
Clustering algorithms

Dalya Baron
Carnegie Observatories

*Vatican Observatory Summer School on Big Data and
Machine Learning 2023 (VOSS-2023)*

Clustering is a key process in data exploration

- ❖ Clustering is one of the first steps in data exploration. Using clustering, we may try to answer one of the most basic questions we can ask — “what is there in my dataset?”.

Clustering is a key process in data exploration

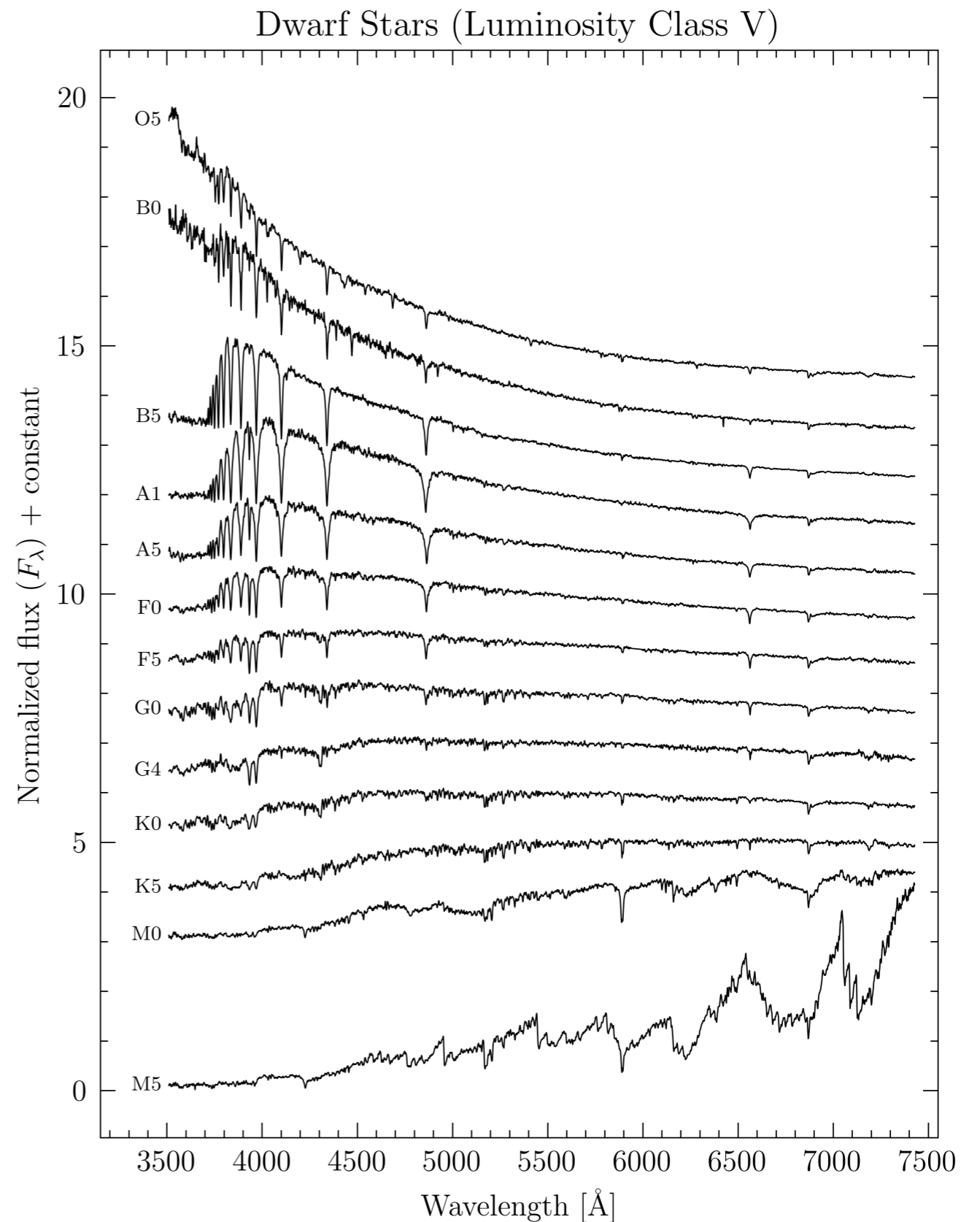
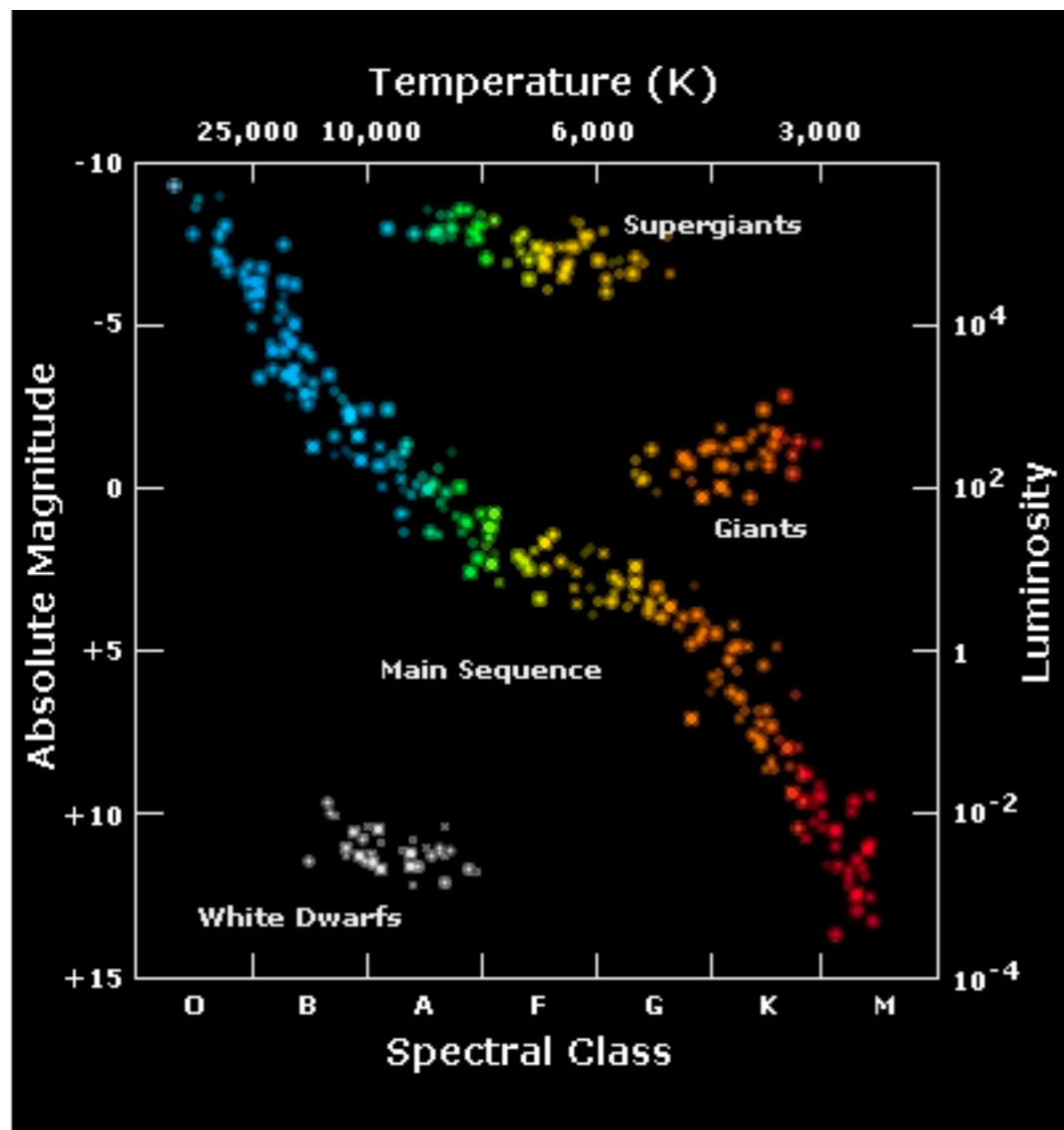
- ❖ Clustering is one of the first steps in data exploration. Using clustering, we may try to answer one of the most basic questions we can ask — “what is there in my dataset?”.
- ❖ Clustering is the task of grouping objects in the sample, such that objects in the same group are more “similar” to each other than to objects in other groups.

Clustering is a key process in data exploration

- ❖ Clustering is one of the first steps in data exploration. Using clustering, we may try to answer one of the most basic questions we can ask — “what is there in my dataset?”.
- ❖ Clustering is the task of grouping objects in the sample, such that objects in the same group are more “similar” to each other than to objects in other groups.
- ❖ Scientists, and in particular astronomers, have been doing cluster analysis well before they used programming or Machine Learning algorithms.

Clusters in astronomy

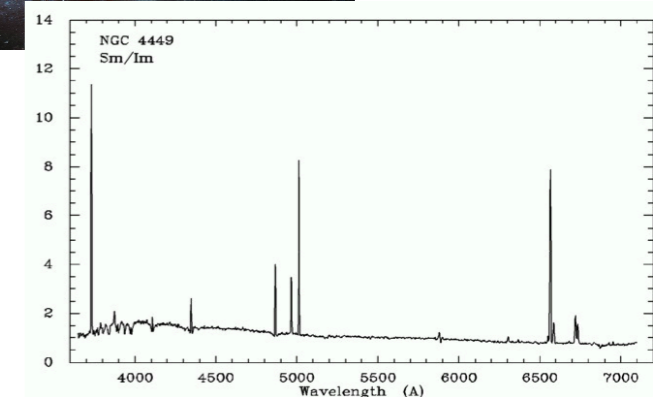
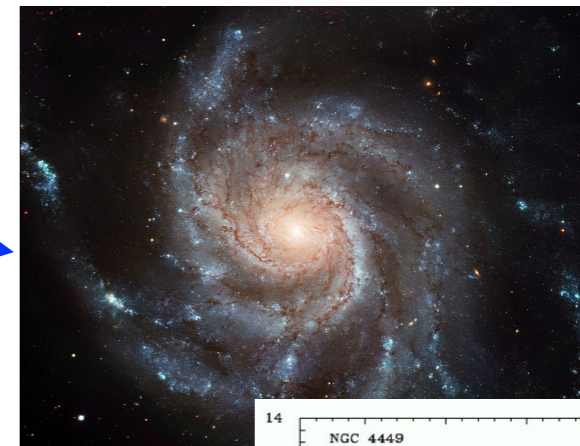
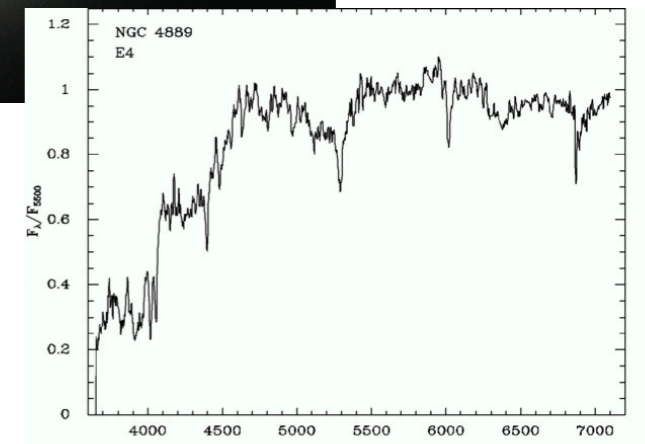
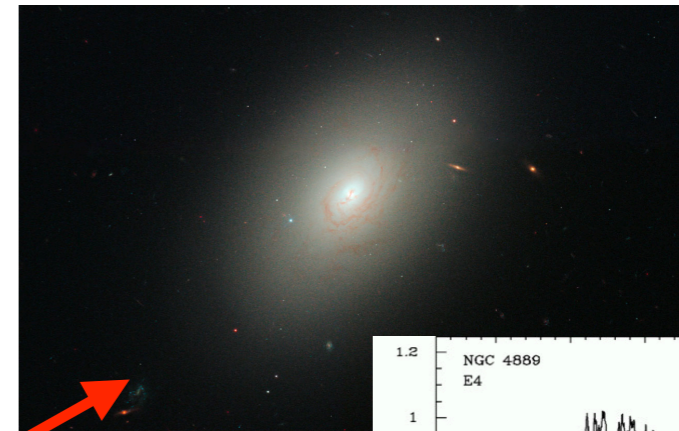
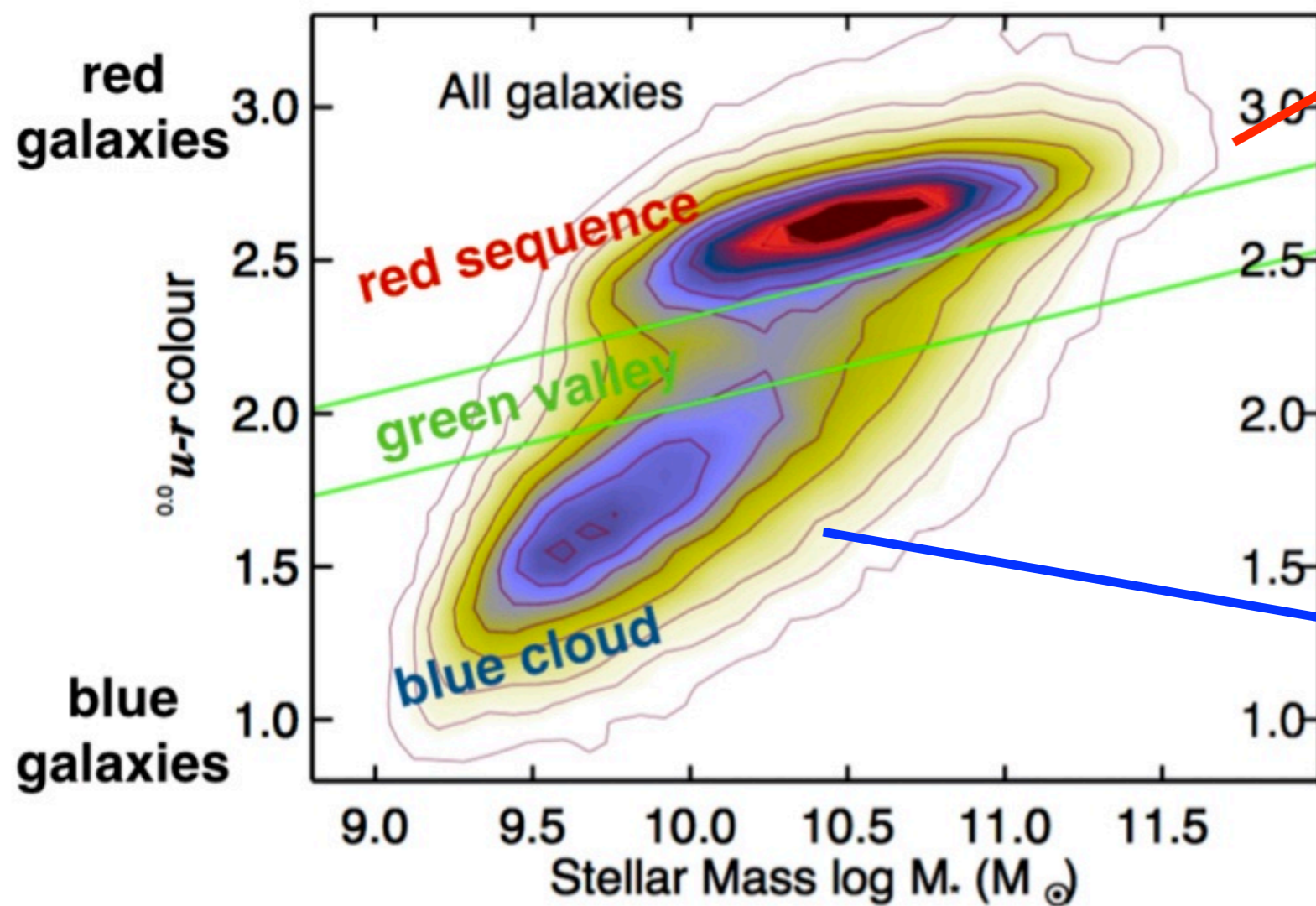
1. Stellar spectral classes



