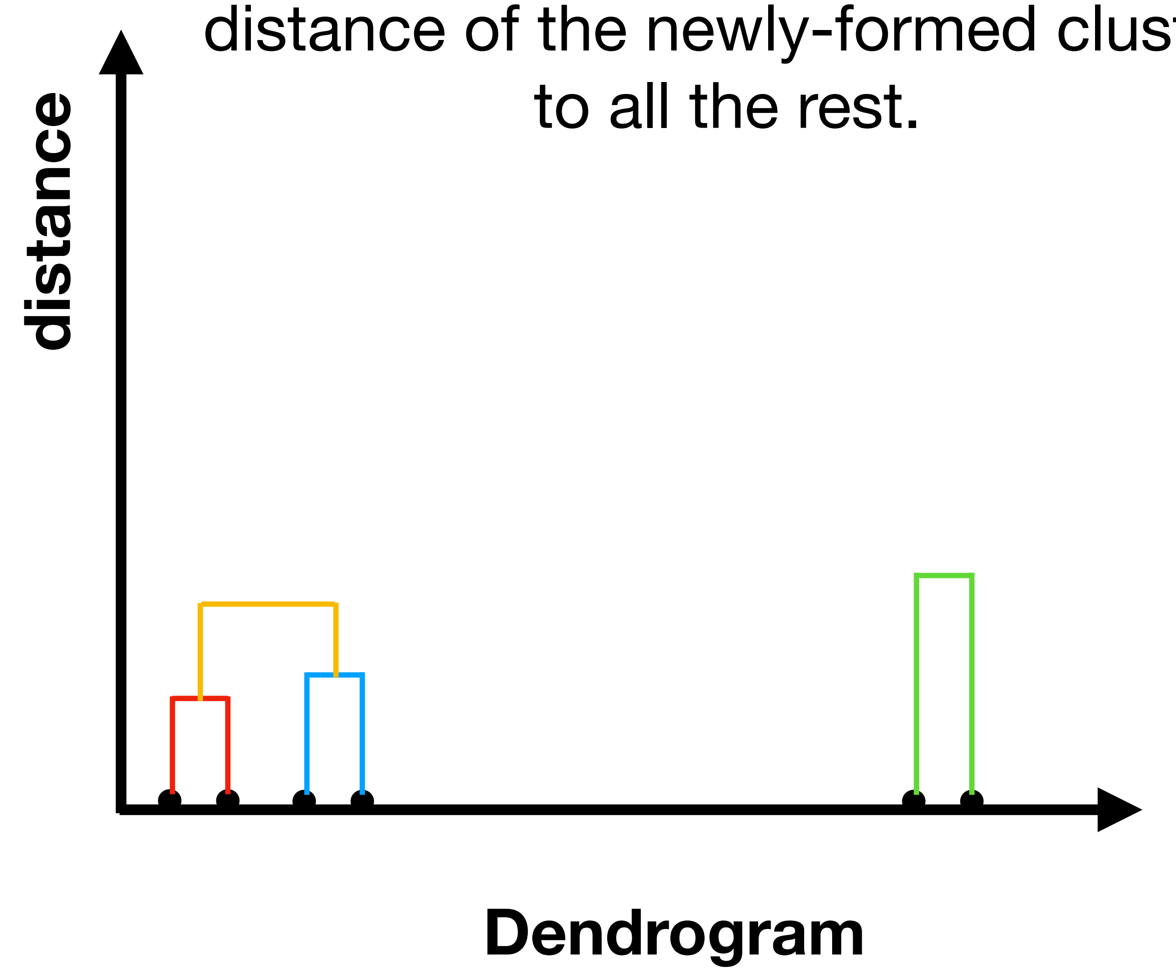
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



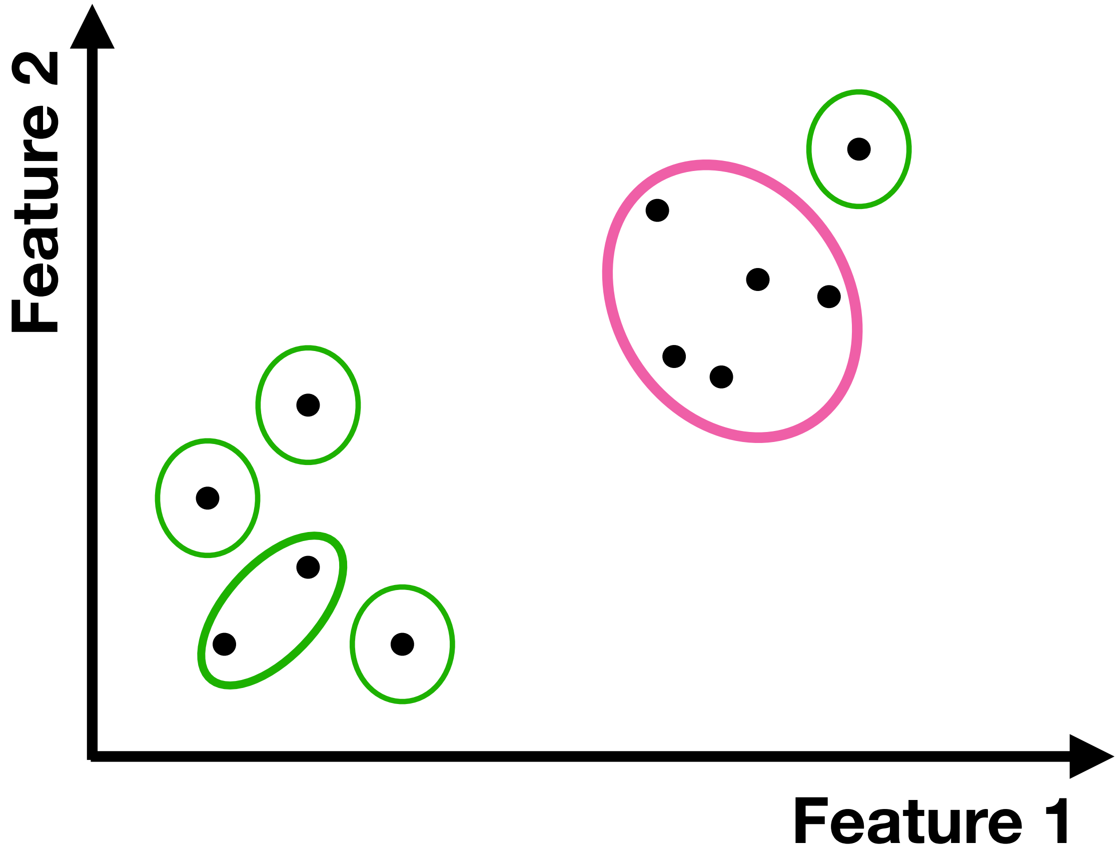


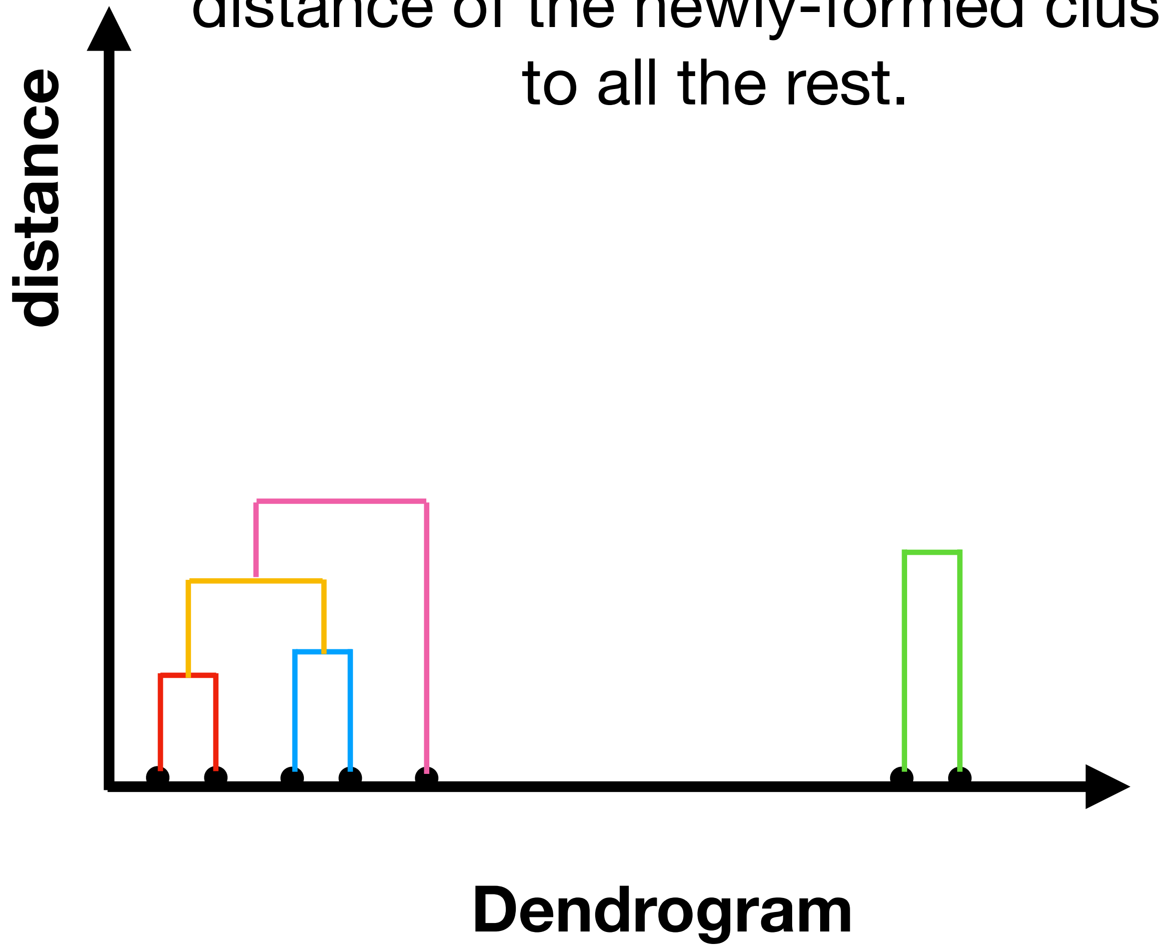
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