Clusters in astronomy

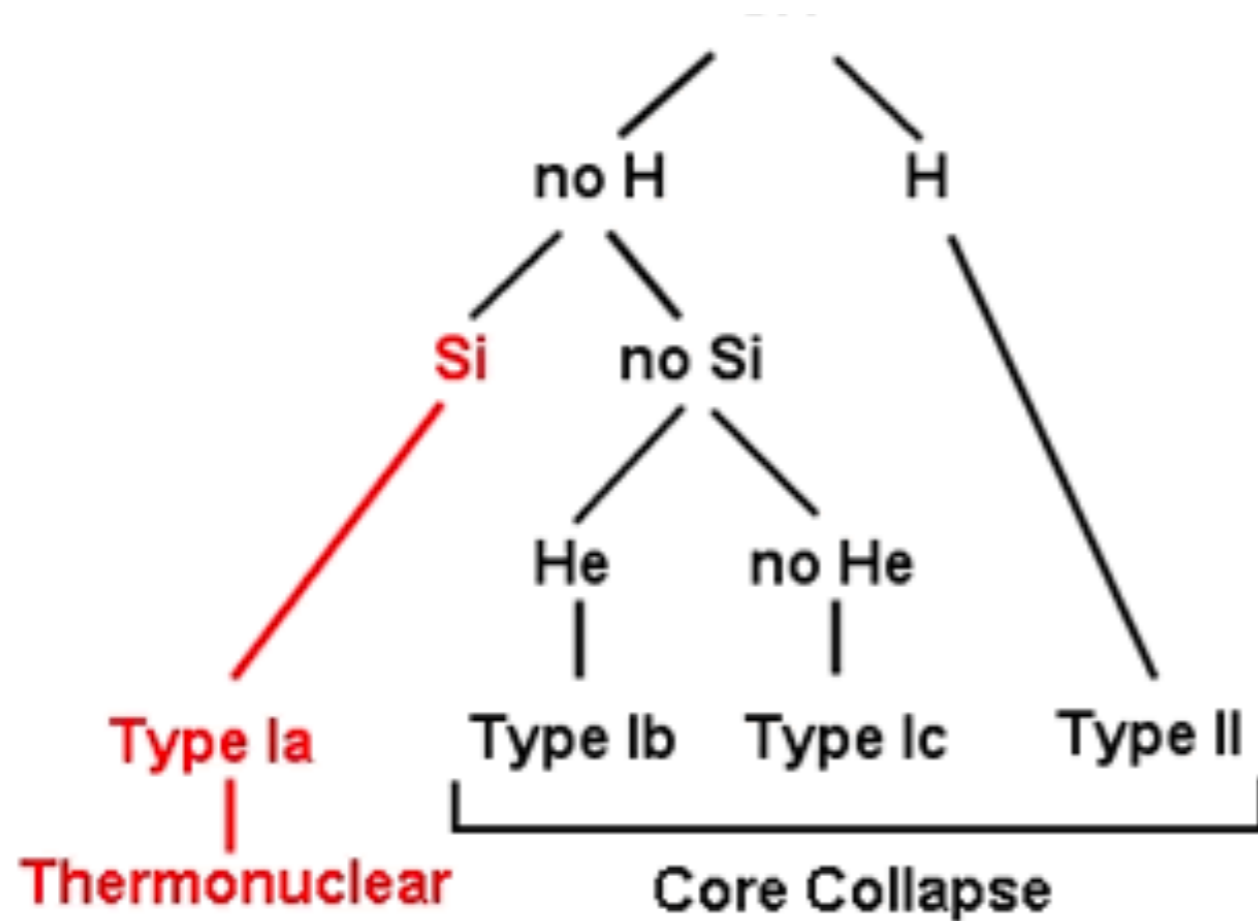
2. Galaxy bimodality

Taken from: Schawinski et al. (2014)

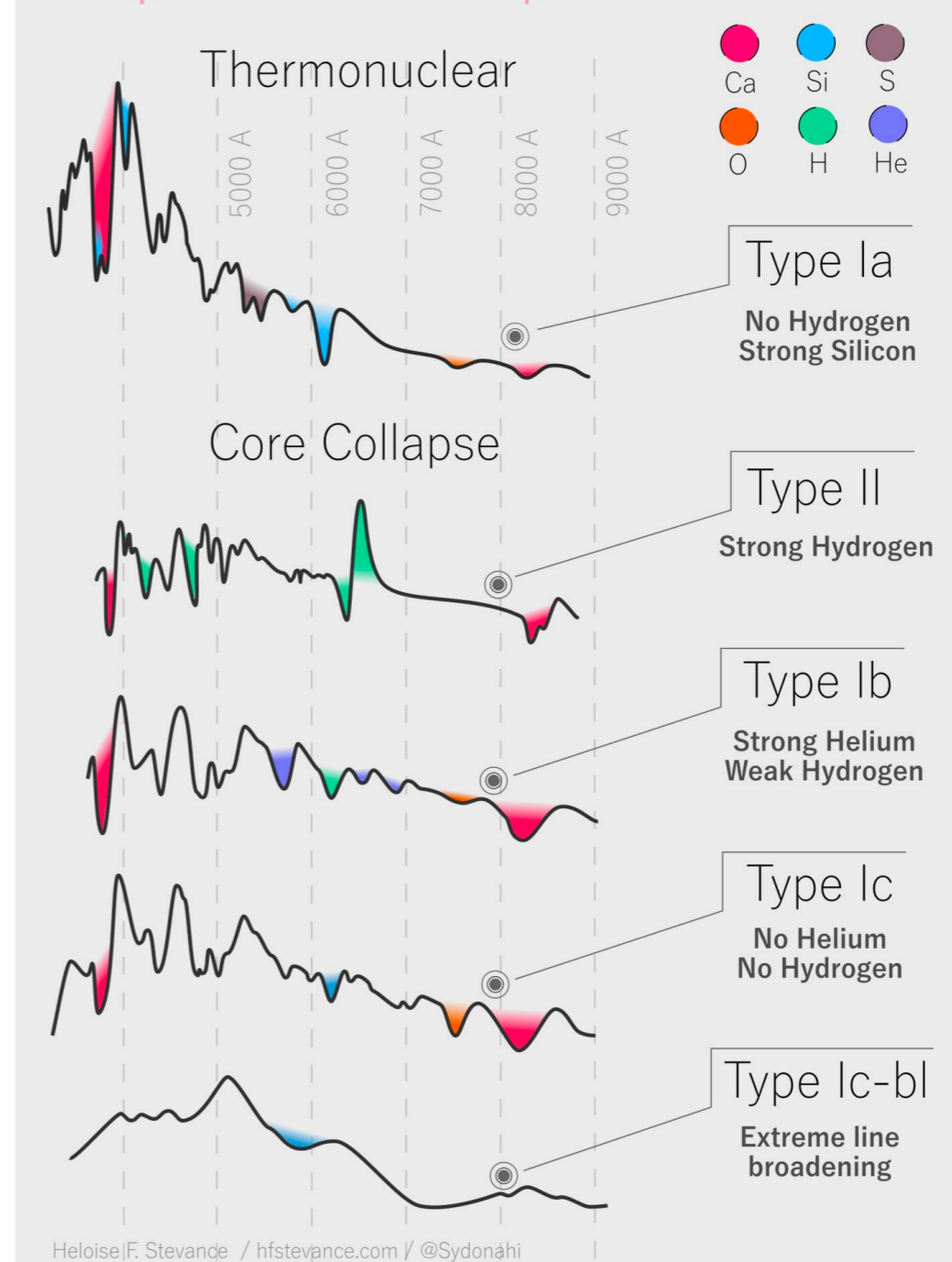


Clusters in astronomy

3. Supernova classes: type Ia and type II supernovae



Supernova Spectra

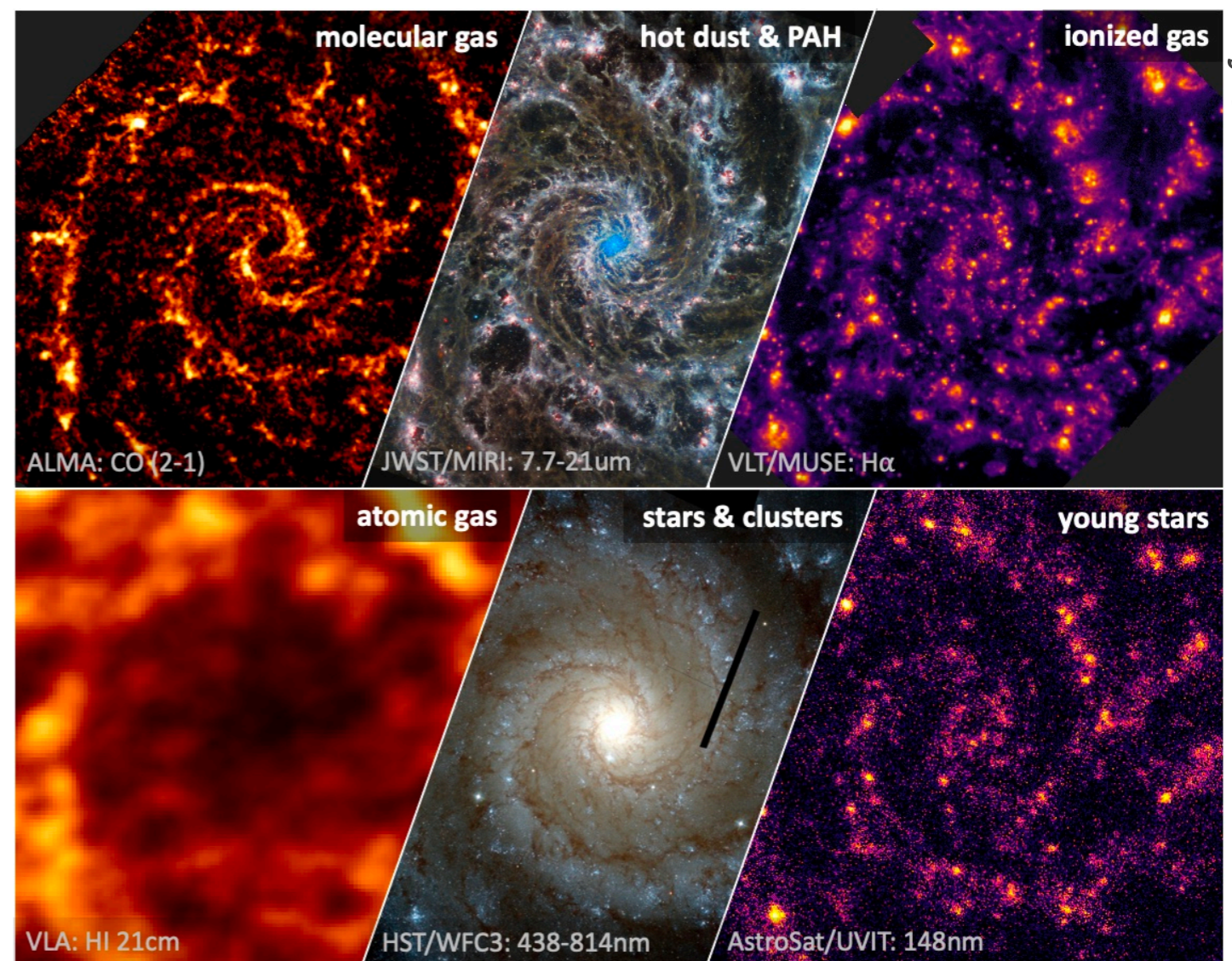


What is fundamentally different now?

1. The volume and rate of information grows exponentially:
 - Sky survey now generate ~ 1 PB of data + derived products. Astronomical databases now routinely include $10^6 - 10^9$ objects.
 - We can no longer manually-inspect all the data we collect.

What is fundamentally different now?

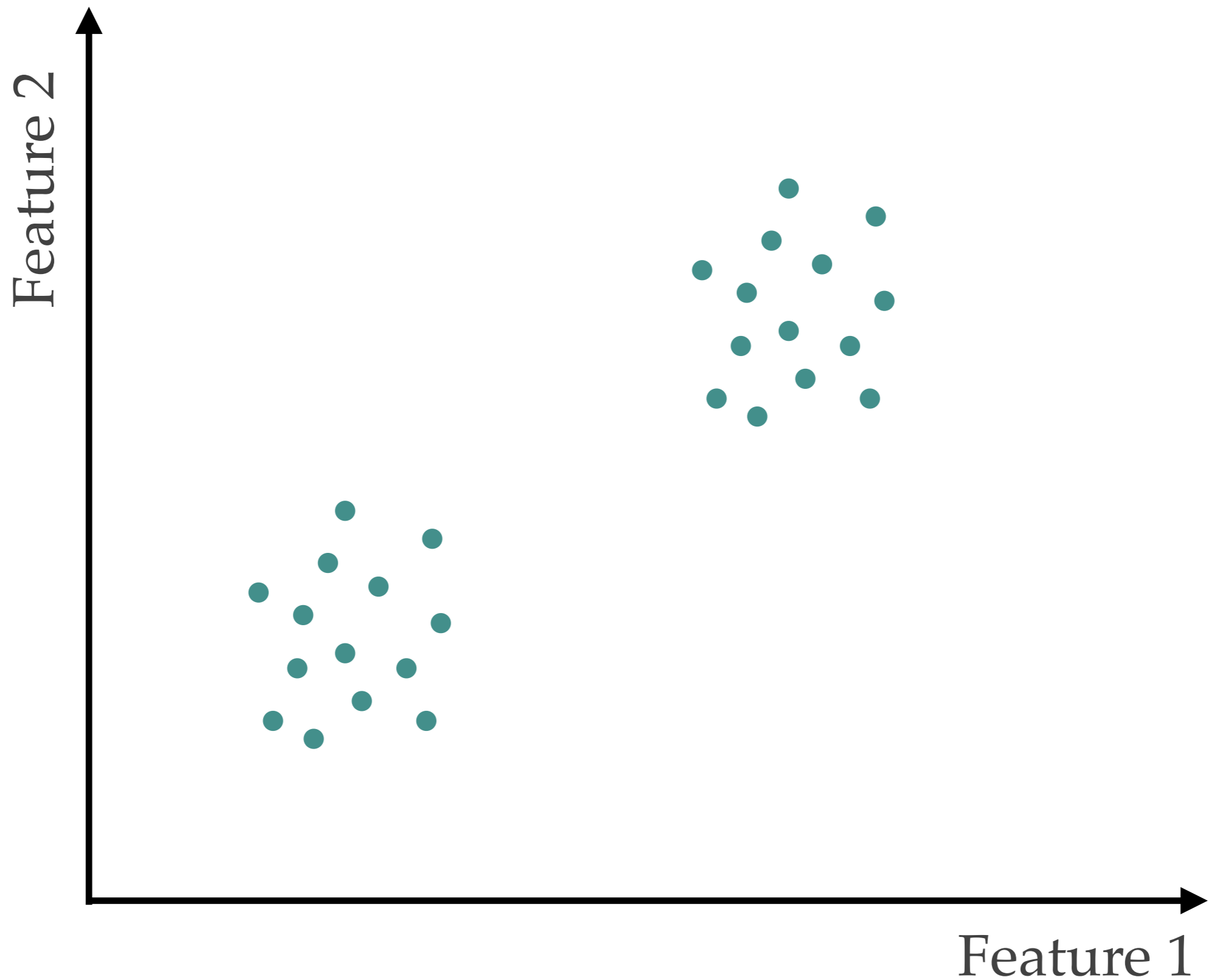
2. A great increase in data dimensionality and complexity:
- Data is heterogeneous and high-dimensional.
 - Patterns and correlations in the data can no longer be visualized in 3D.



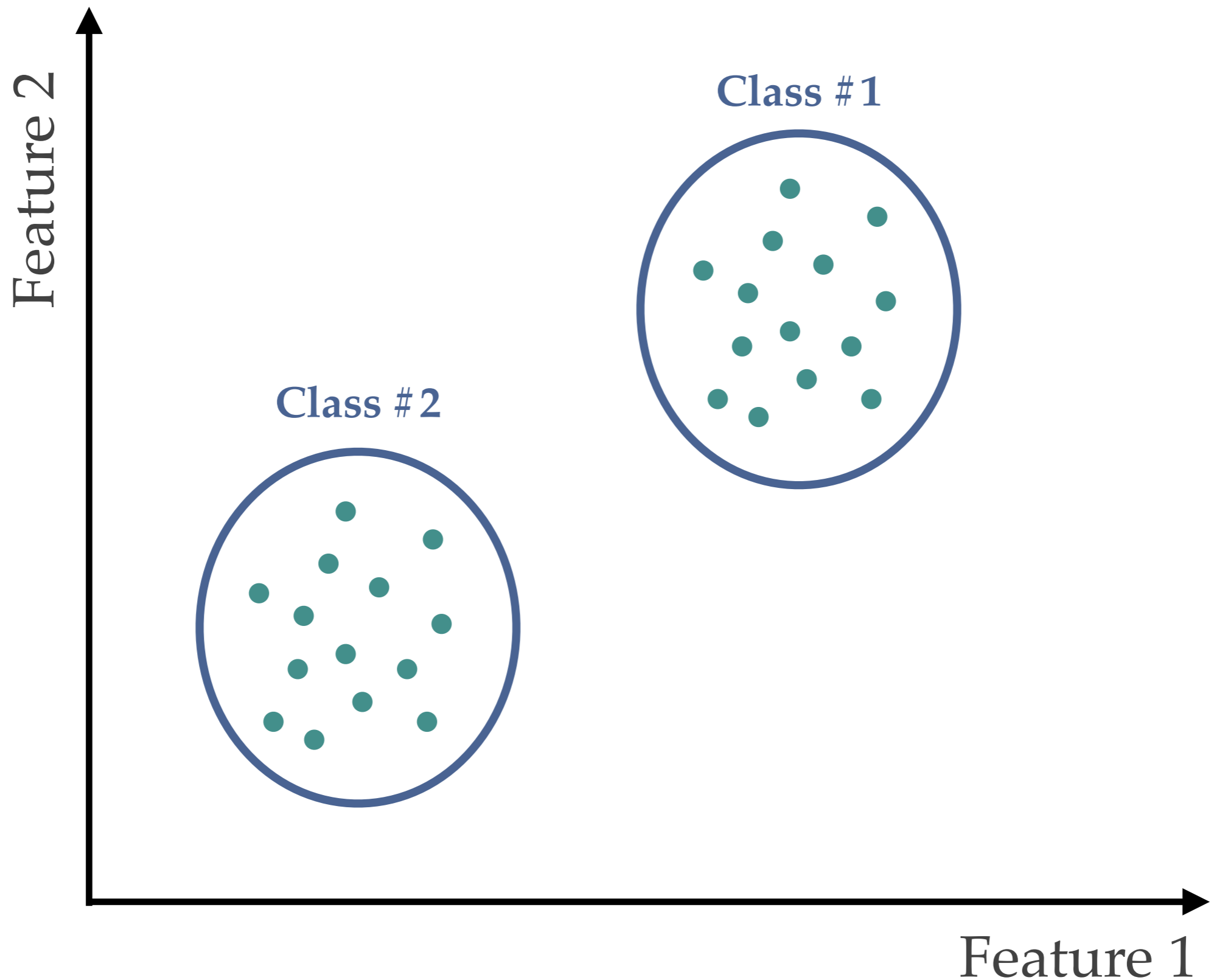
By J. Sun

Image by the PHANGS collaboration

Clustering



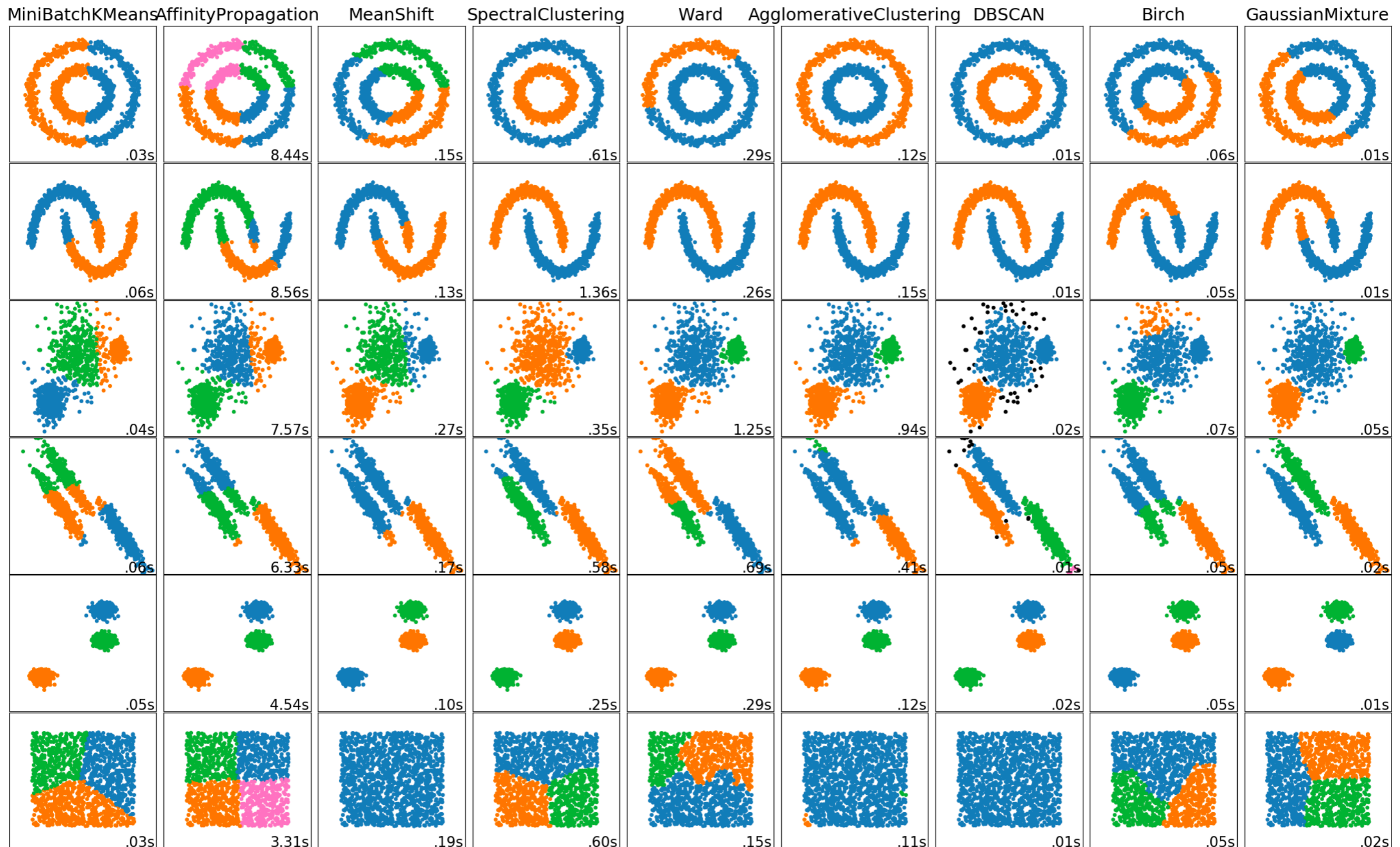
Clustering



Clustering

From Scikit-learn's example gallery:

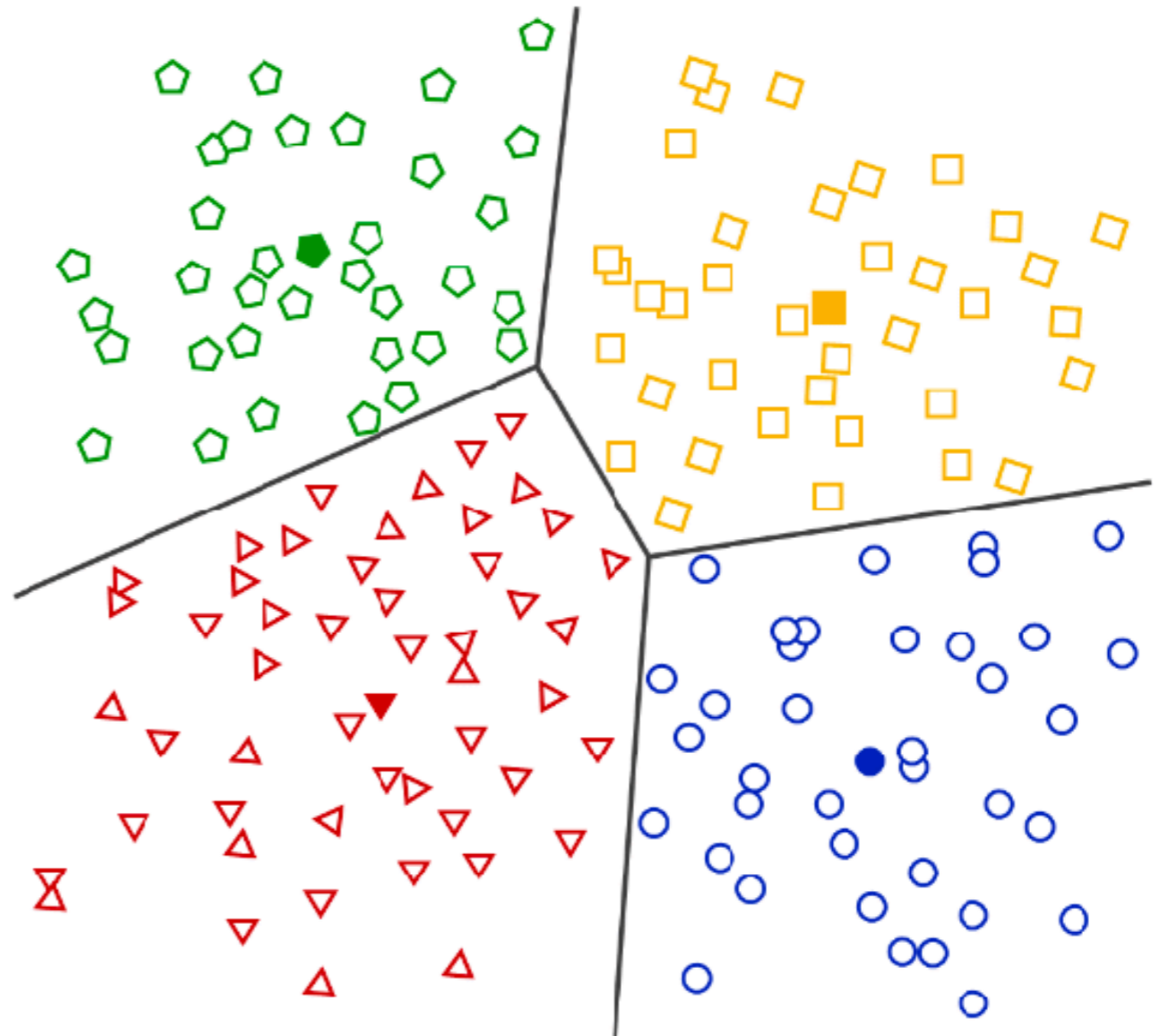
see [this](#) comparison between algorithms



Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

1. **Centroid-based / Partition-based clustering (e.g., K-means):** algorithms that divide the data into non-hierarchical clusters by defining “cluster centers”.