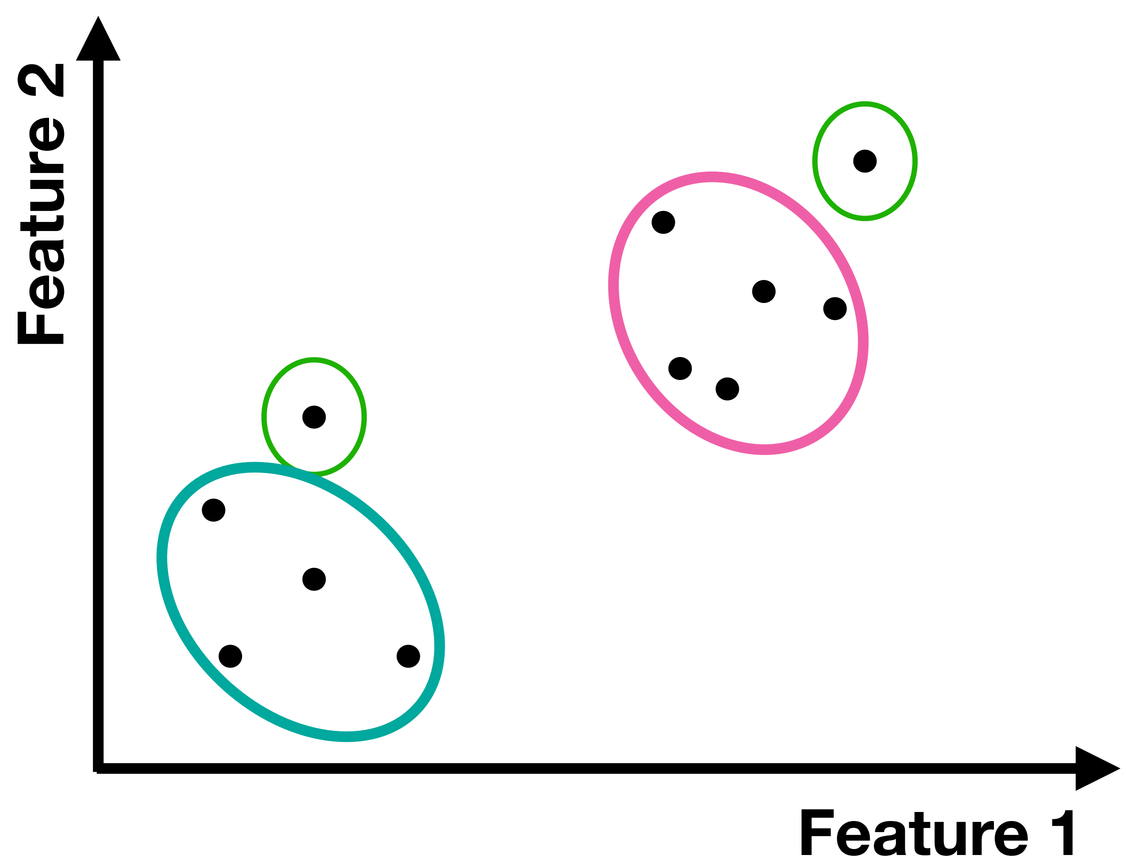


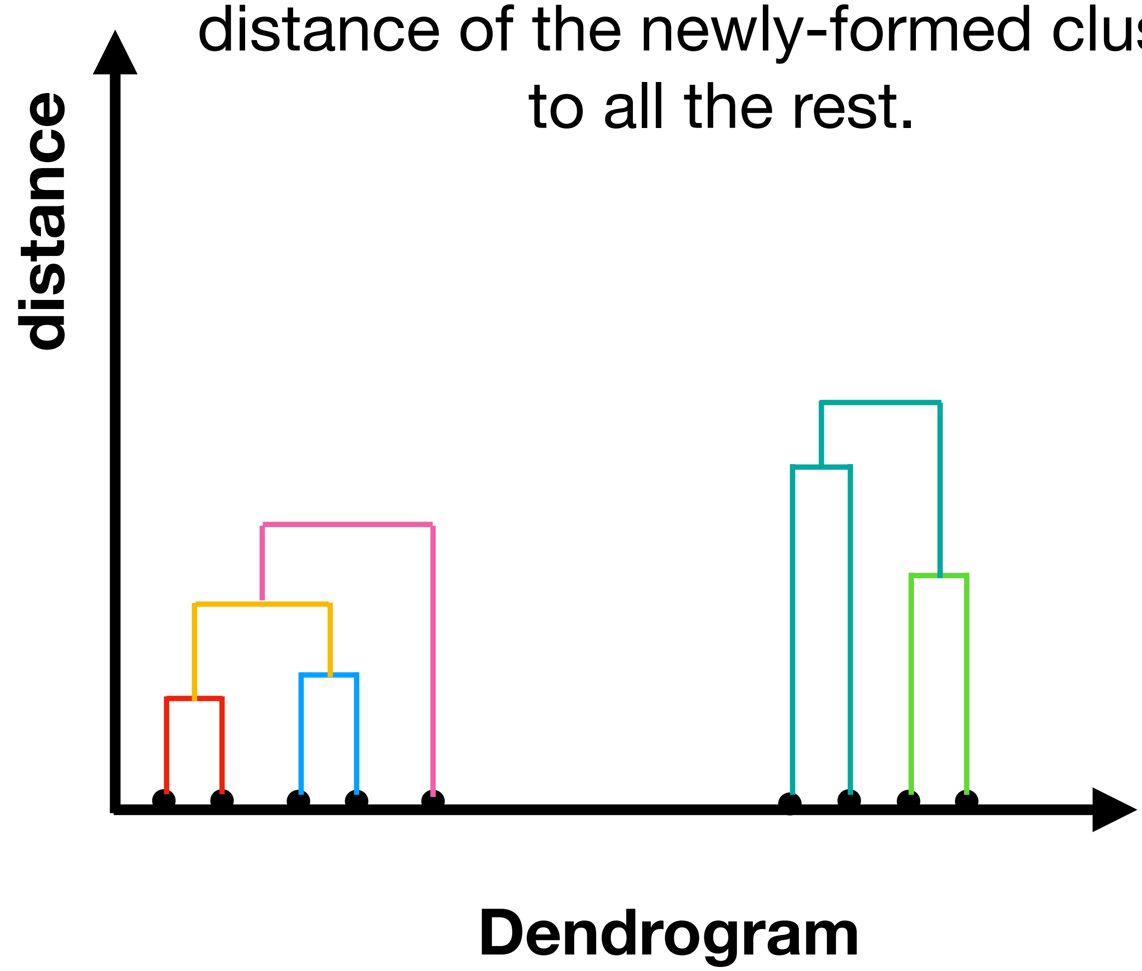
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



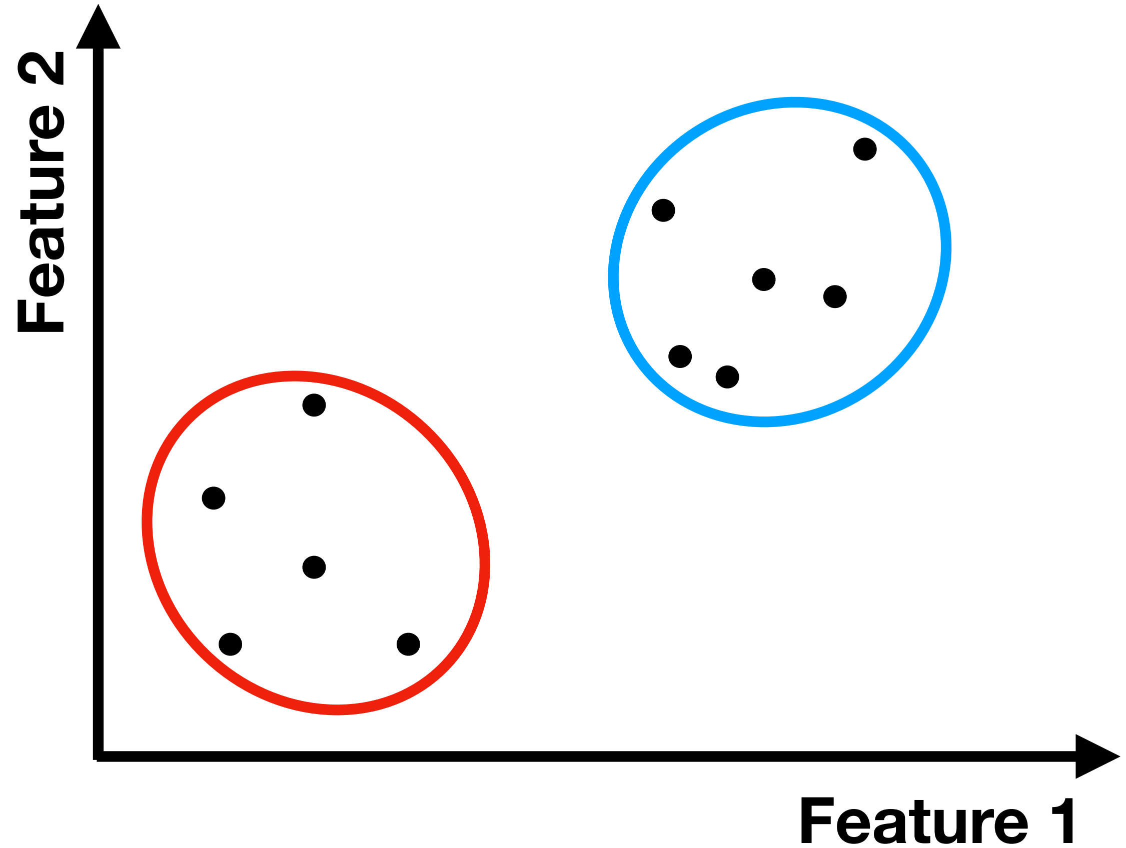


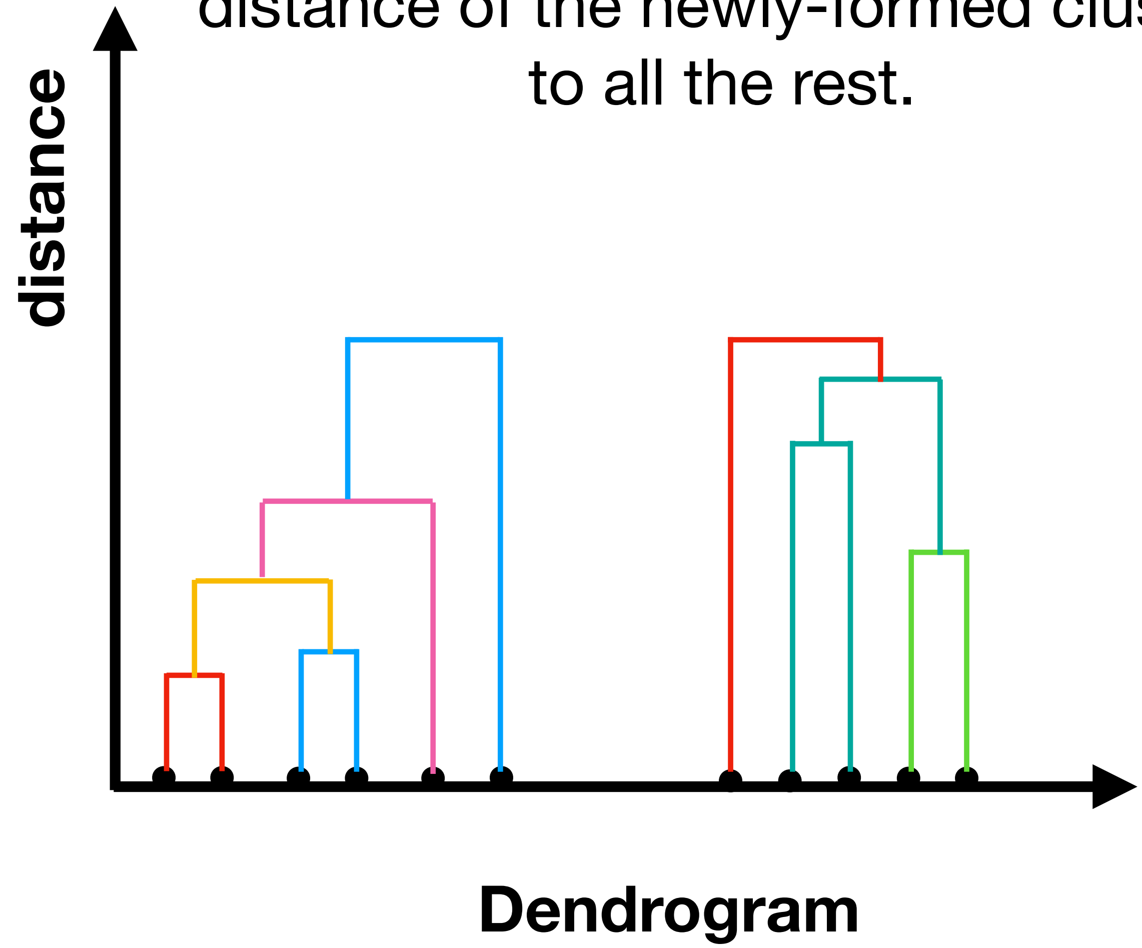
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

Next: the algorithm merges the two closest clusters into a single cluster. **Then,** the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