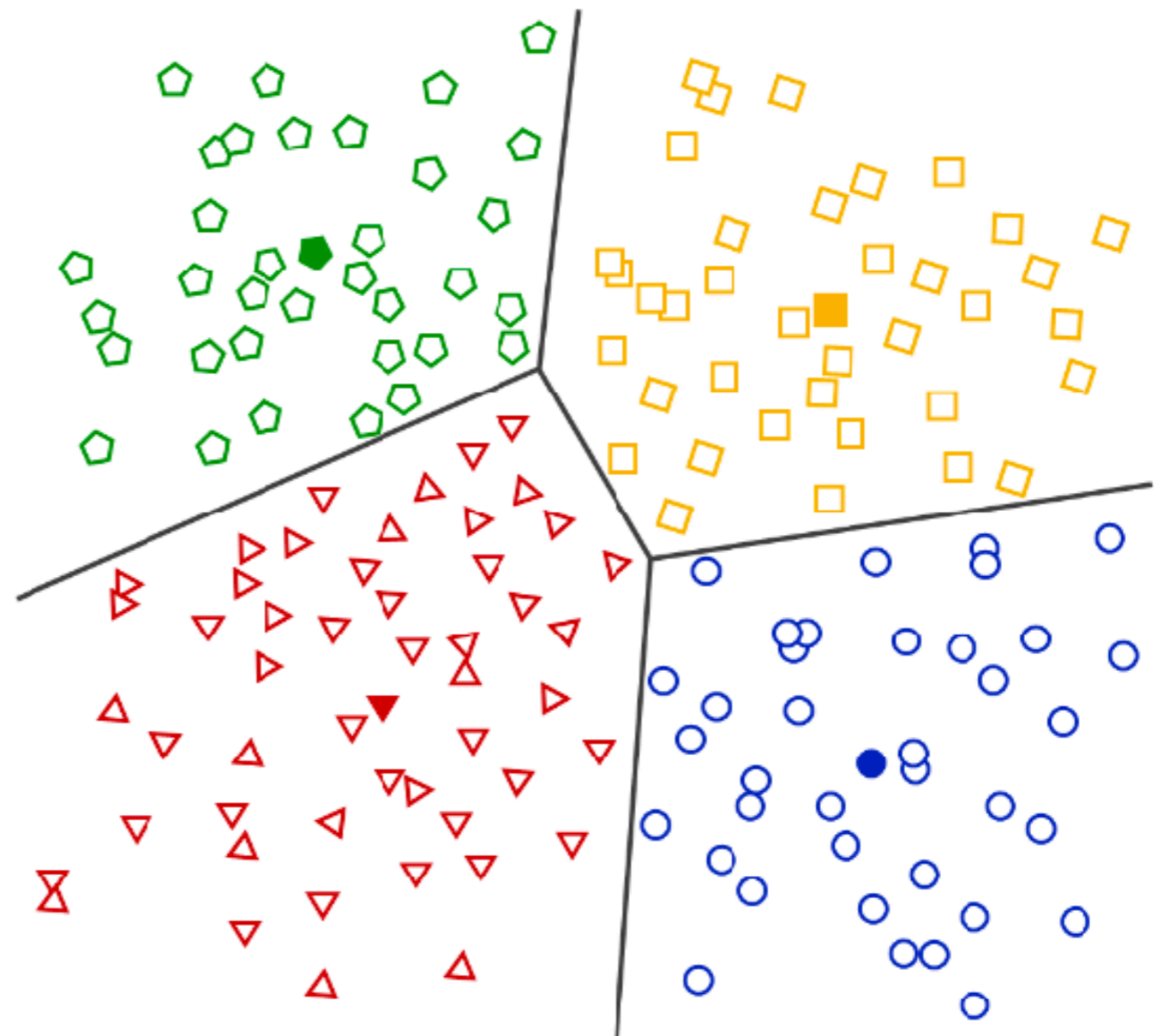


Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

1. Centroid-based / Partition-based clustering (e.g., K-means): algorithms that divide the data into non-hierarchical clusters by defining “cluster centers”.

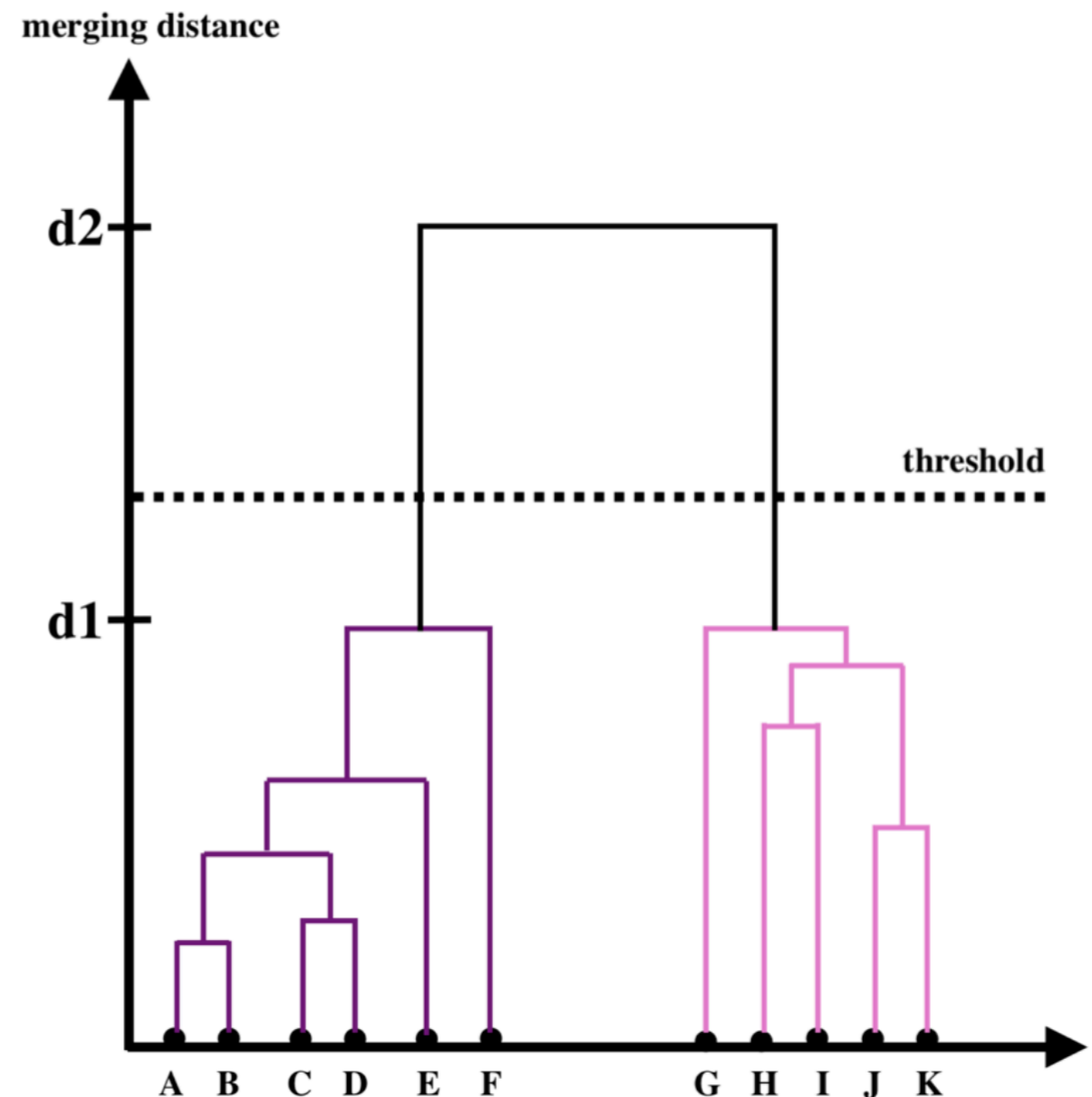
- Scales well with number of **samples** and number of **features**.
- Sensitive to initial conditions (may get stuck in a local minimum) and outliers.
- Even cluster size, flat geometry.



Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

2. Hierarchical clustering: algorithms that create tree of clusters by merging close clusters into larger clusters.

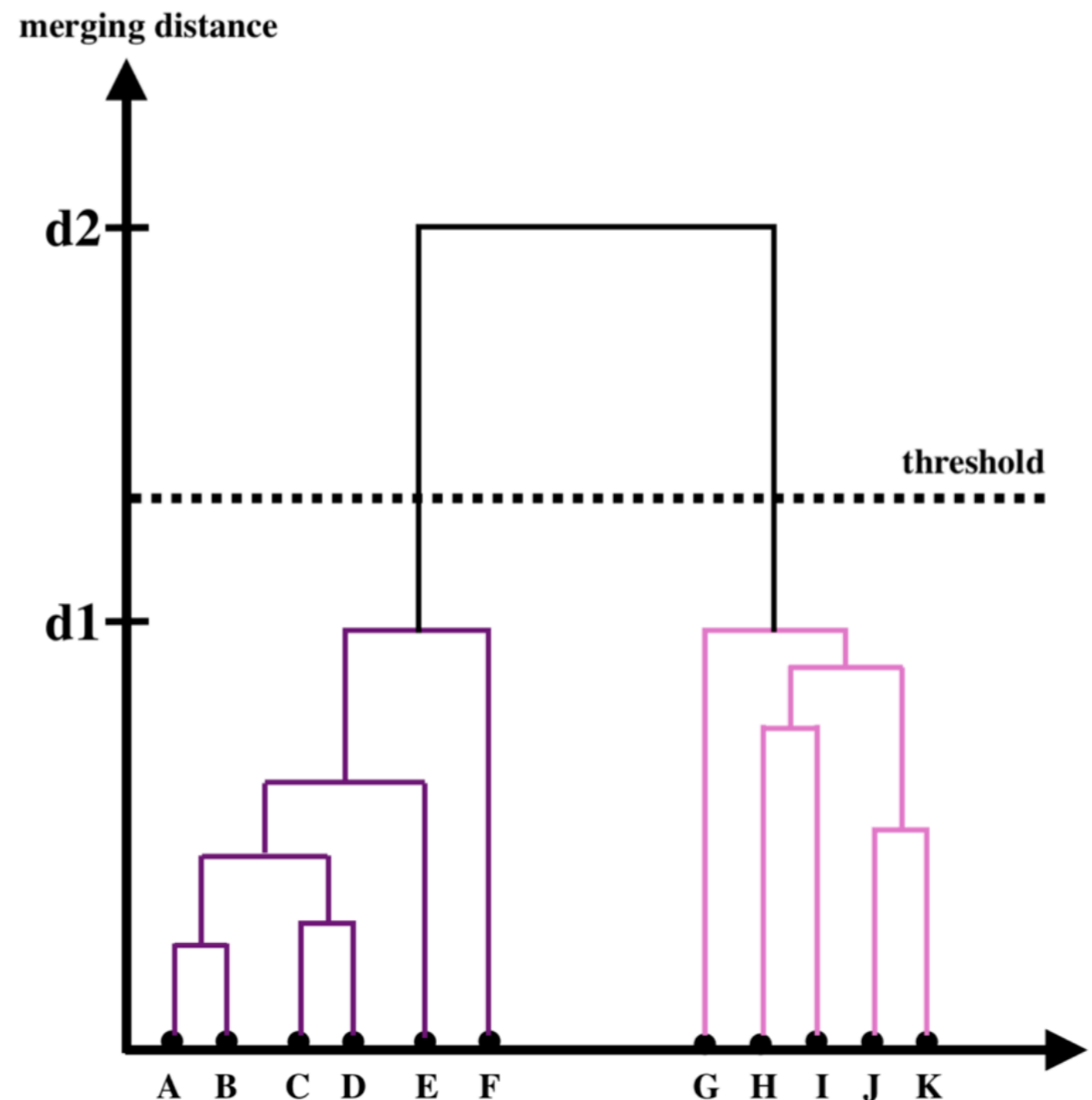


Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

2. Hierarchical clustering: algorithms that create tree of clusters by merging close clusters into larger clusters.

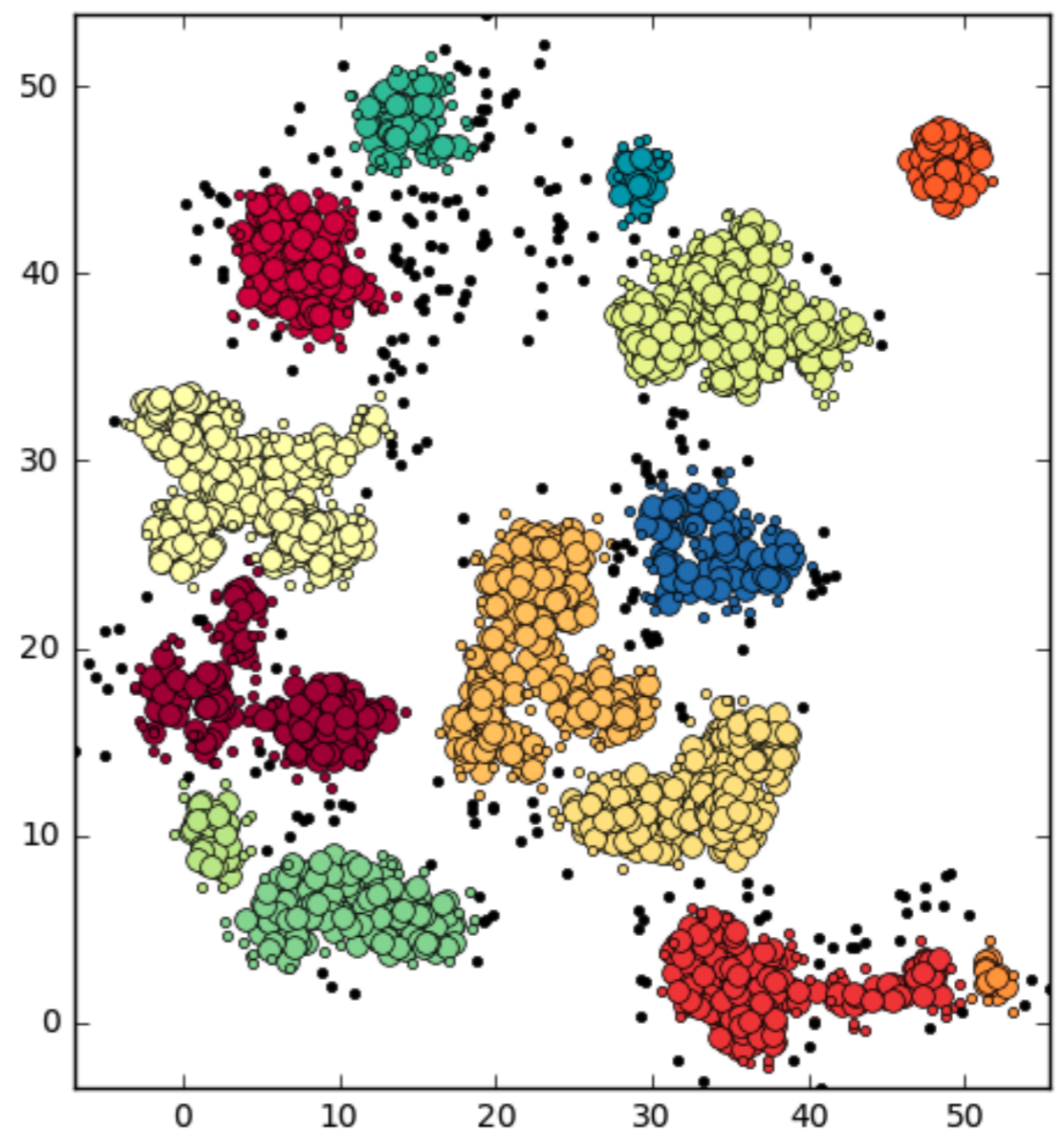
- Typical examples: BIRCH, CURE, ROCK, and Chameleon.
- Scales well with number of **samples** and number of **features**.
- Can work with many clusters, non-even cluster sizes.



Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

3. **Density-based clustering:** high density regions in the data space are considered to belong to the same cluster.



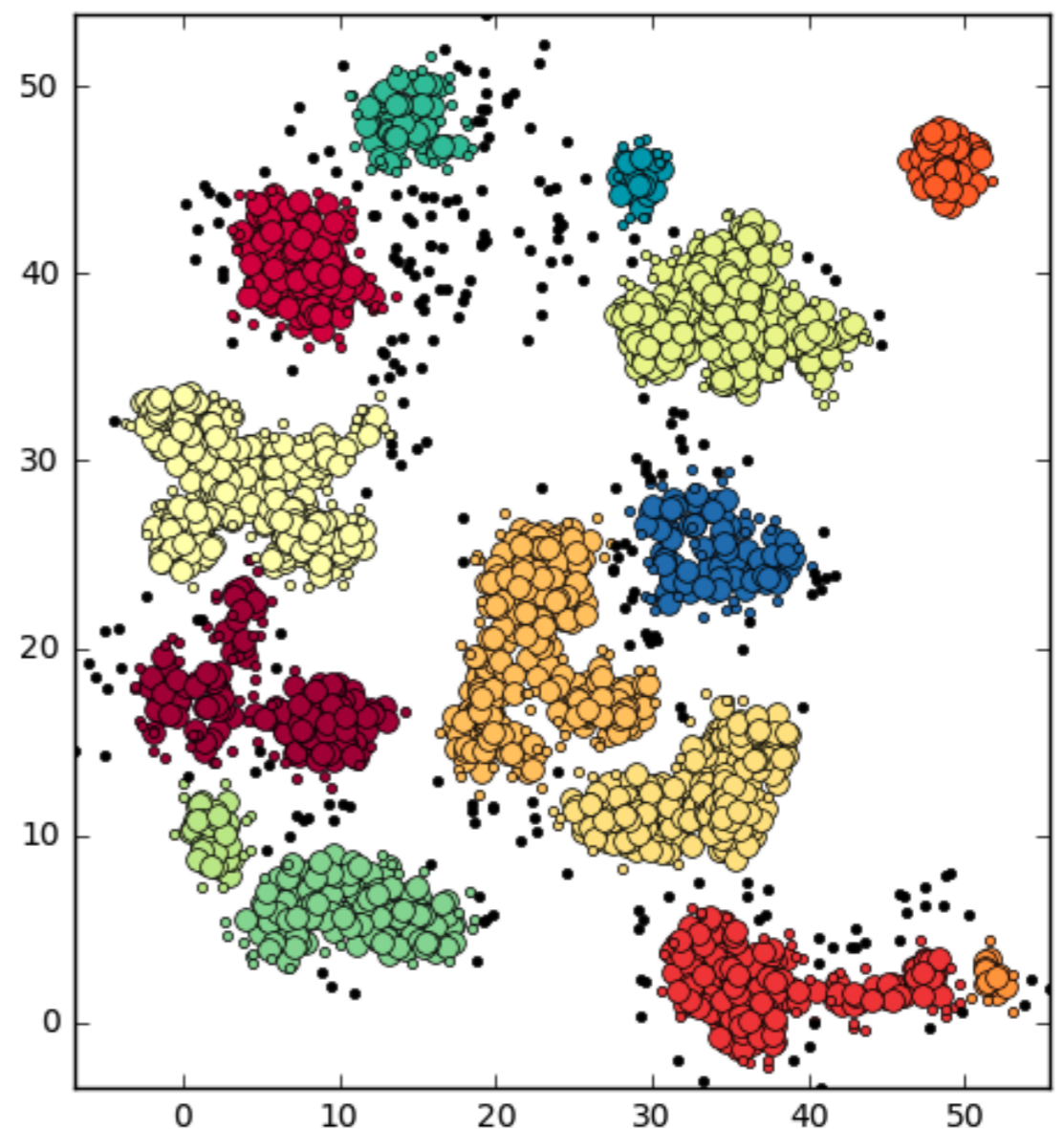
taken from [here](#).

Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

3. **Density-based clustering:** high density regions in the data space are considered to belong to the same cluster.

- Typical examples: DBSCAN, OPTICS, and Mean-shift.
- Scales well with number of **samples**. Not so well with number of **features**.
- Uneven cluster sizes, non-flat geometries. **Marks outliers**.
- Struggles with data with varying density.



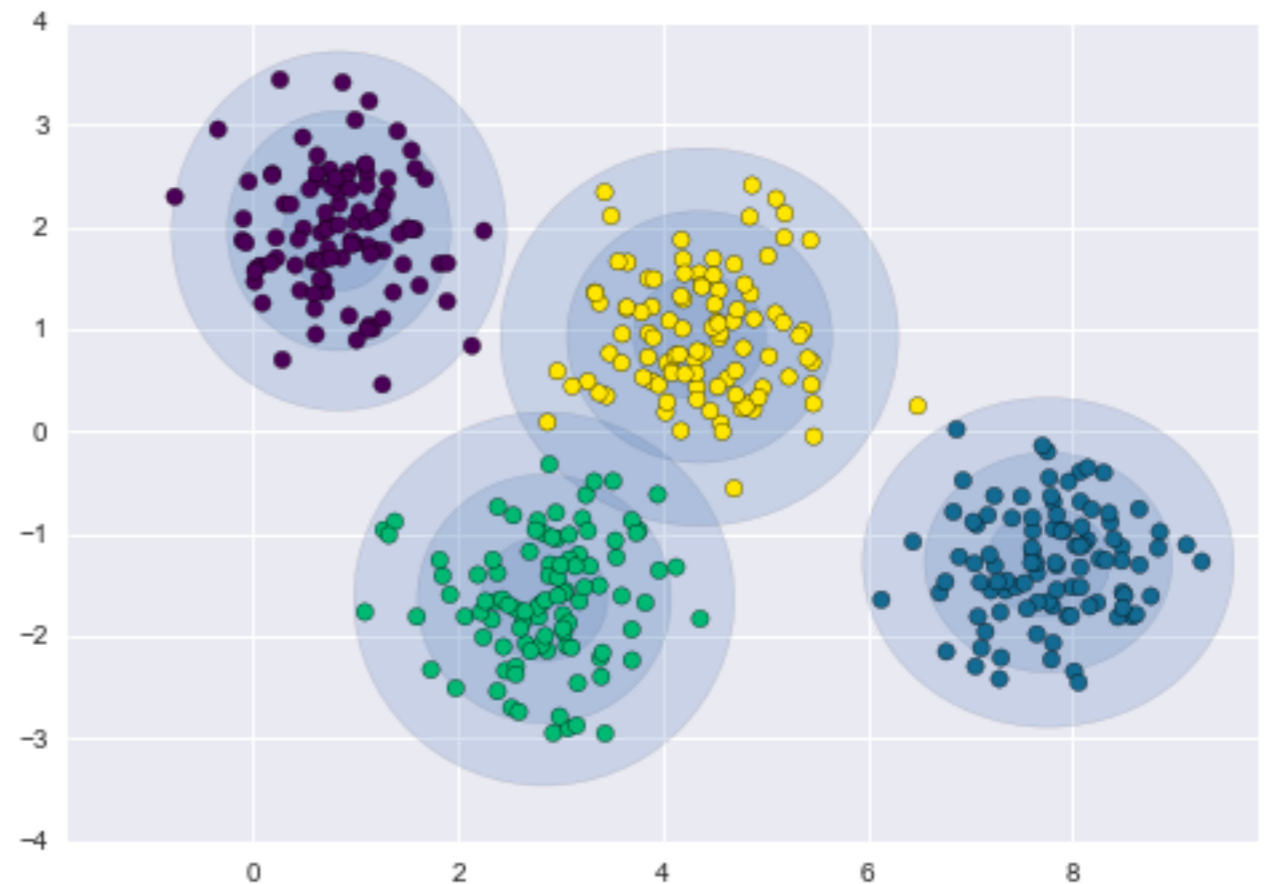
taken from [here](#).

Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

4. **Distribution-based clustering:** algorithm assumes that data is composed of distributions. Same cluster's data points need to belong to the same probability distribution.

Credit: Jake VanderPlas.
Tutorial available [here](#).



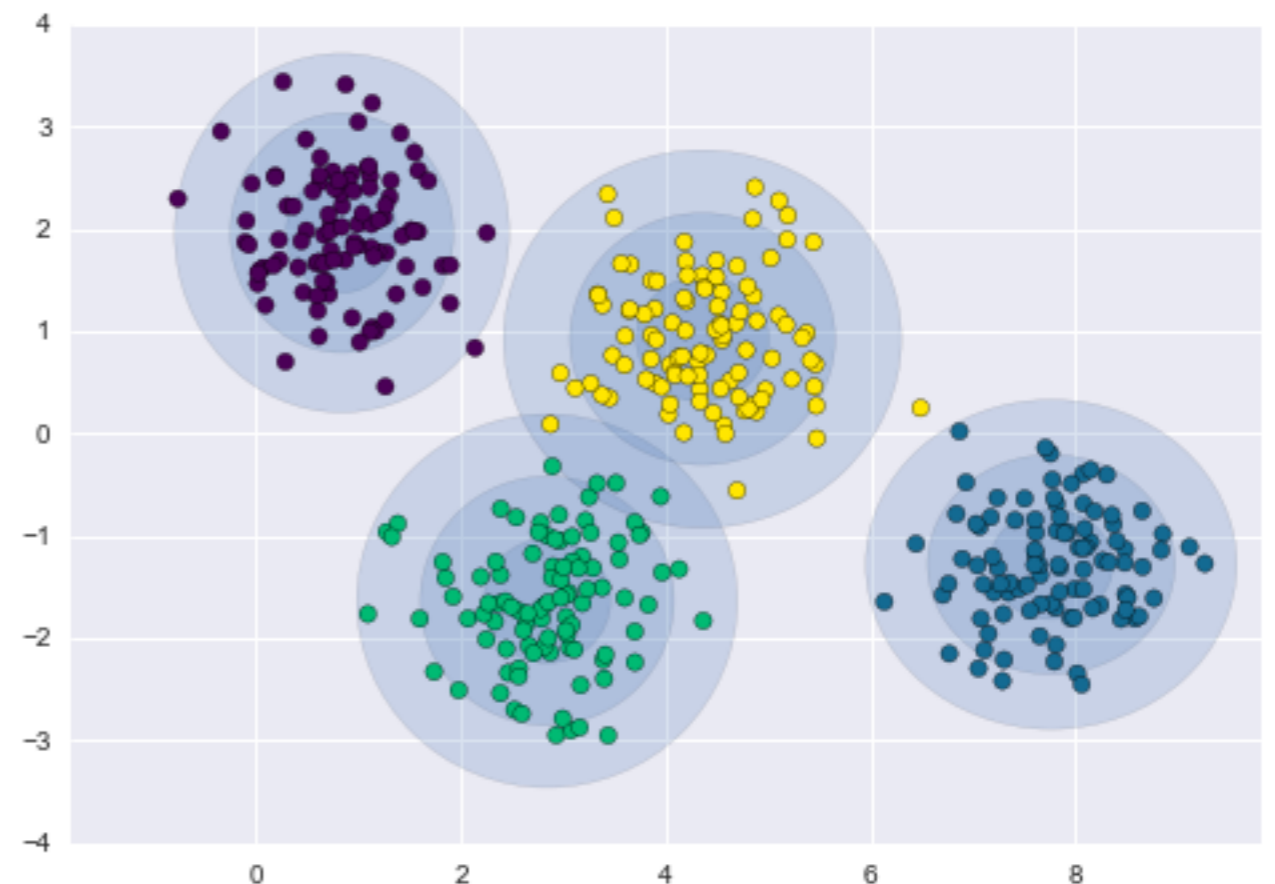
Different types of clustering algorithms

A review paper about classical and modern clustering algorithms can be found [here](#).

4. **Distribution-based clustering:** algorithm assumes that data is composed of distributions. Same cluster's data points need to belong to the same probability distribution.

- Typical examples: DBCLASD and GMM. **Need to assume the distribution.**
- Does not scale well with the number of samples or features.
- Flat geometry.
- Assigns probabilities for every point.

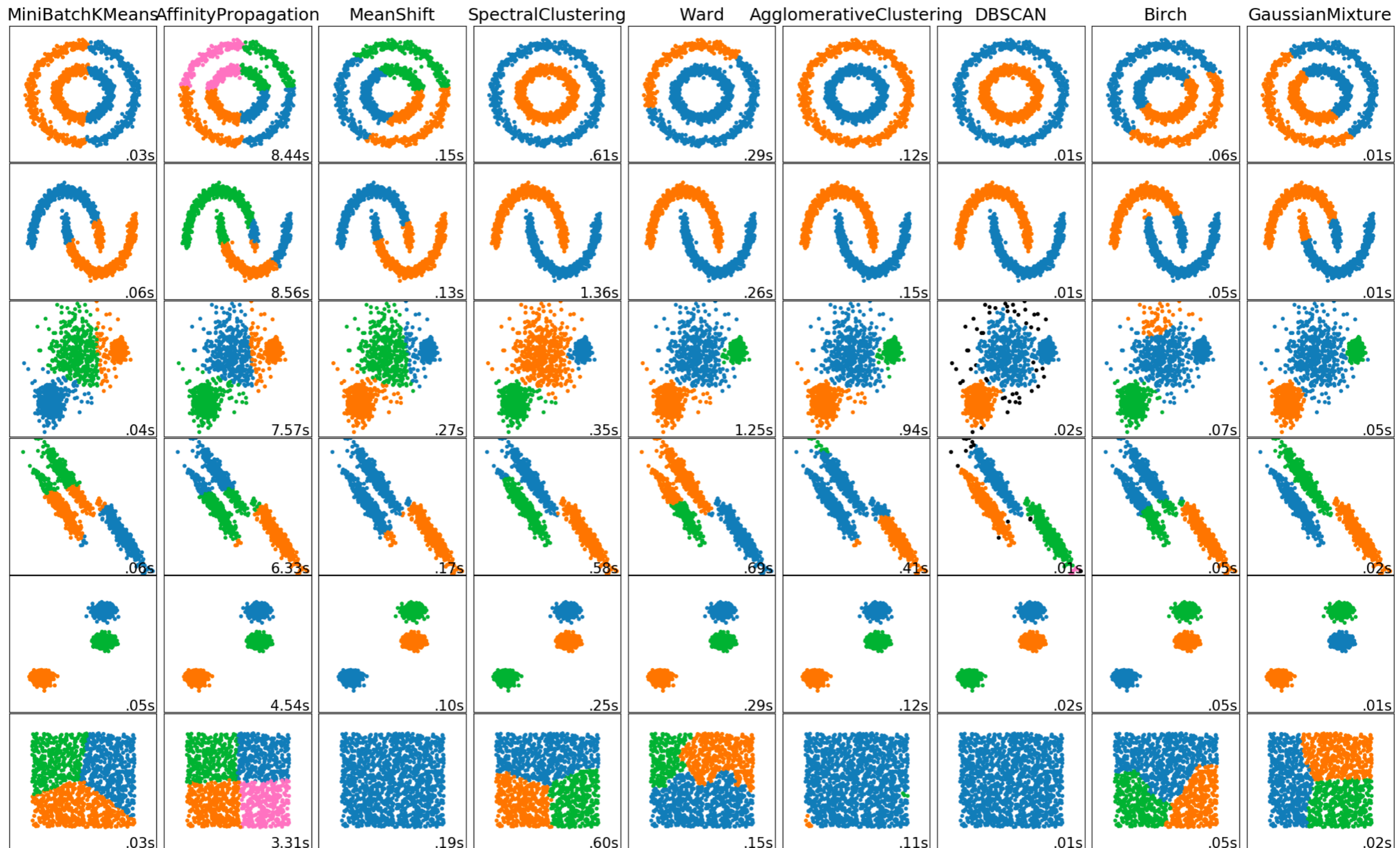
Credit: Jake VanderPlas.
Tutorial available [here](#).