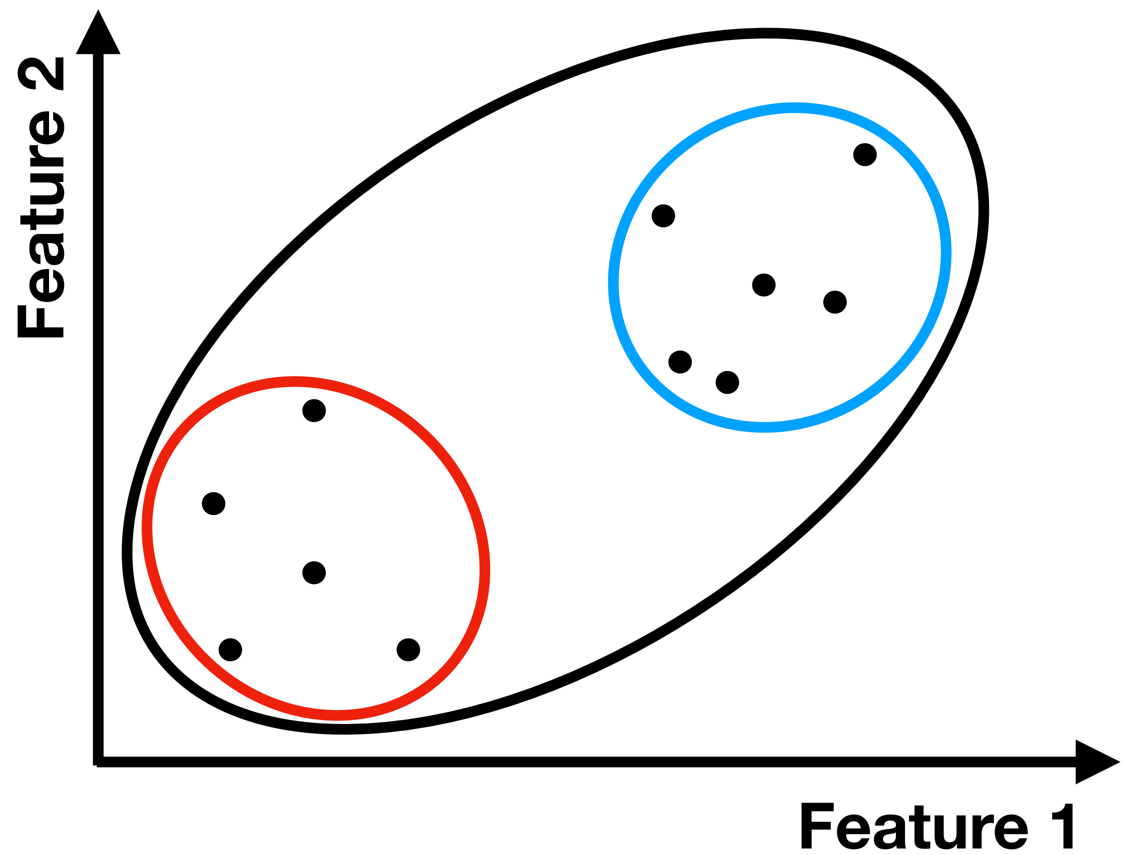


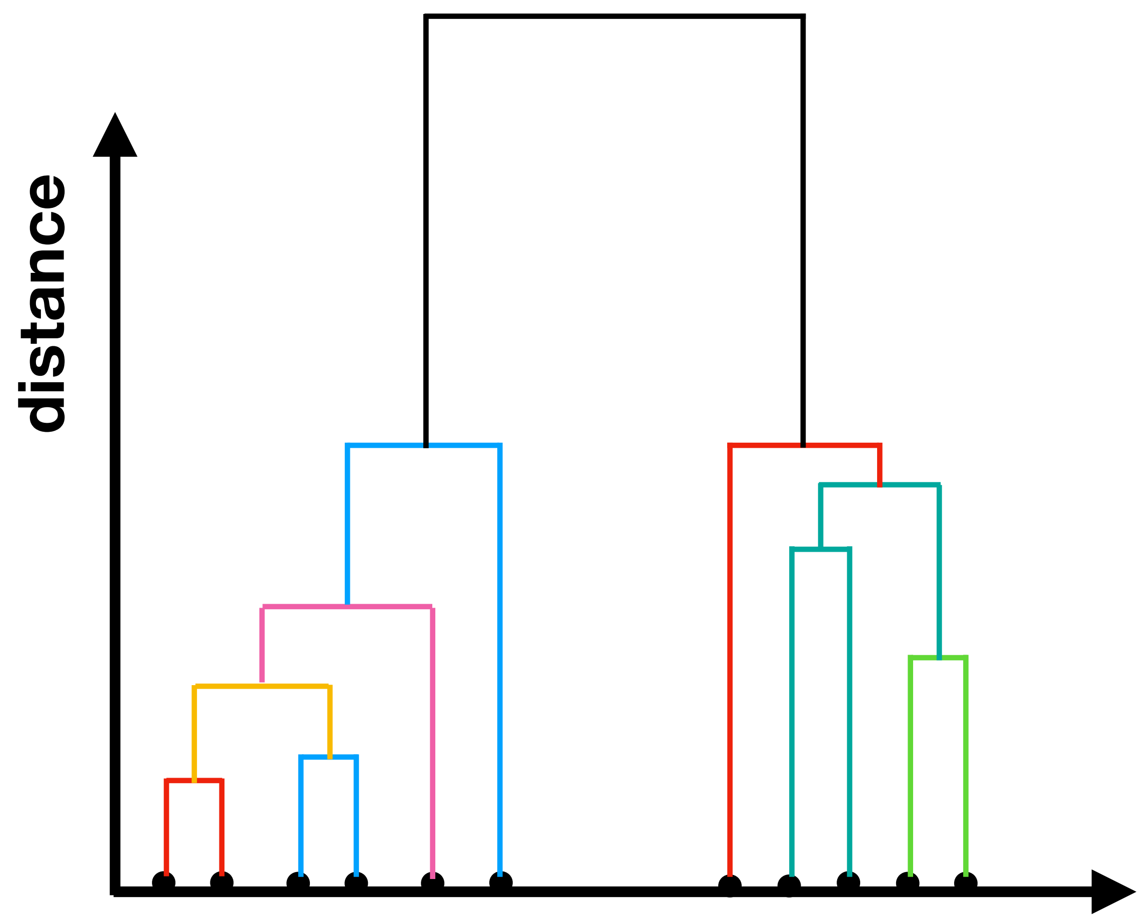
Hierarchal Clustering

Input: measured features, or a **distance matrix** that represents the pair-wise distances between the objects. Also, we must specify a **linkage method**.

Initialization: each object is a cluster of size 1.

The process stops when all the objects are merged into a single cluster





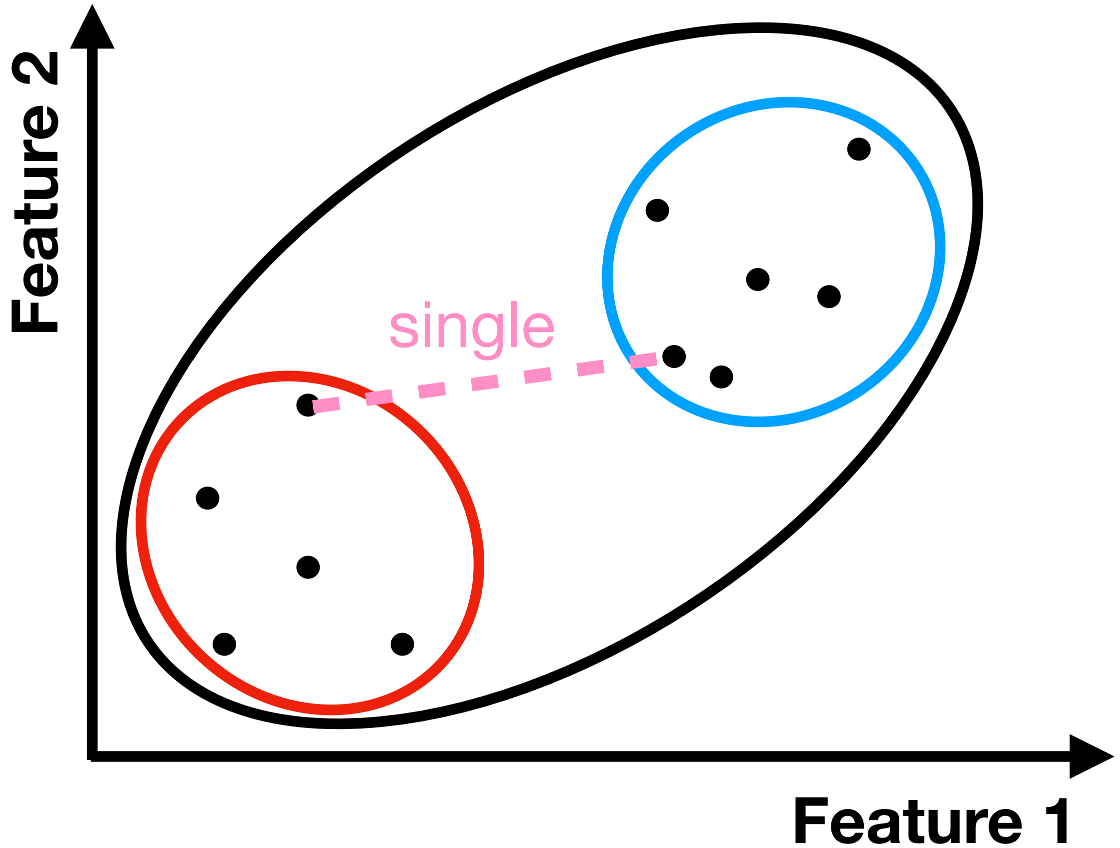
Dendrogram

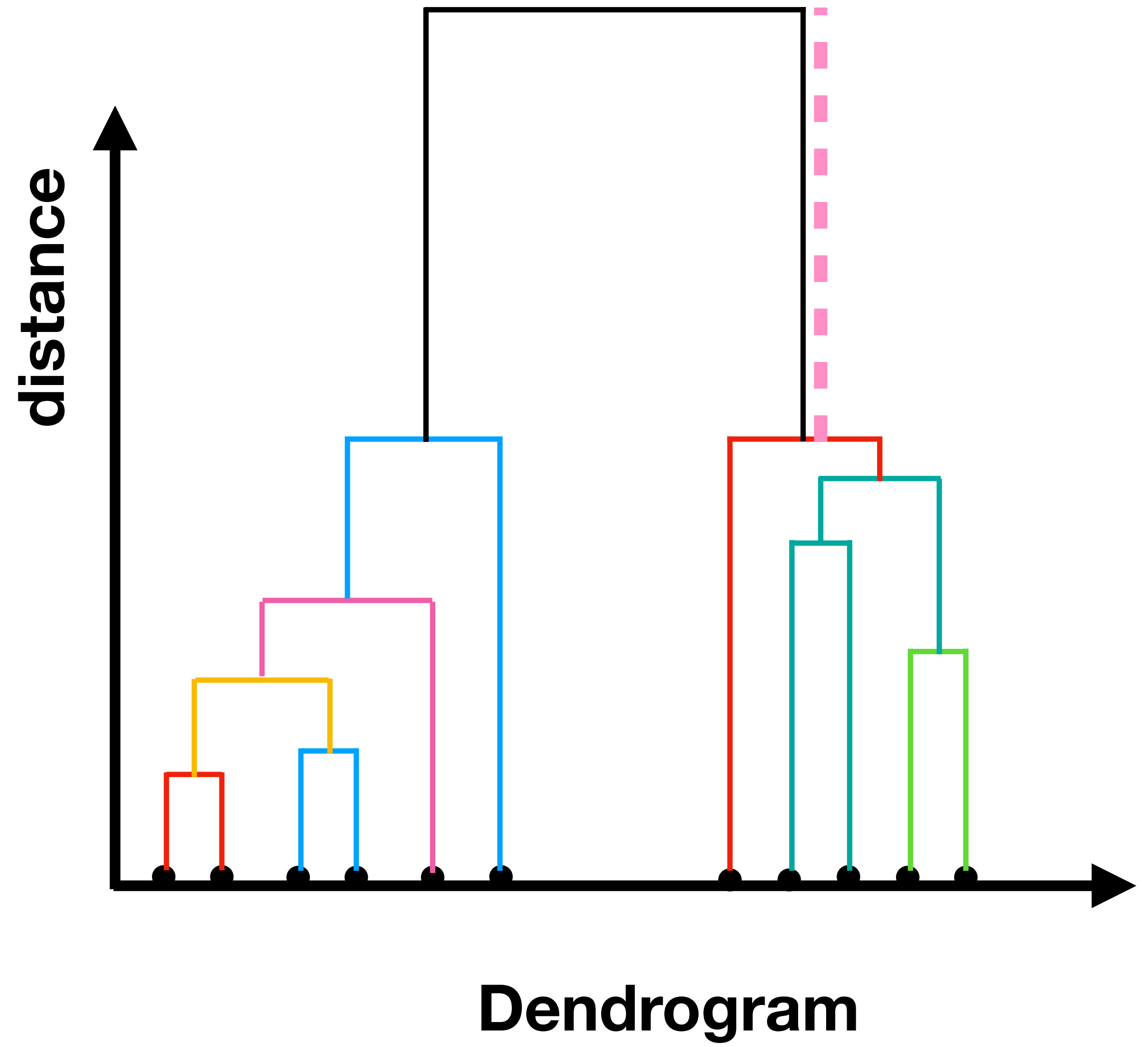
The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Internal choices and/or internal cost function:

The linkage method is used to define a distance between two newly formed clusters. Methods include: **single** (minimal), **complete** (maximal), **average**, etc.



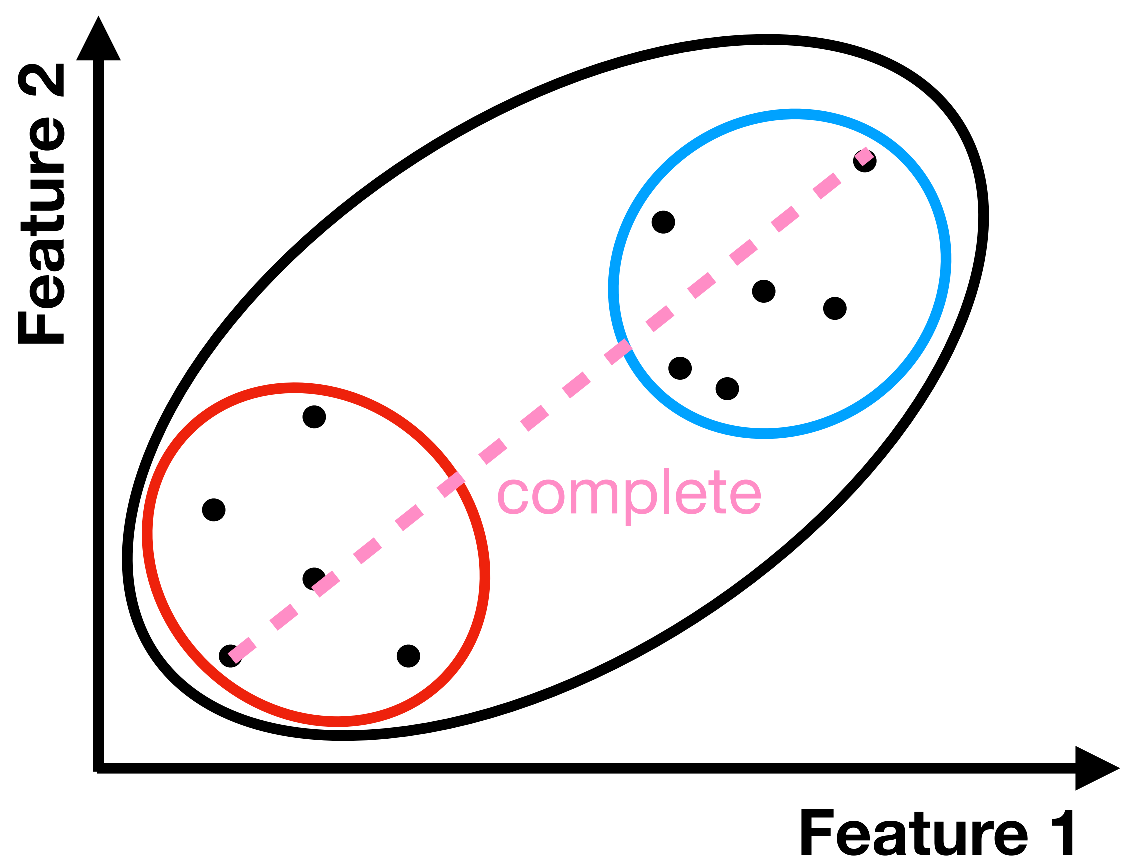


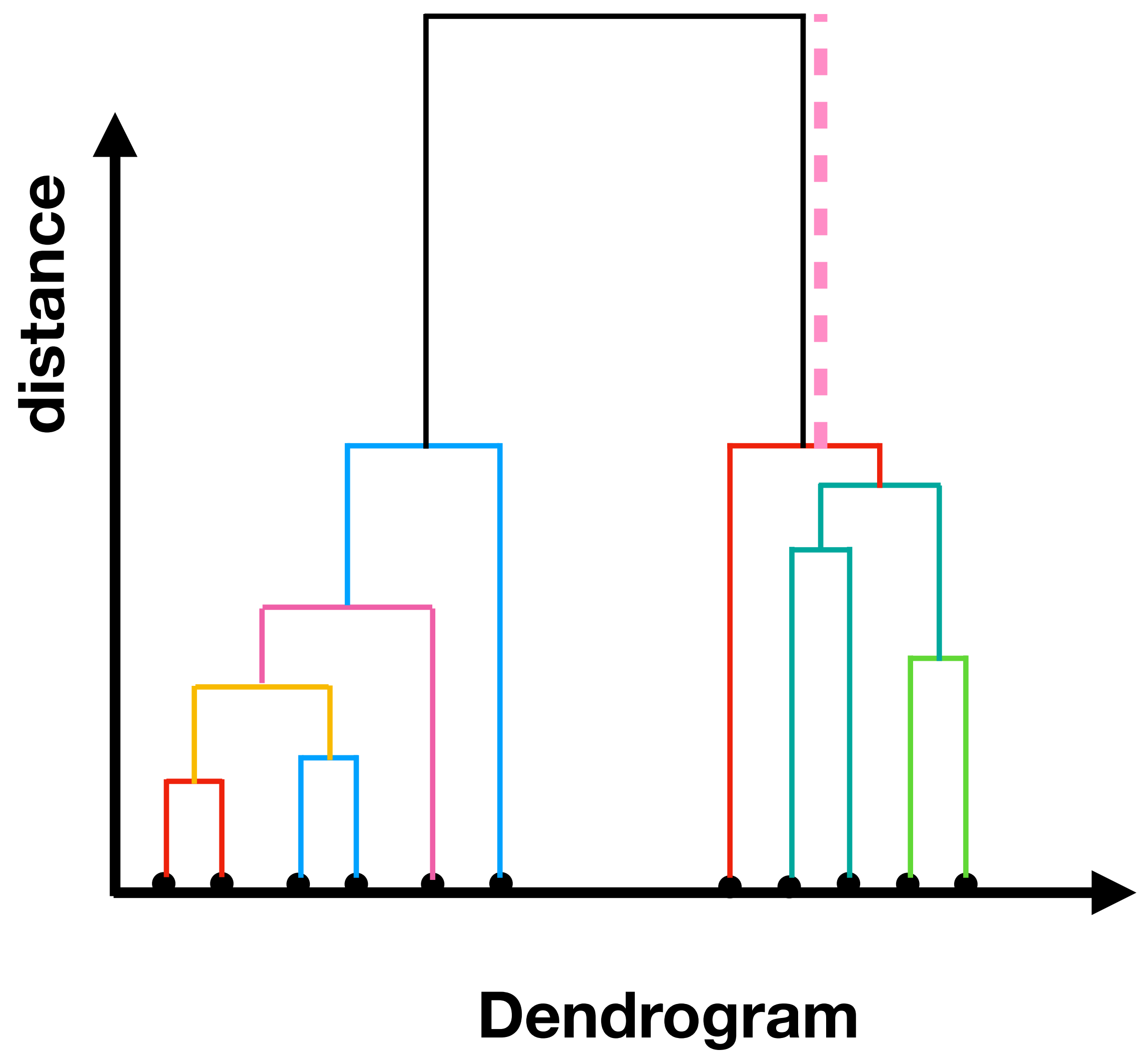
The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Internal choices and/or internal cost function:

The linkage method is used to define a distance between two newly formed clusters. Methods include: **single** (minimal), **complete** (maximal), **average**, etc.



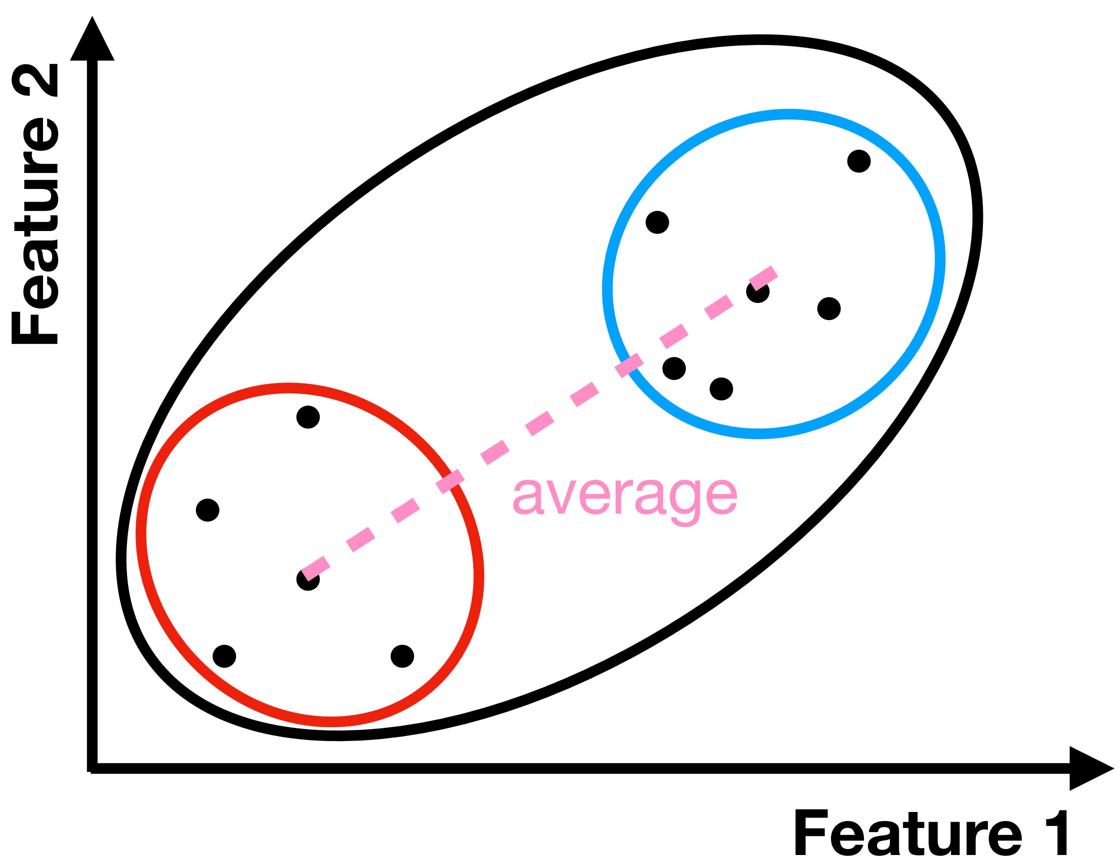


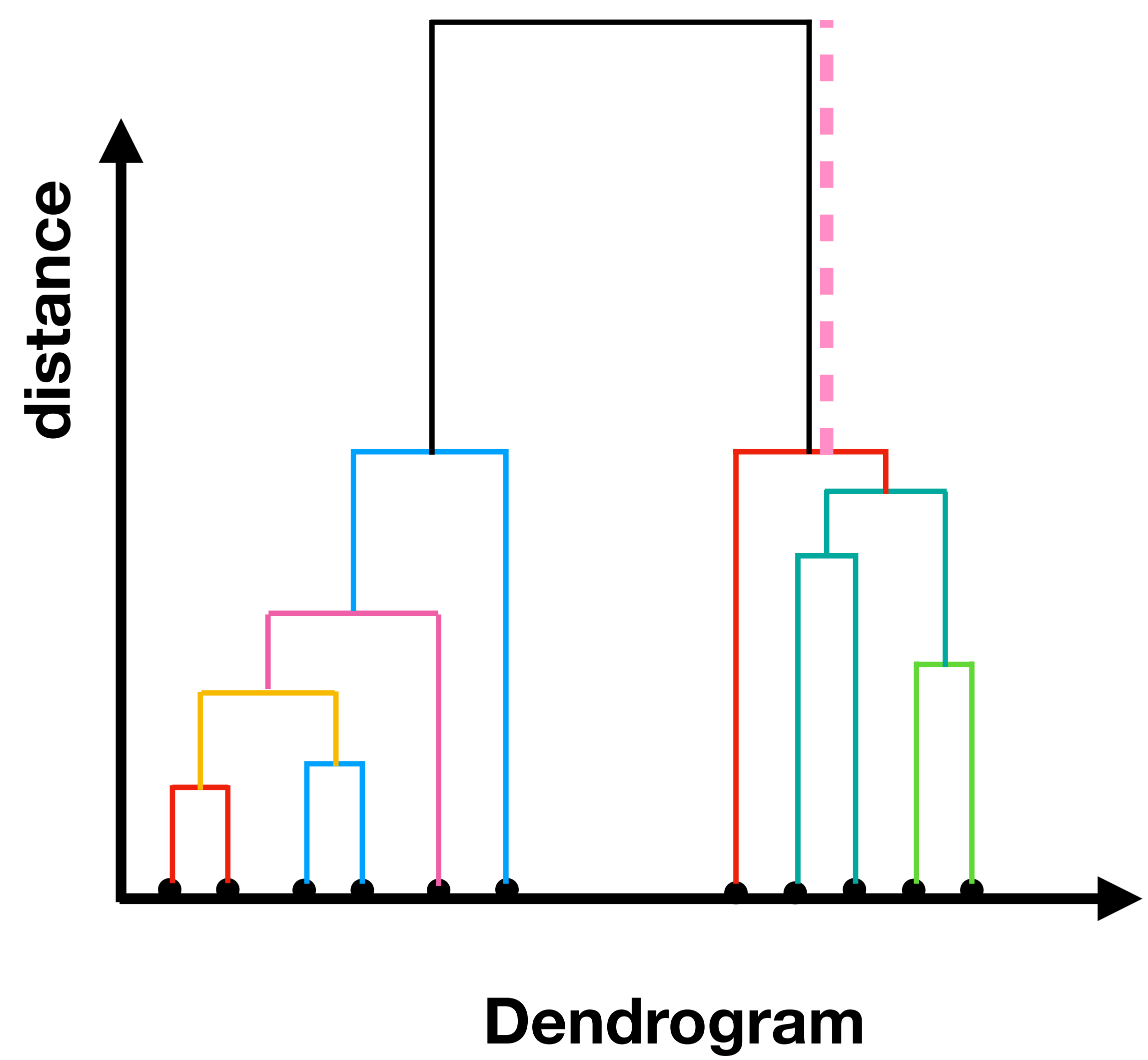
The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Internal choices and/or internal cost function:

The linkage method is used to define a distance between two newly formed clusters. Methods include: **single** (minimal), **complete** (maximal), **average**, etc.