How should I choose which algorithm to use?

From Scikit-learn's example gallery:

see [this](#) comparison between algorithms



K-means

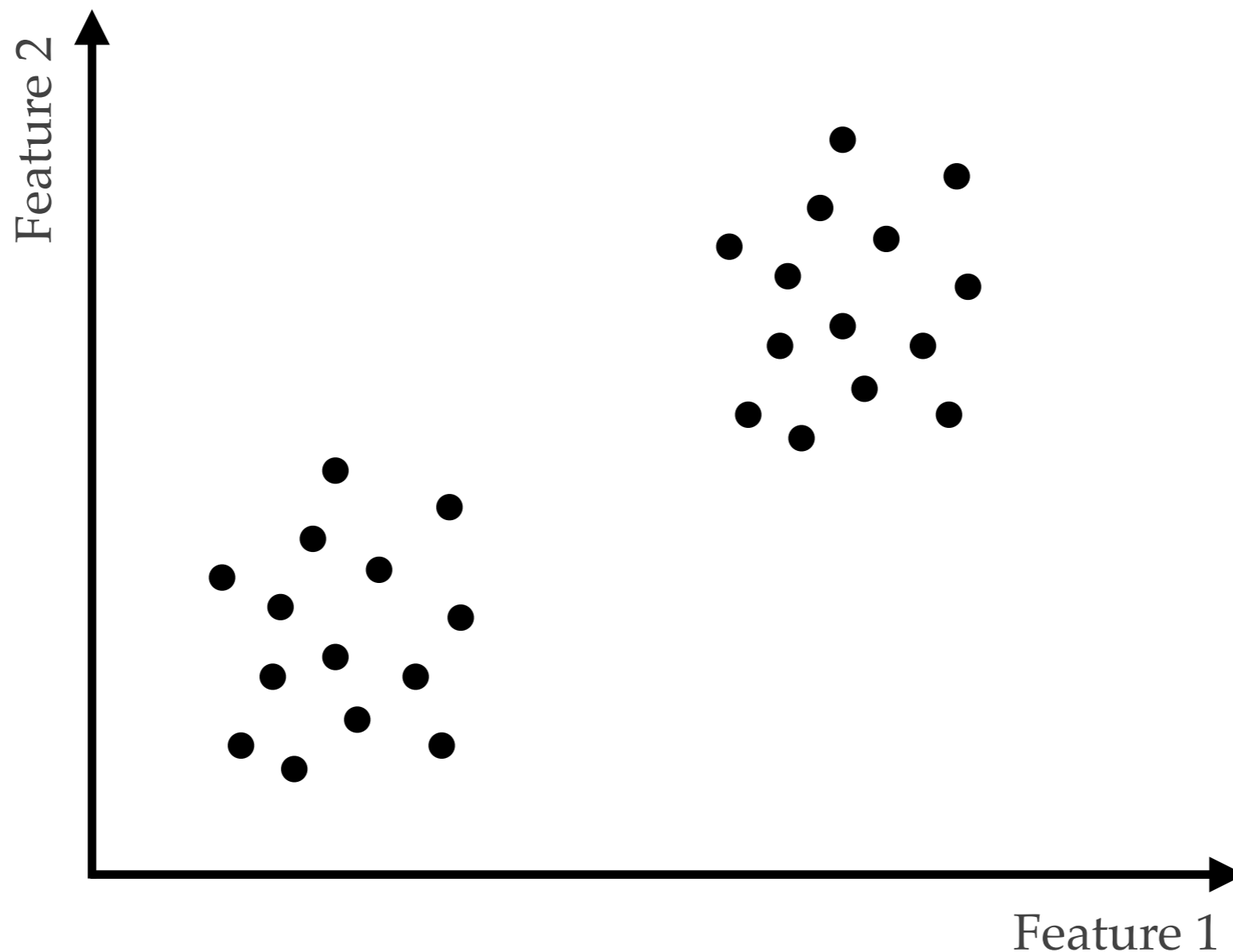
Input: measured features, and the number of clusters, k .

The algorithm will classify **all** the objects in the sample into k clusters.



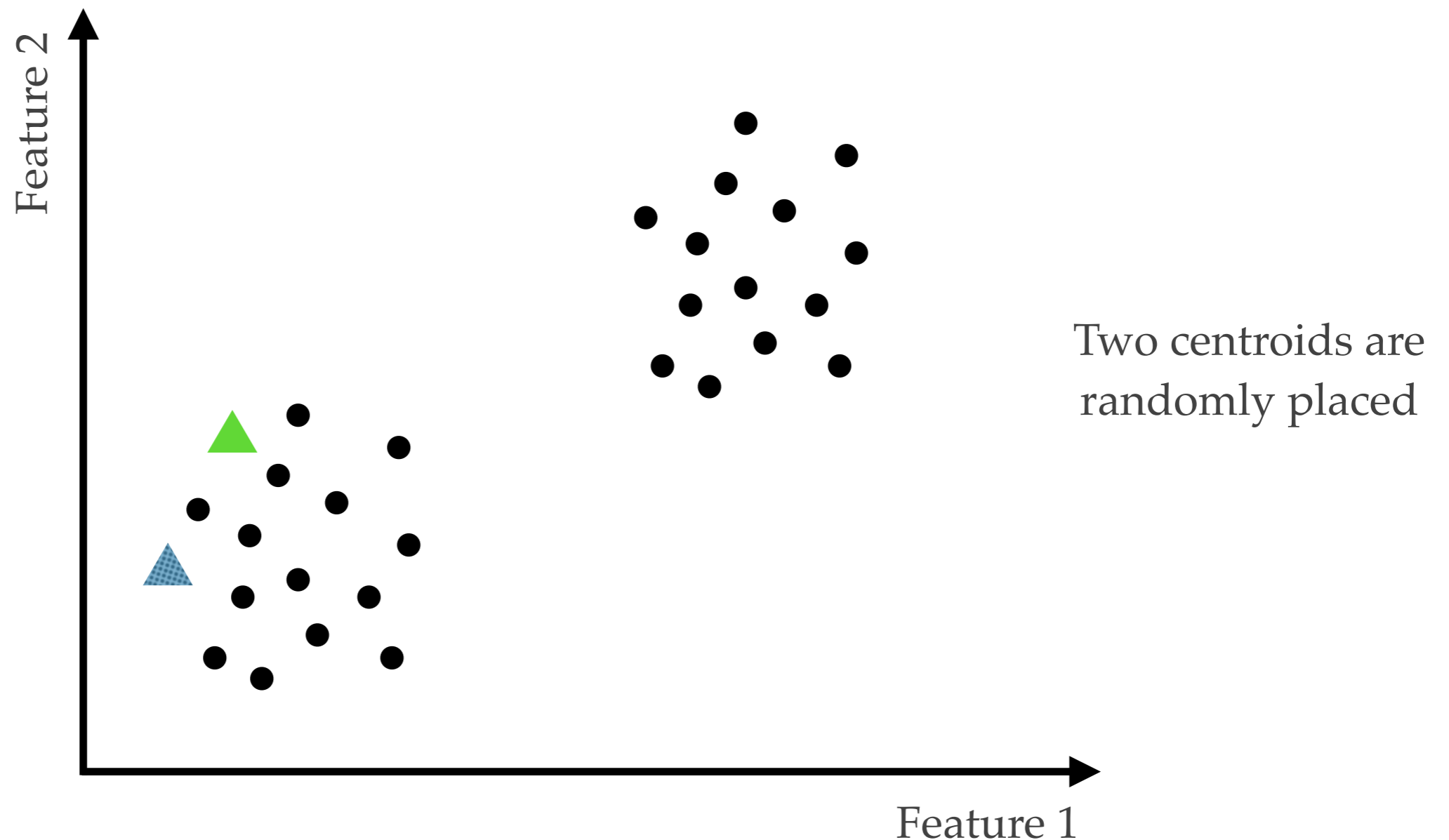
K-means

- (I) Random assignment of **k** points that represent the centroids of the clusters.
- Iterate:
- (II) Associate each object with a single cluster, using the **distance** from the cluster centroid.
 - (III) Recalculate the cluster centroid according to the objects that are associated with it.



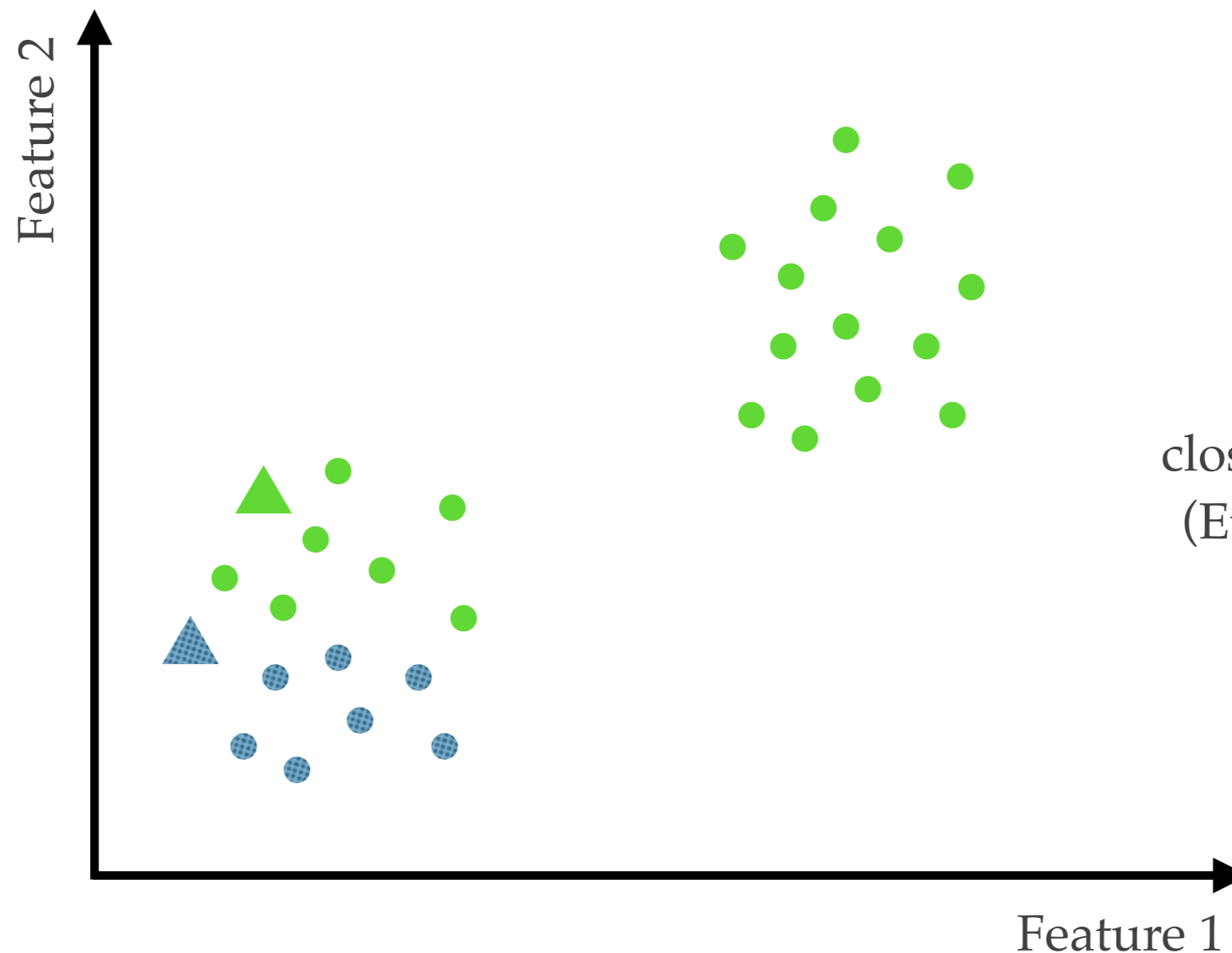
K-means

- (I) Random assignment of **k** points that represent the centroids of the clusters.
- Iterate:
- (II) Associate each object with a single cluster, using the **distance** from the cluster centroid.
 - (III) Recalculate the cluster centroid according to the objects that are associated with it.