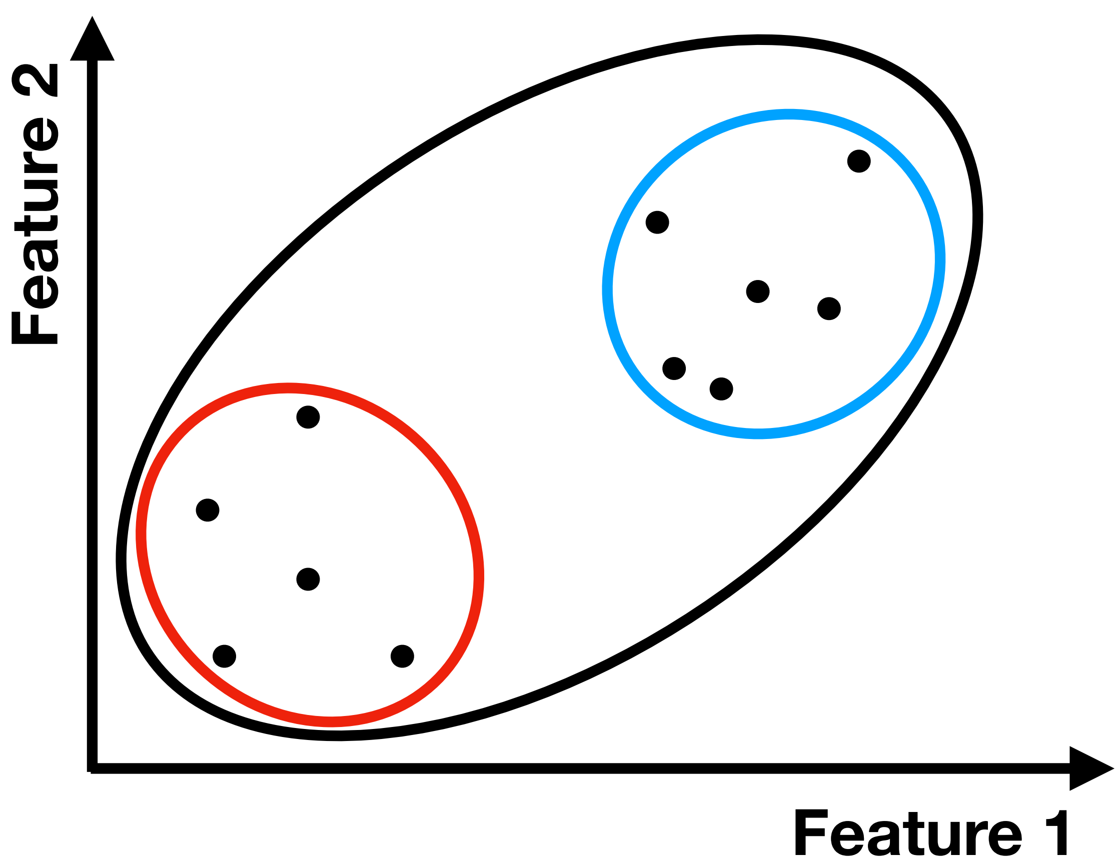


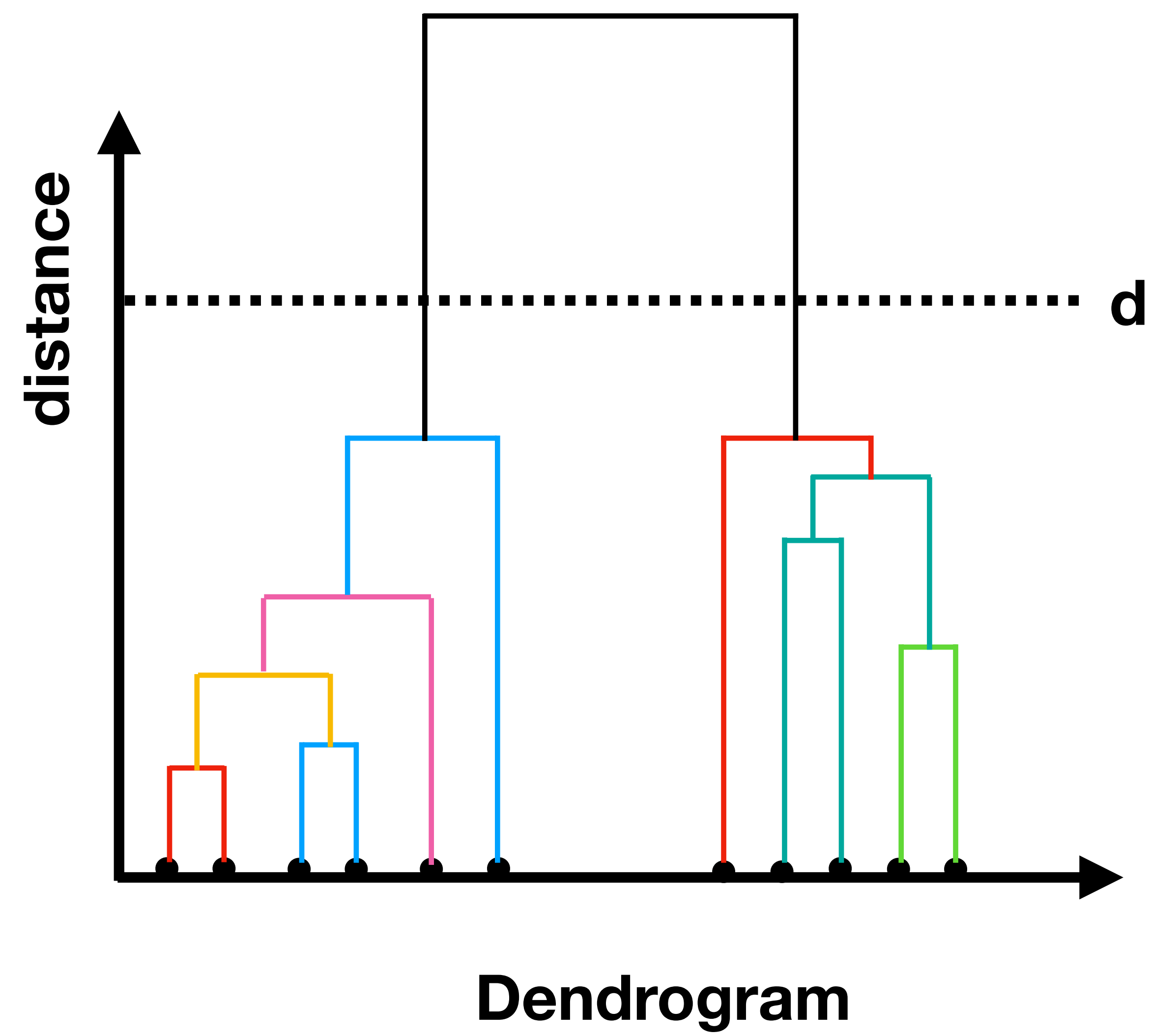


The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, \ldots\}) = \overrightarrow{y}$$

Hyper-parameters: clusters are defined beneath a threshold d. Alternatively, we can select a threshold d that corresponds to the desired number of clusters, k.

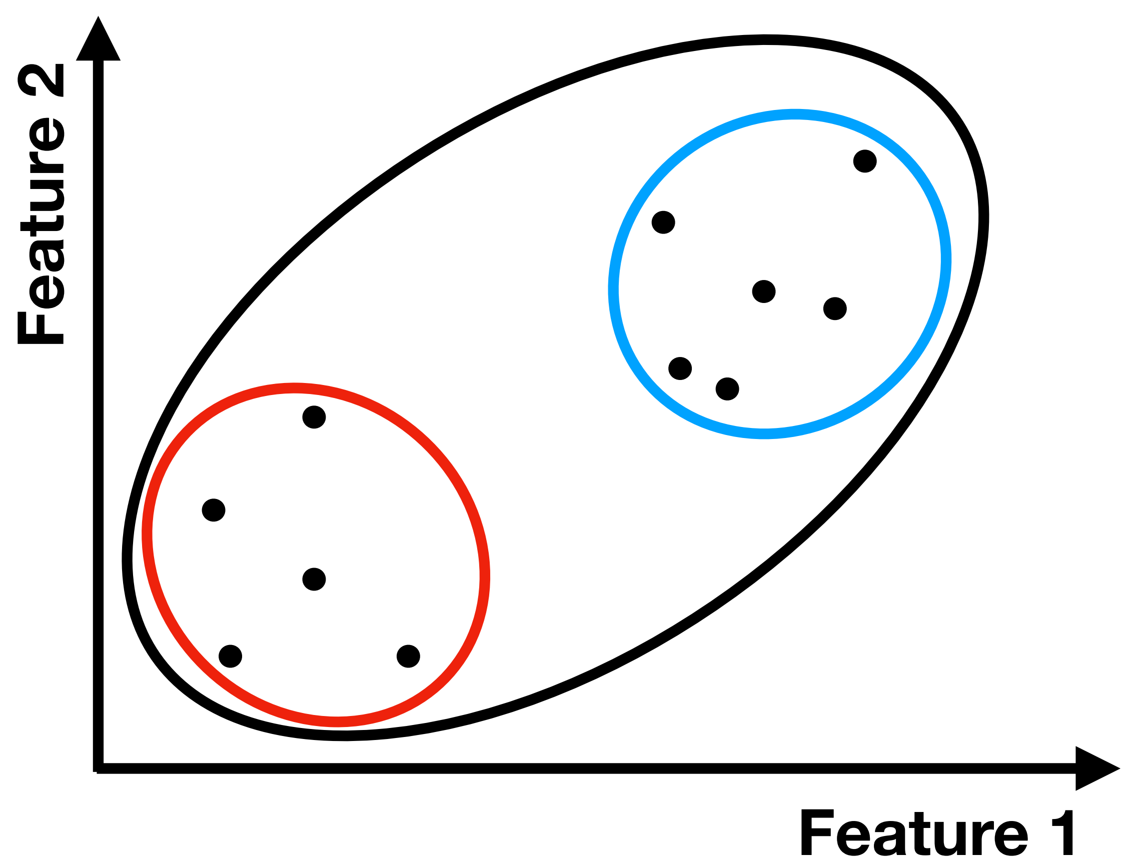


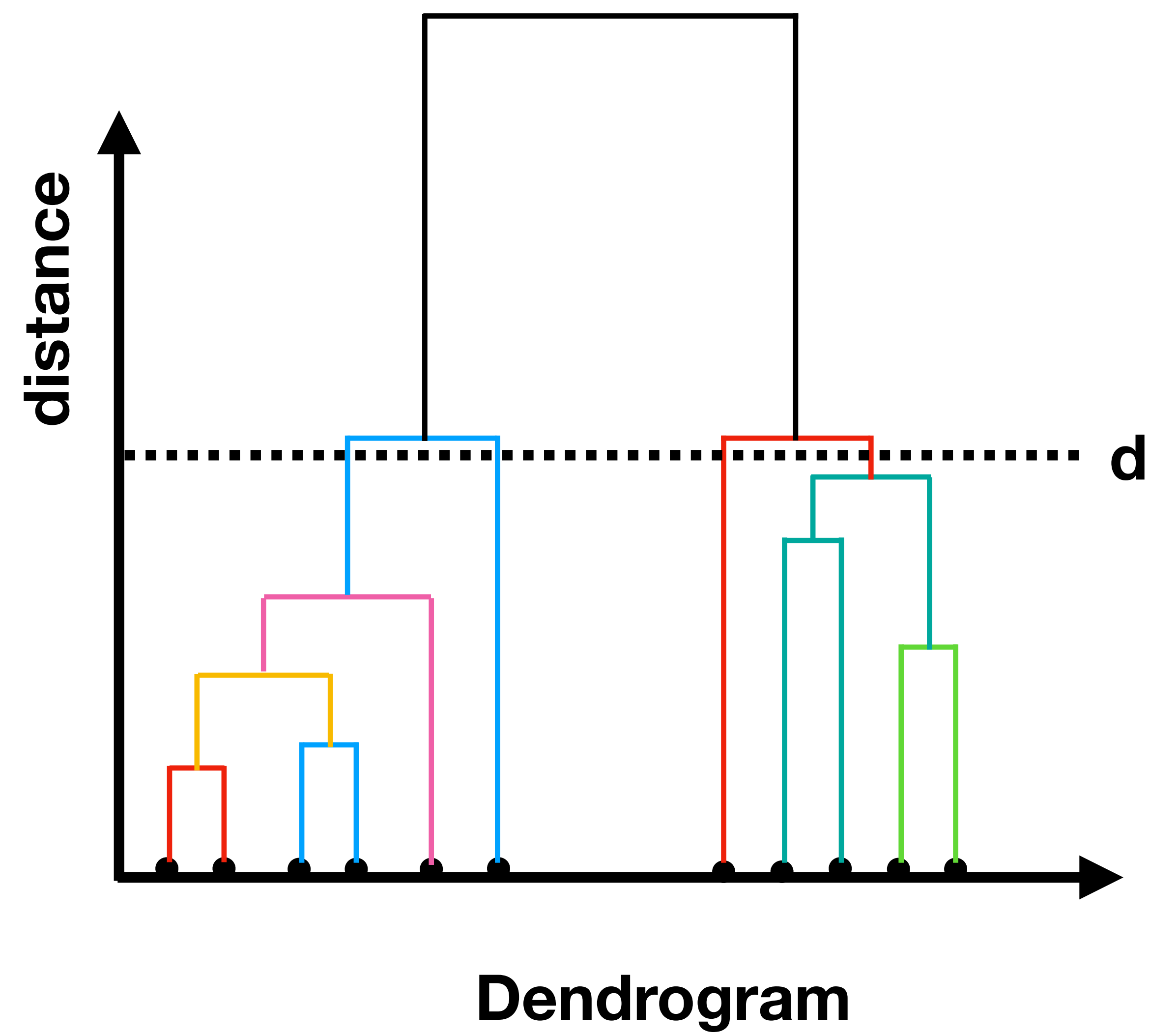


The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Hyper-parameters: clusters are defined beneath a threshold d. Alternatively, we can select a threshold d that corresponds to the desired number of clusters, k.

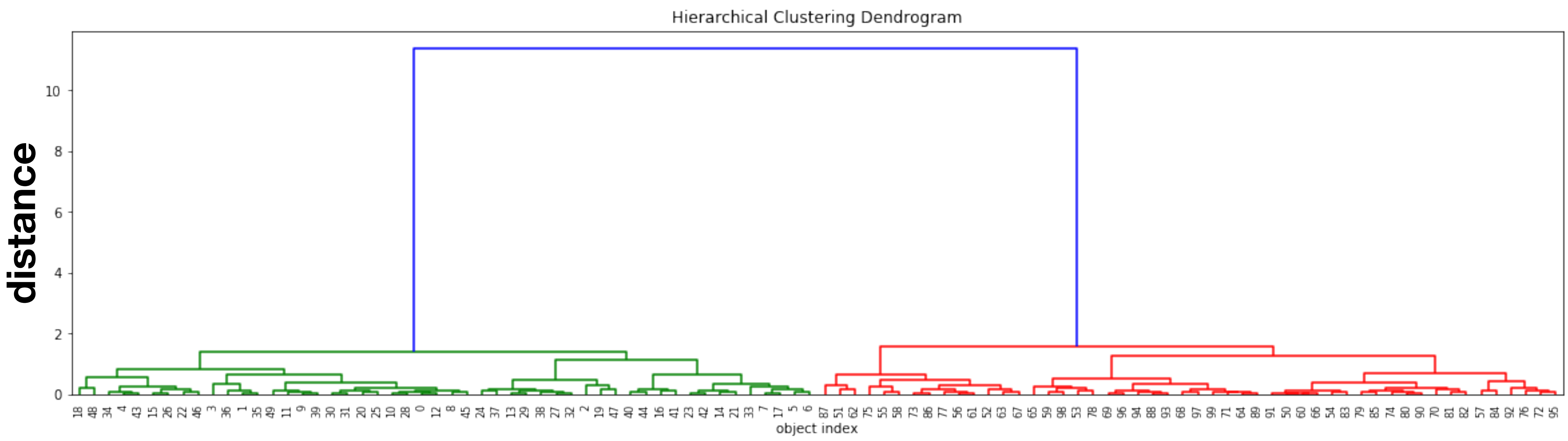




The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

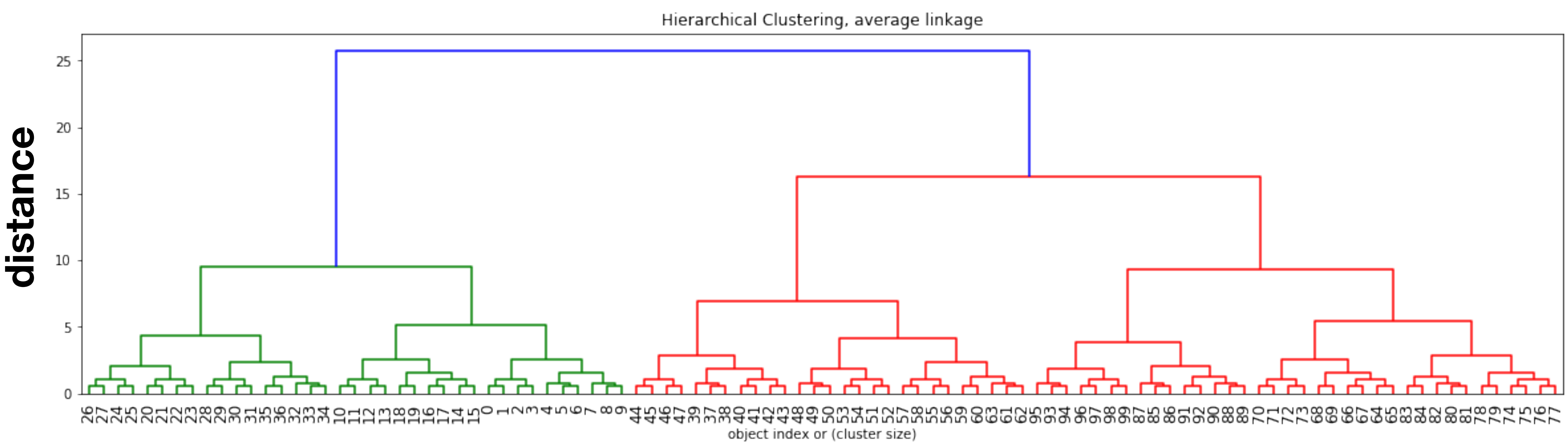
Hyper-parameters: clusters are defined beneath a threshold d. Alternatively, we can select a threshold d that corresponds to the desired number of clusters, k.
We can use the resulting dendrogram to choose a “good” threshold:



The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

Hyper-parameters: clusters are defined beneath a threshold d. Alternatively, we can select a threshold d that corresponds to the desired number of clusters, k.
We can use the resulting dendrogram to choose a "good" threshold:



The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

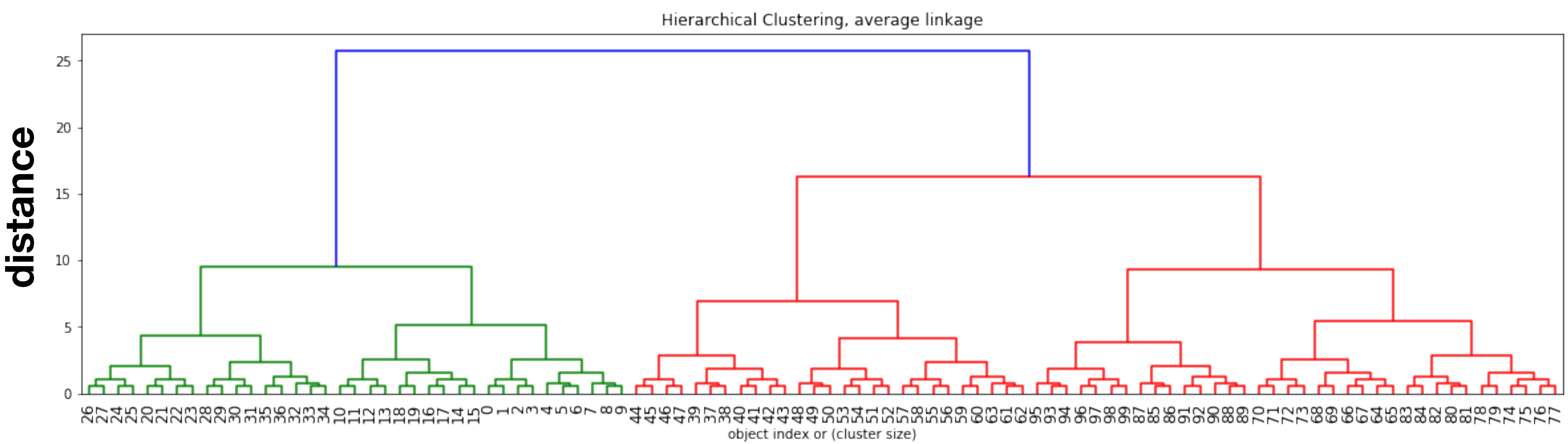
Input dataset: can either be a list of objects with measured properties, or a distance matrix that represents pair-wise distances between objects.

What happens if we have an outlier in the dataset?

The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, ...\}) = \overrightarrow{y}$$

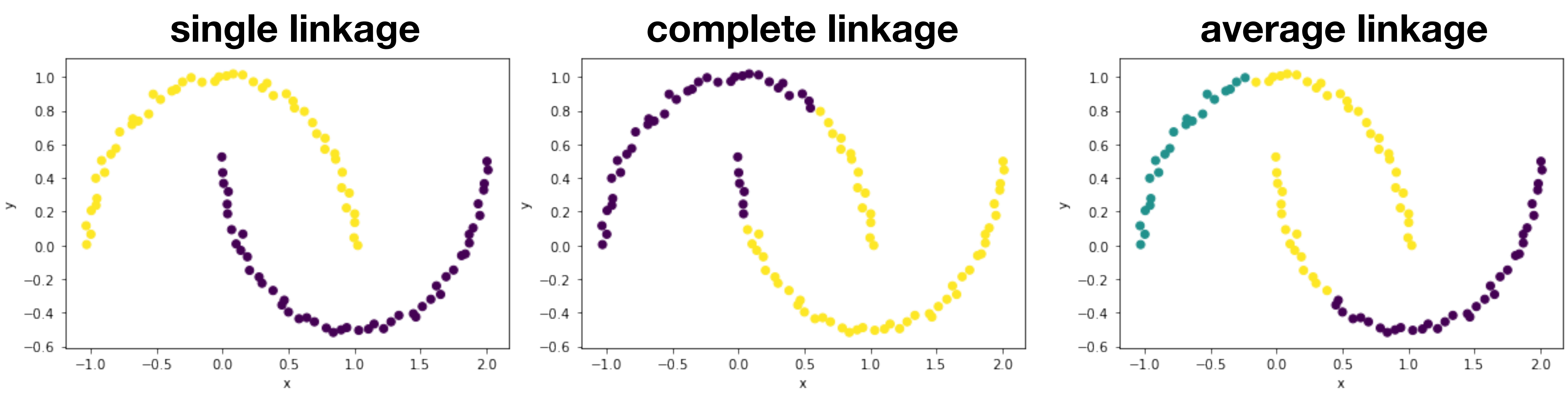
Input dataset: can either be a list of objects with measured properties, or a distance matrix that represents pair-wise distances between objects.
What happens if the dataset does not have clear clusters?



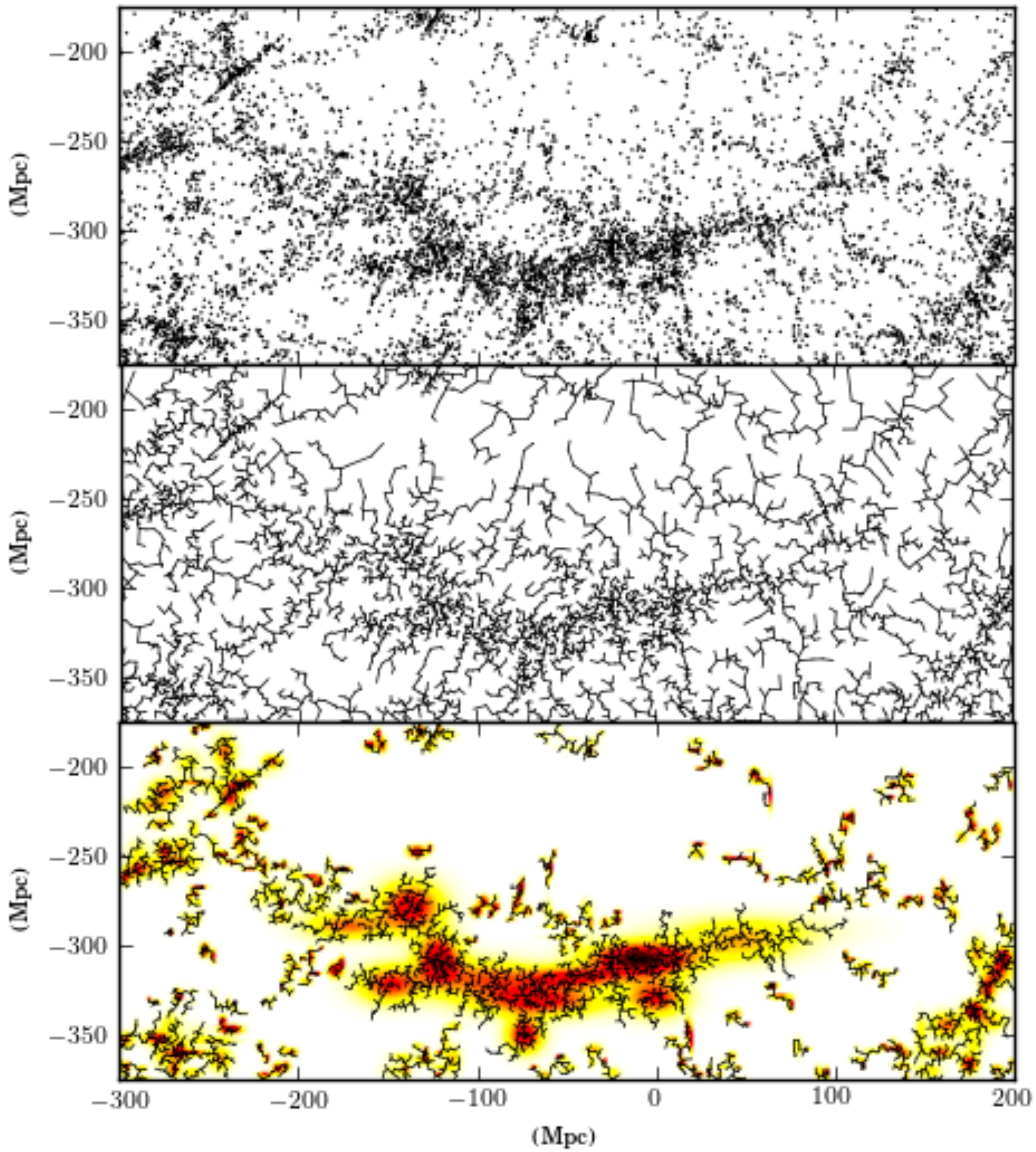
The anatomy of Hierarchal Clustering

$$f(\overrightarrow{X}, \{a_1, a_2, \ldots\}) = \overrightarrow{y}$$

Input dataset: can either be a list of objects with measured properties, or a distance matrix that represents pair-wise distances between objects.
Different linkage methods are helpful with different datasets.