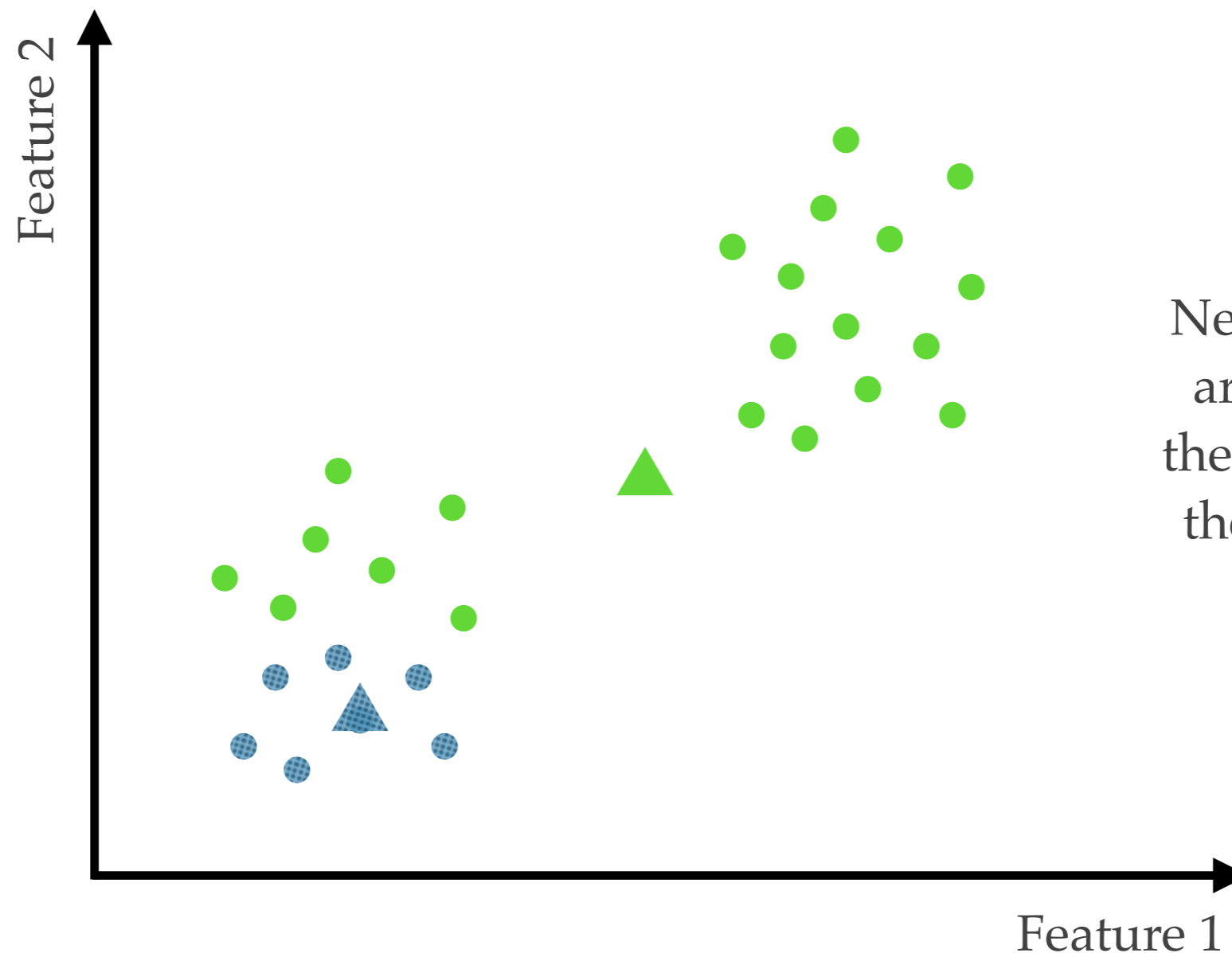
K-means

- (I) Random assignment of **k** points that represent the centroids of the clusters.
- Iterate:
- (II) Associate each object with a single cluster, using the **distance** from the cluster centroid.
 - (III) Recalculate the cluster centroid according to the objects that are associated with it.



K-means

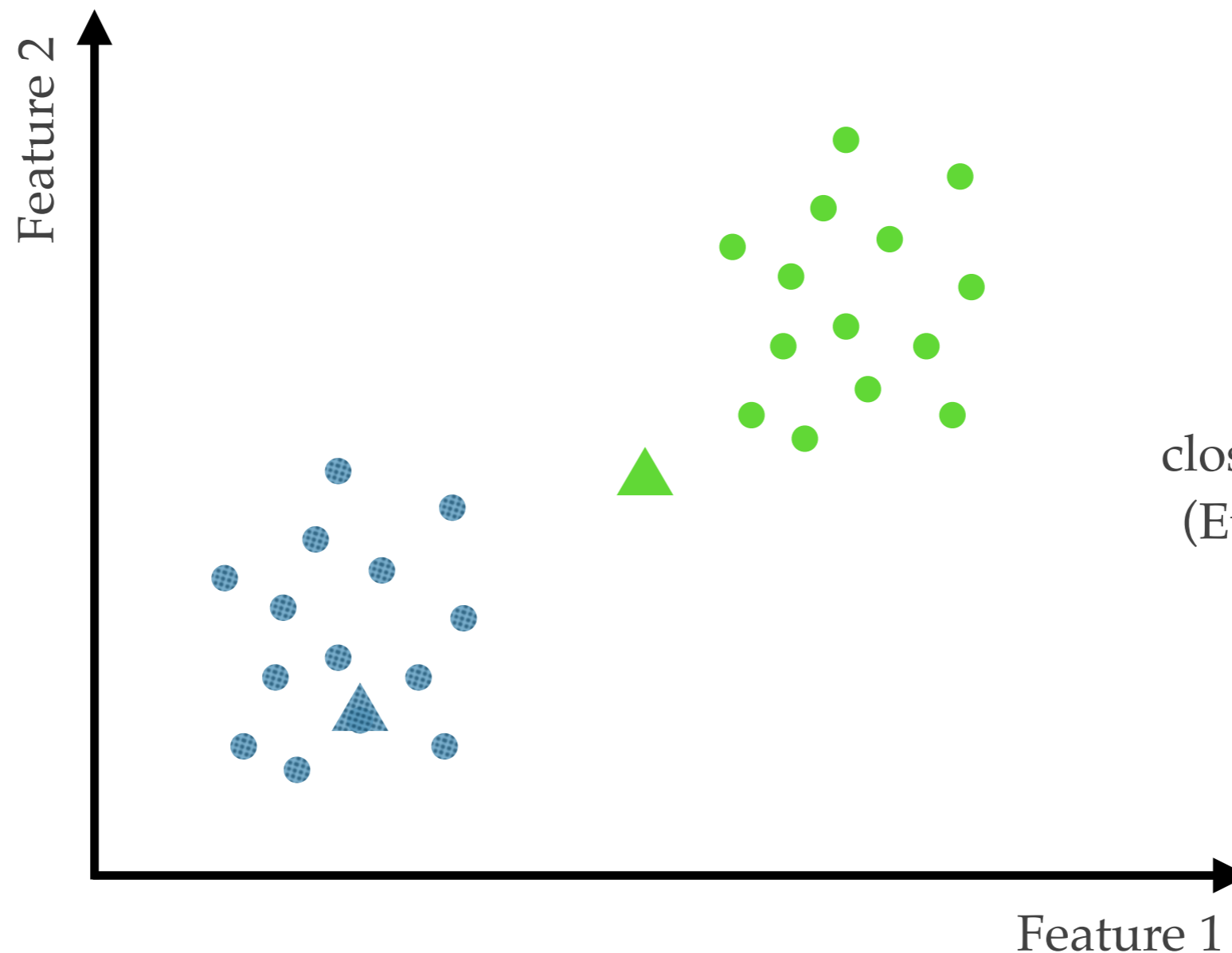
- (I) Random assignment of **k** points that represent the centroids of the clusters.
- Iterate:
- (II) Associate each object with a single cluster, using the **distance** from the cluster centroid.
 - (III) Recalculate the cluster centroid according to the objects that are associated with it.



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

K-means

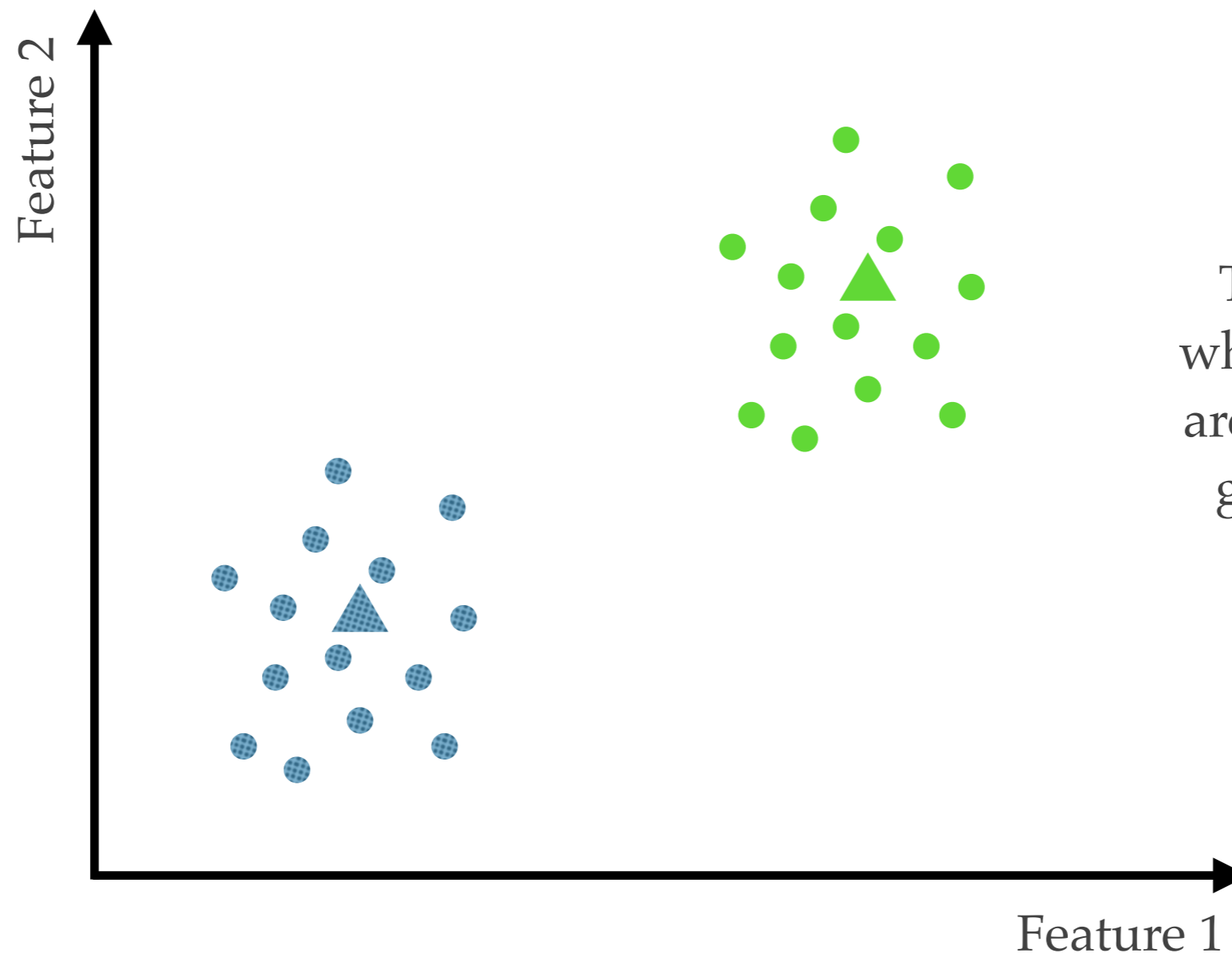
- (I) Random assignment of **k** points that represent the centroids of the clusters.
- Iterate:
- (II) Associate each object with a single cluster, using the **distance** from the cluster centroid.
 - (III) Recalculate the cluster centroid according to the objects that are associated with it.



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

K-means

- (I) Random assignment of **k** points that represent the centroids of the clusters.
- Iterate:
- (II) Associate each object with a single cluster, using the **distance** from the cluster centroid.
 - (III) Recalculate 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

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

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:

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

Annotations for the cost function J :

- cluster centroids**: points to μ_k
- cluster members**: points to x_i
- Euclidean distance**: points to the norm $||x_i - \mu_k||$

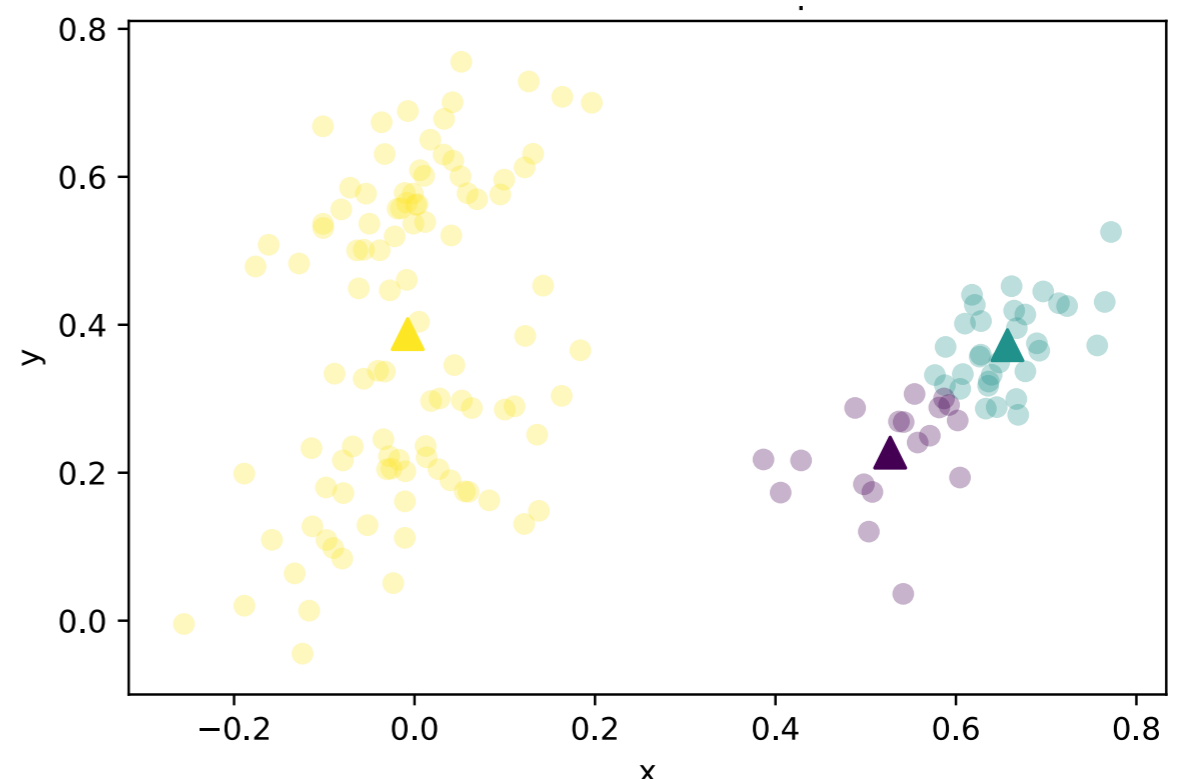
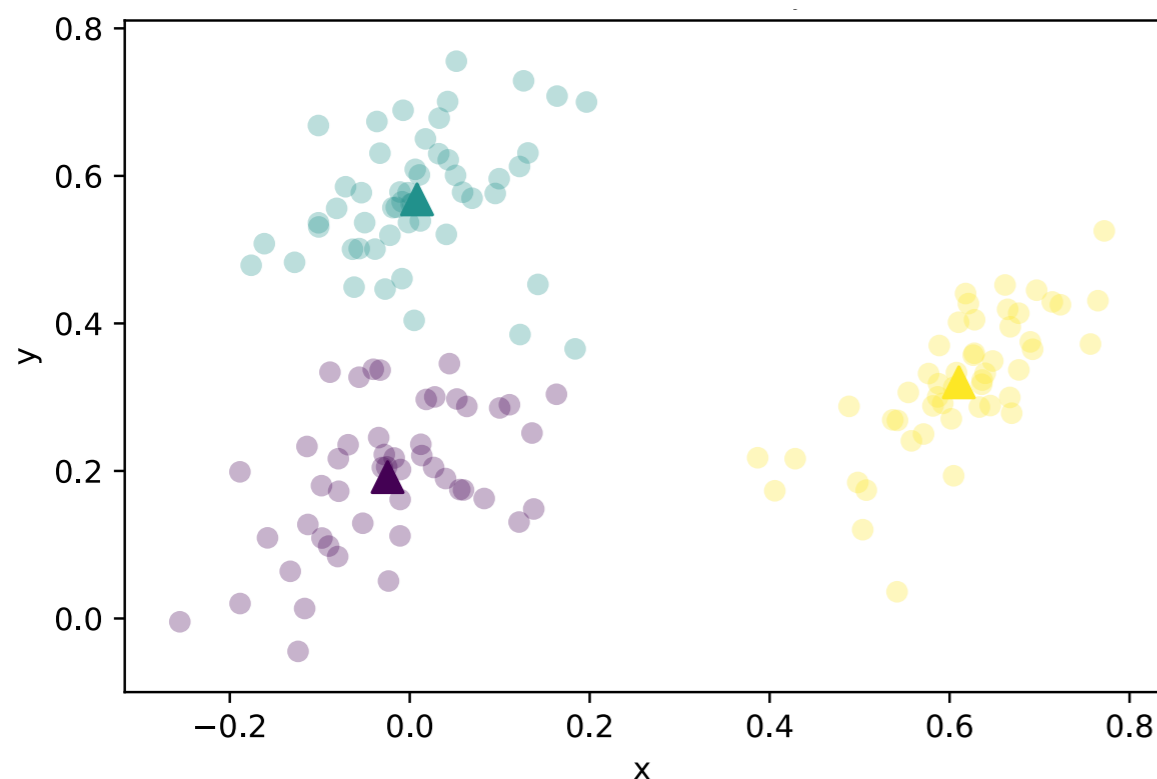
The anatomy of K-means