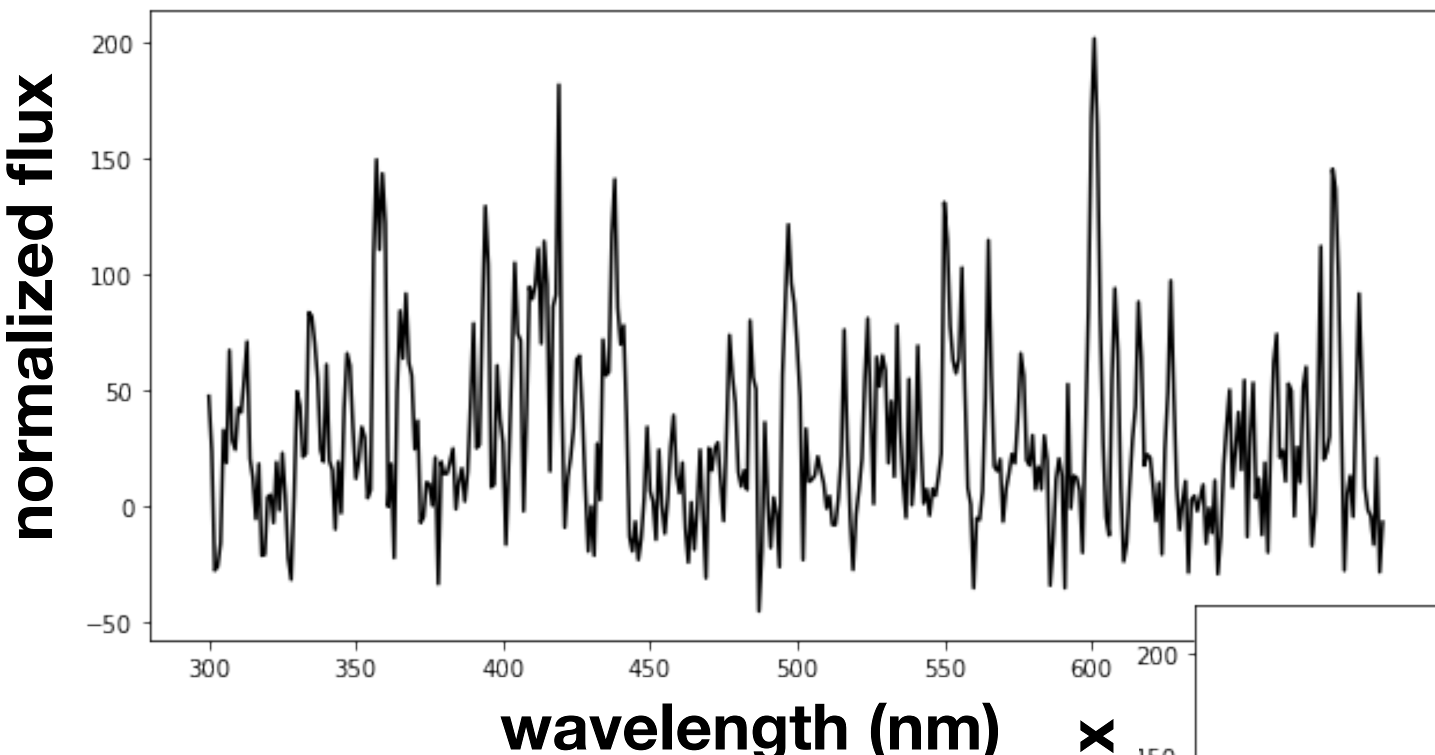
Hierarchal Clustering in Astronomy

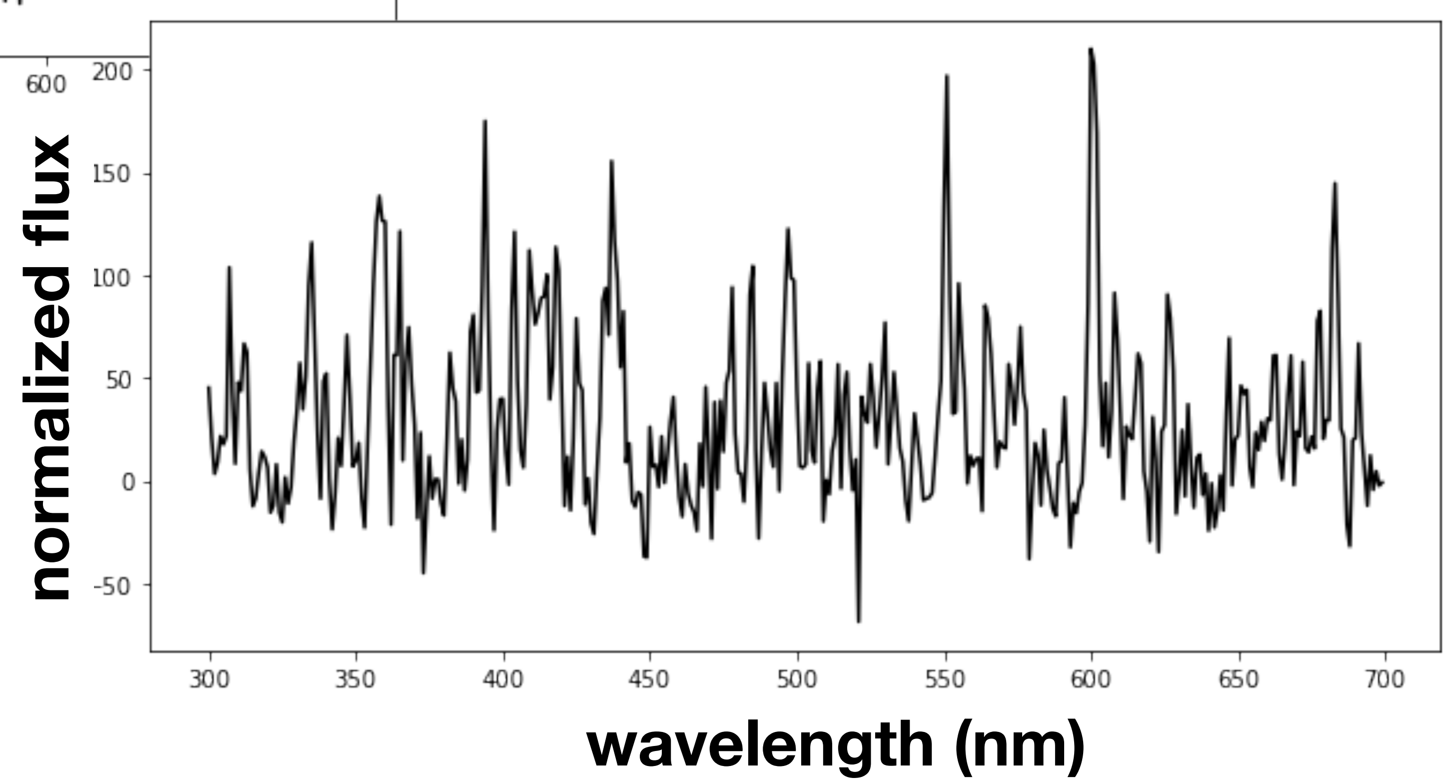


“Statistics, Data Mining, and Machine Learning in Astronomy”, by Ivezic, Connolly, Vanderplas, and Gray (2013).

Visualizing similarity matrices with Hierarchical Clustering

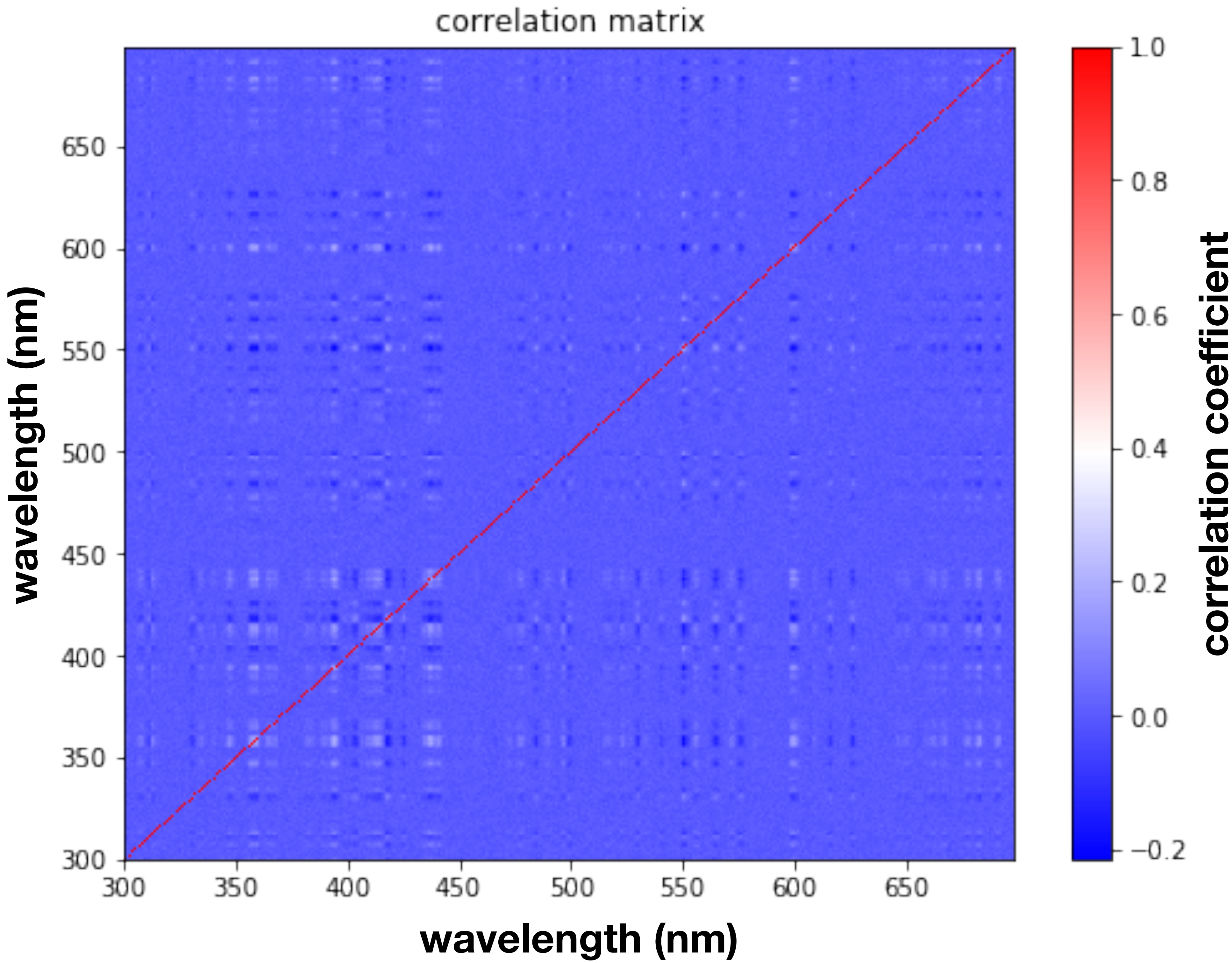
Input: 10,000 emission line spectra, covering the wavelength range 300 - 700 nm. There are ~90 emission lines in each spectrum, with an average SNR of 2-4.





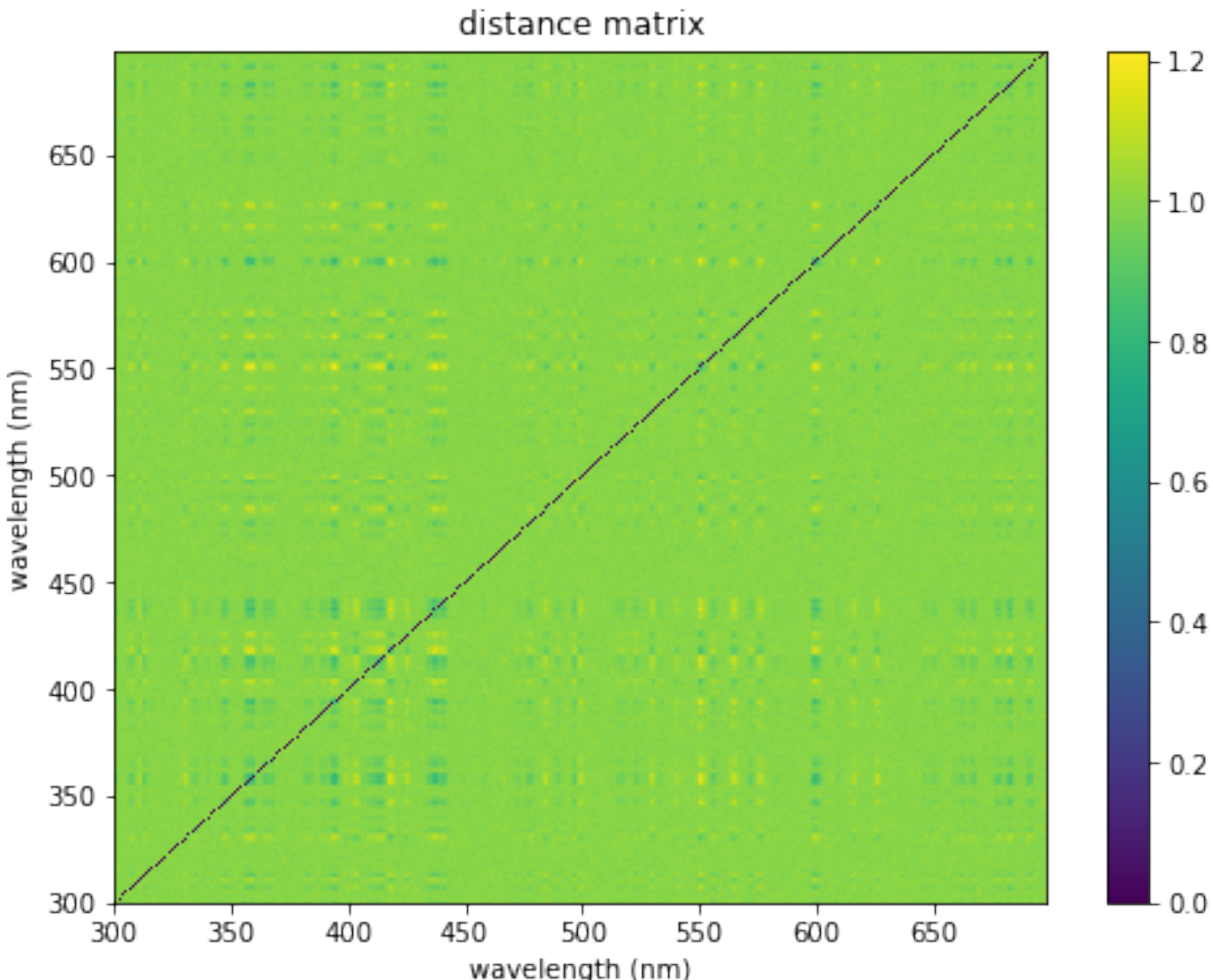
Visualizing similarity matrices with Hierarchical Clustering

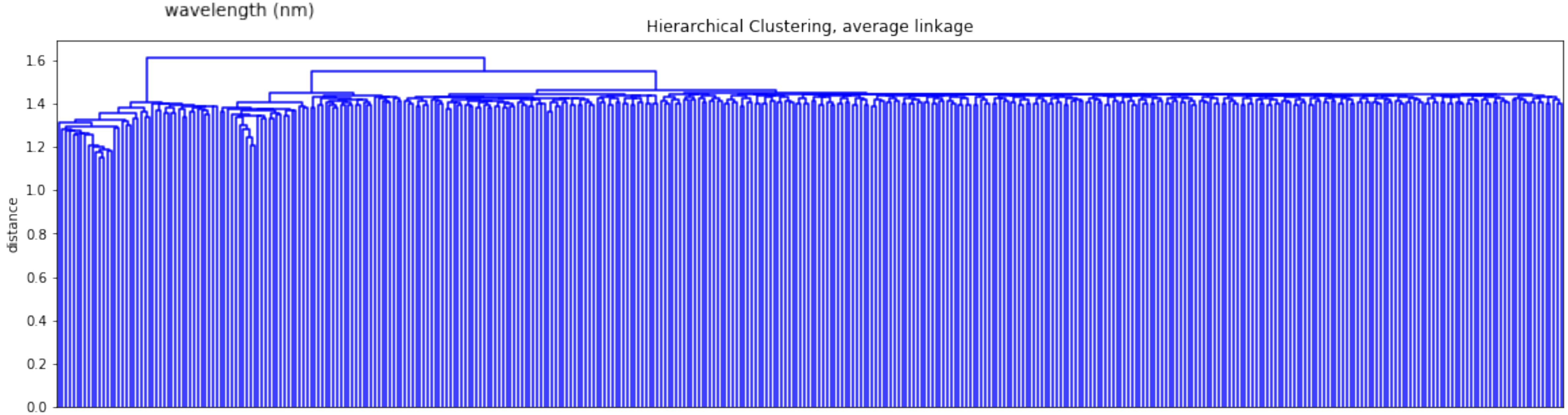
We compute a correlation matrix of all the observed wavelengths.



Visualizing similarity matrices with Hierarchical Clustering

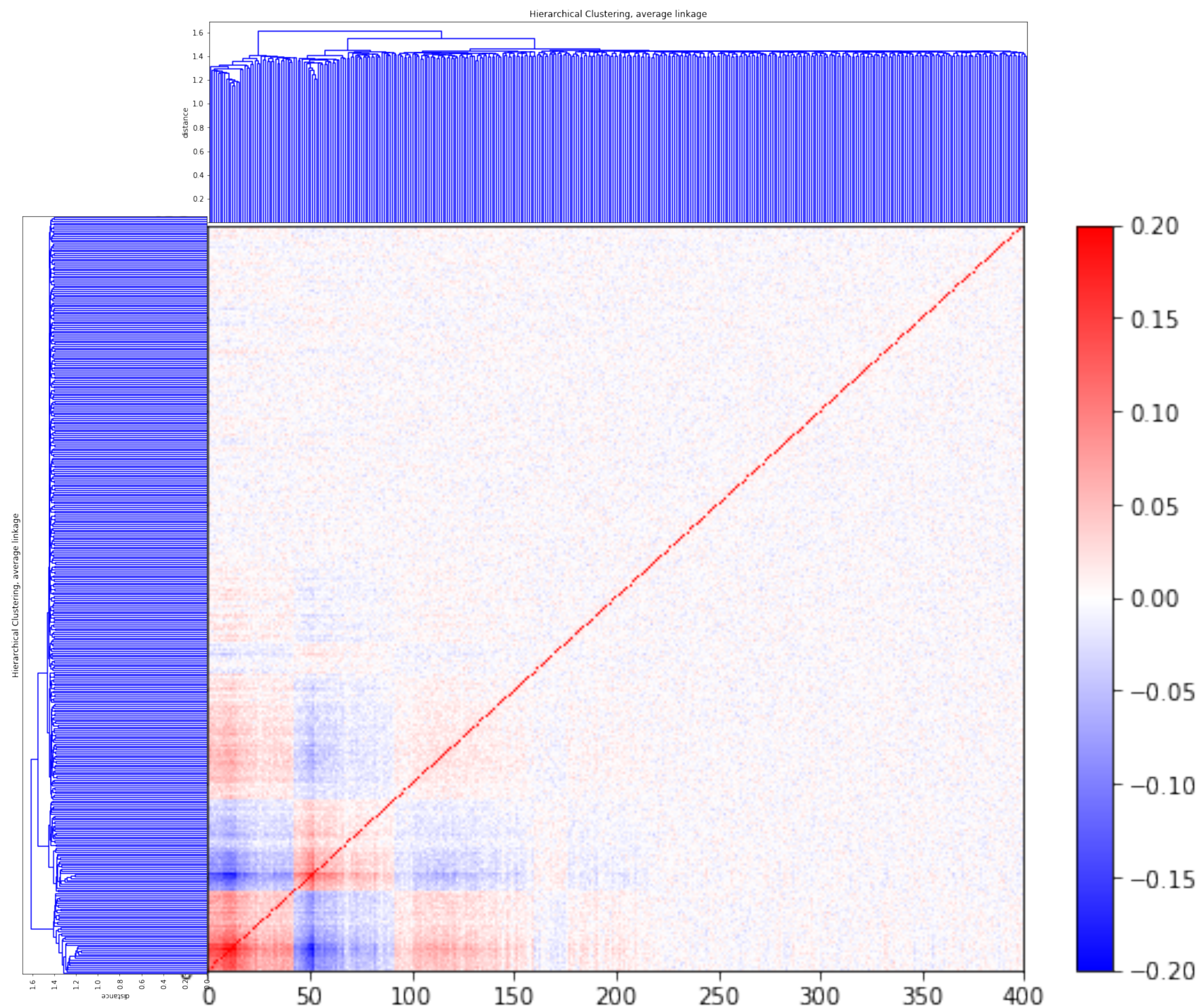
We convert the correlation matrix to a distance matrix, and build a dendrogram





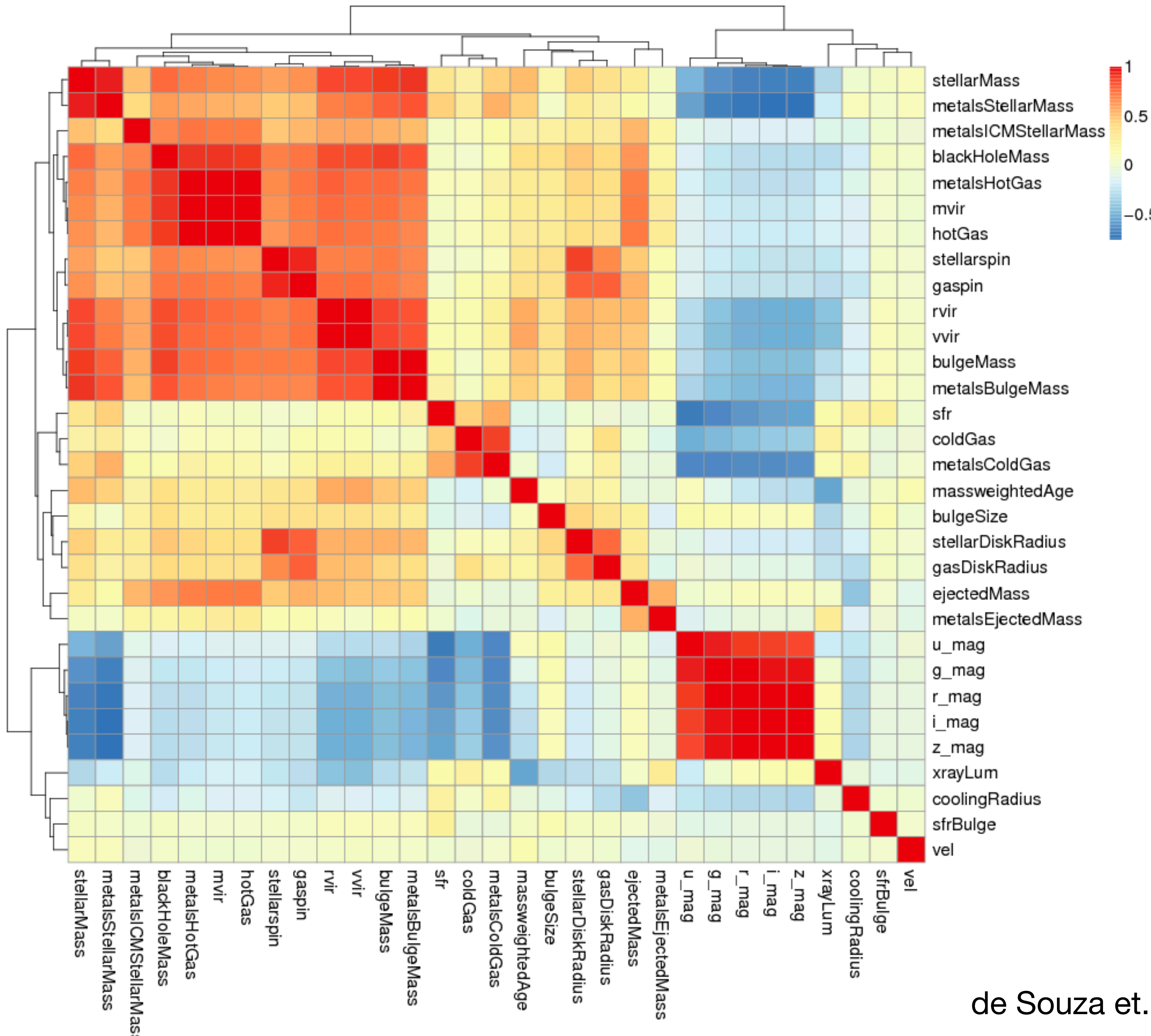
Visualizing similarity matrices with Hierarchical Clustering

We reorder the correlation matrix (the wavelengths) according to the resulting dendrogram.



reordered axis

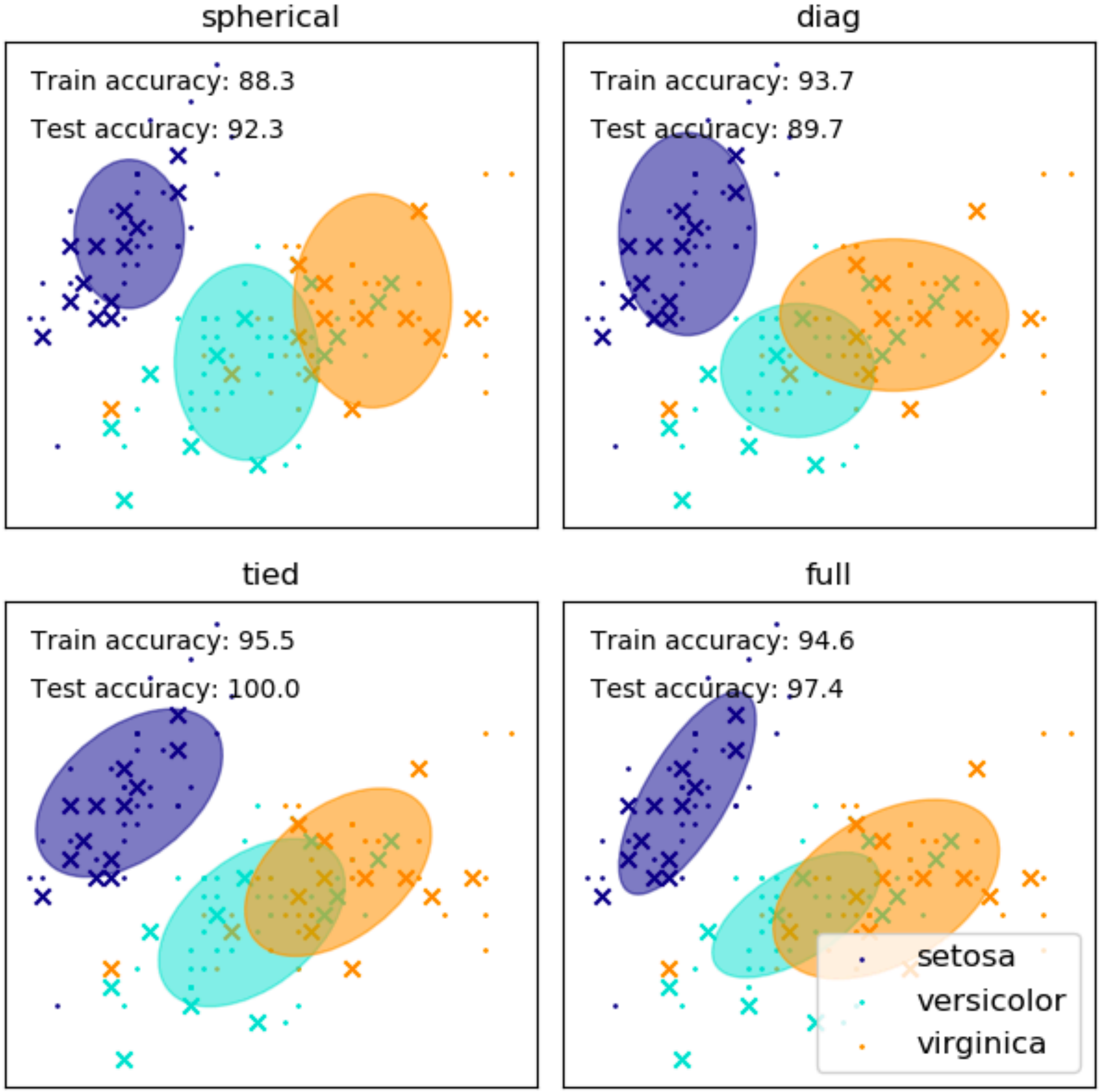
Visualizing similarity matrices with Hierarchical Clustering



de Souza et. al 2015

Questions?

Gaussian Mixture models



See: http://scikit-learn.org/stable/auto_examples/mixture/plot_gmm_covariances.html#sphx-glr-auto-examples-mixture-plot-gmm-covariances-py

Questions?