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

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:

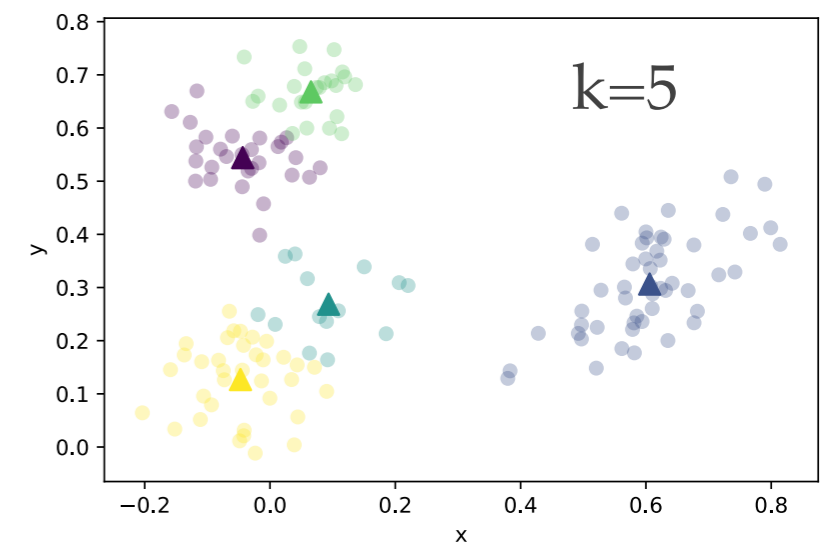
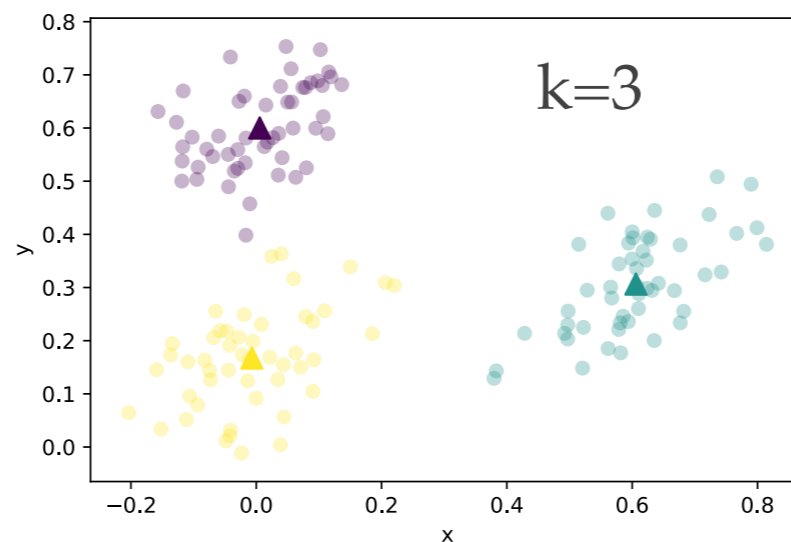
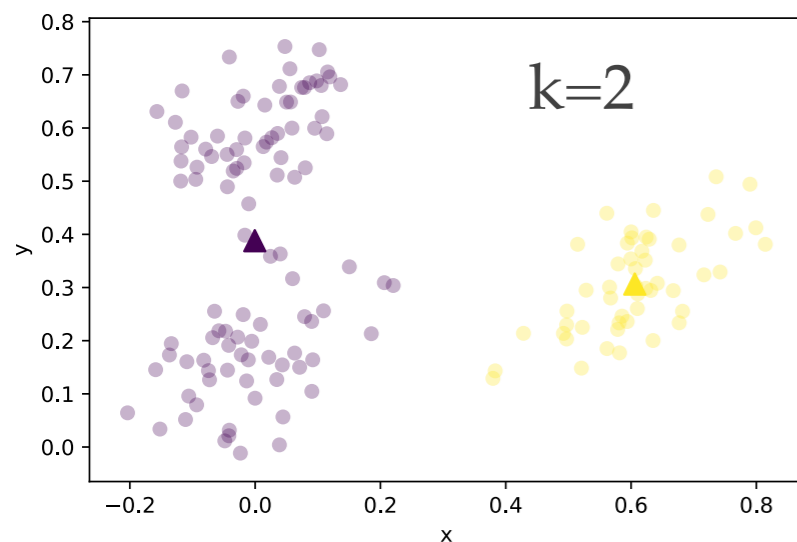
k=3, and two different random placements of centroids



The anatomy of K-means

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

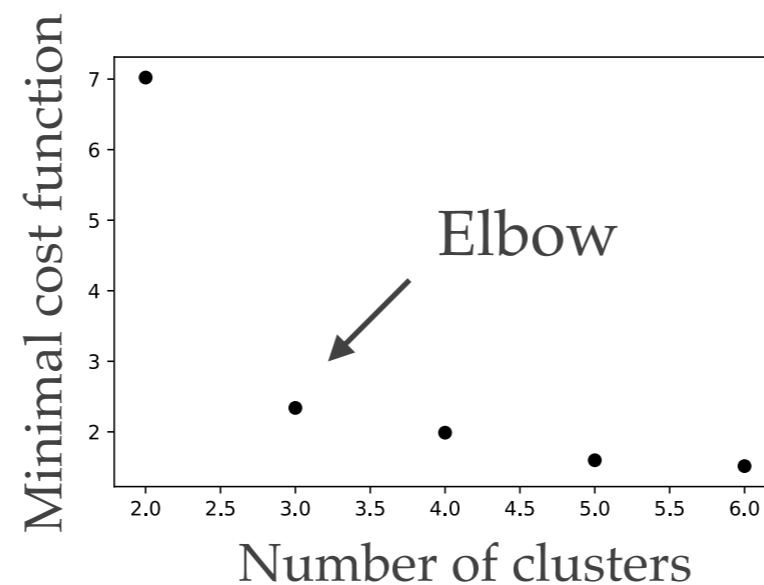
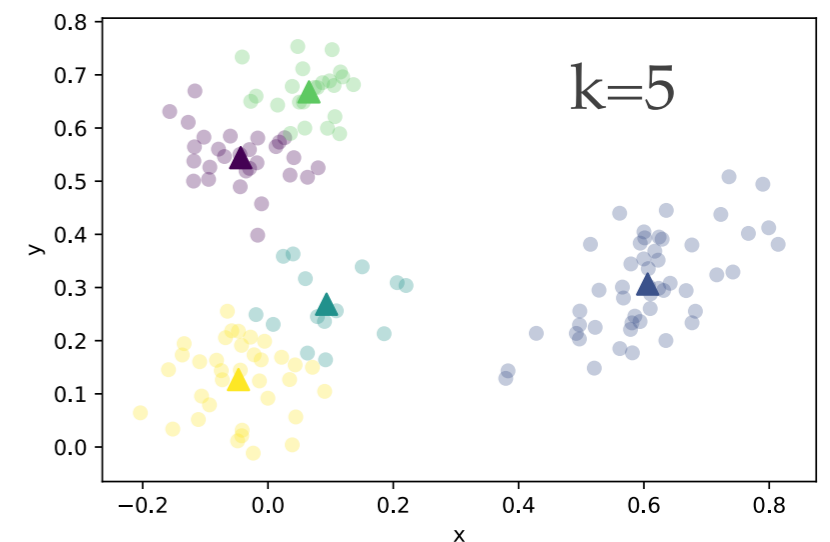
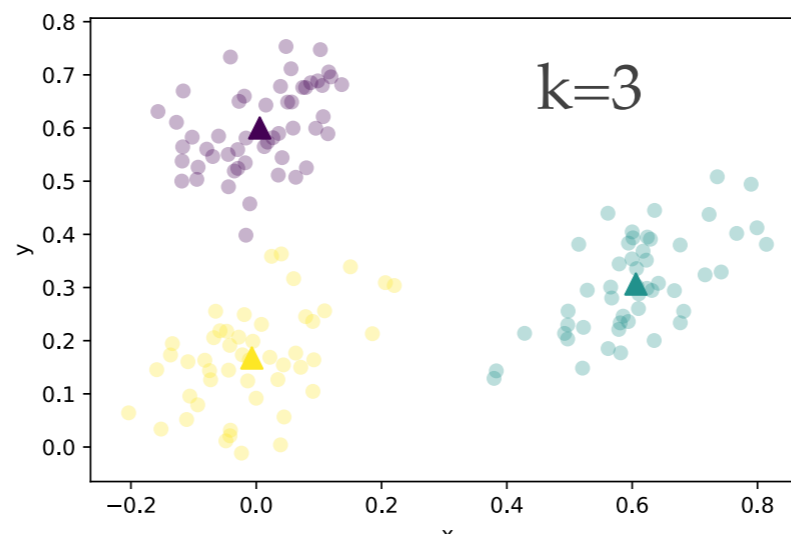
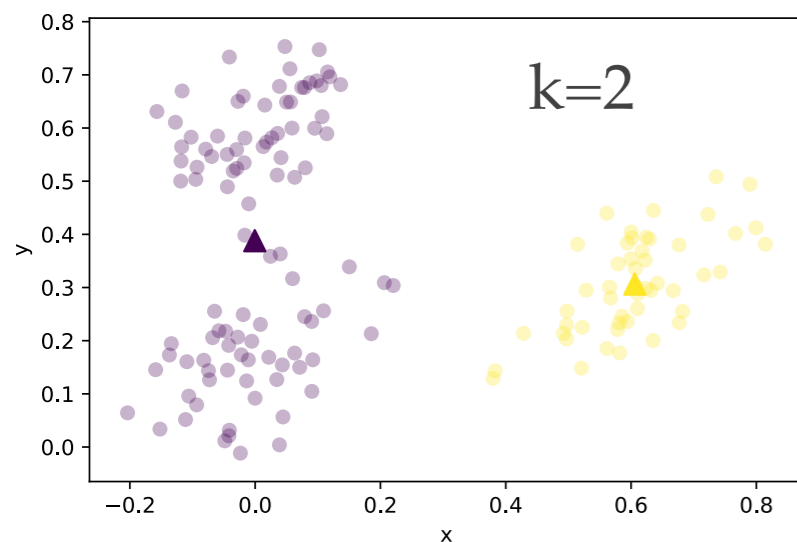
Hyper-parameters: the number of clusters, k .
Can we find the optimal k using the cost function?



The anatomy of K-means

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

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

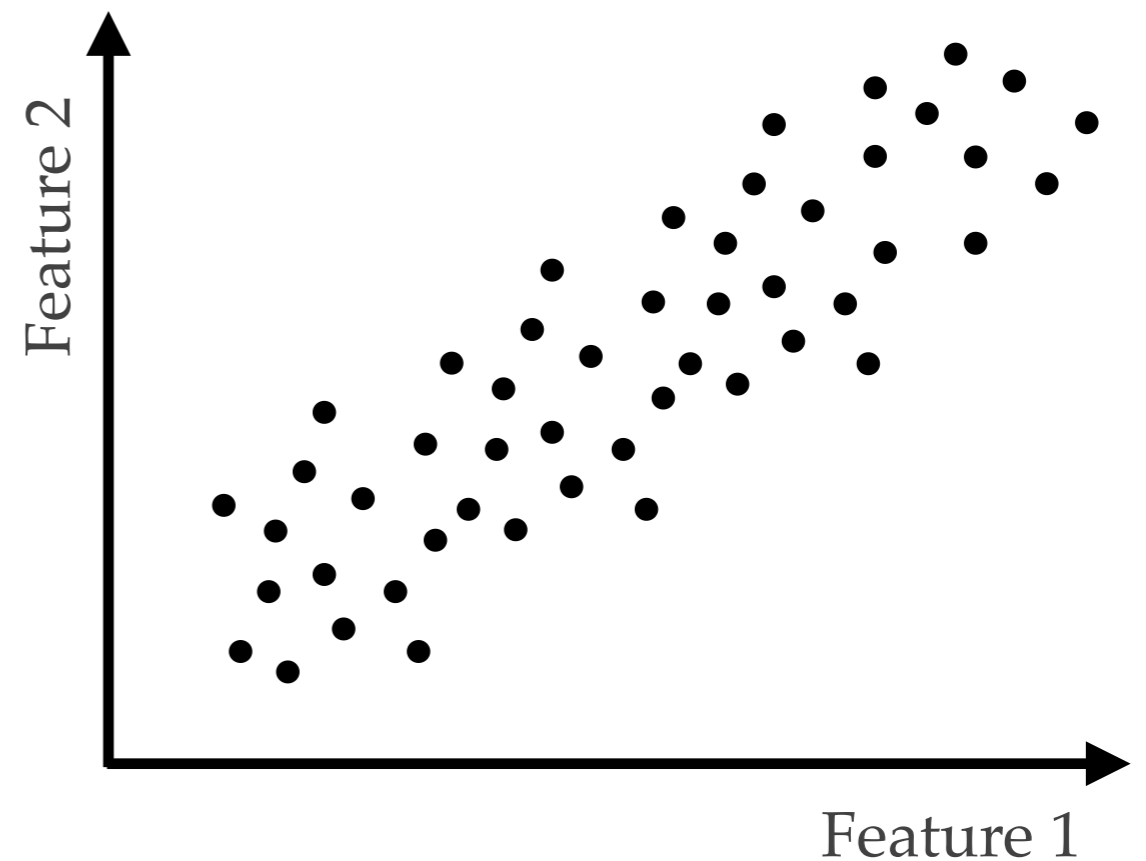
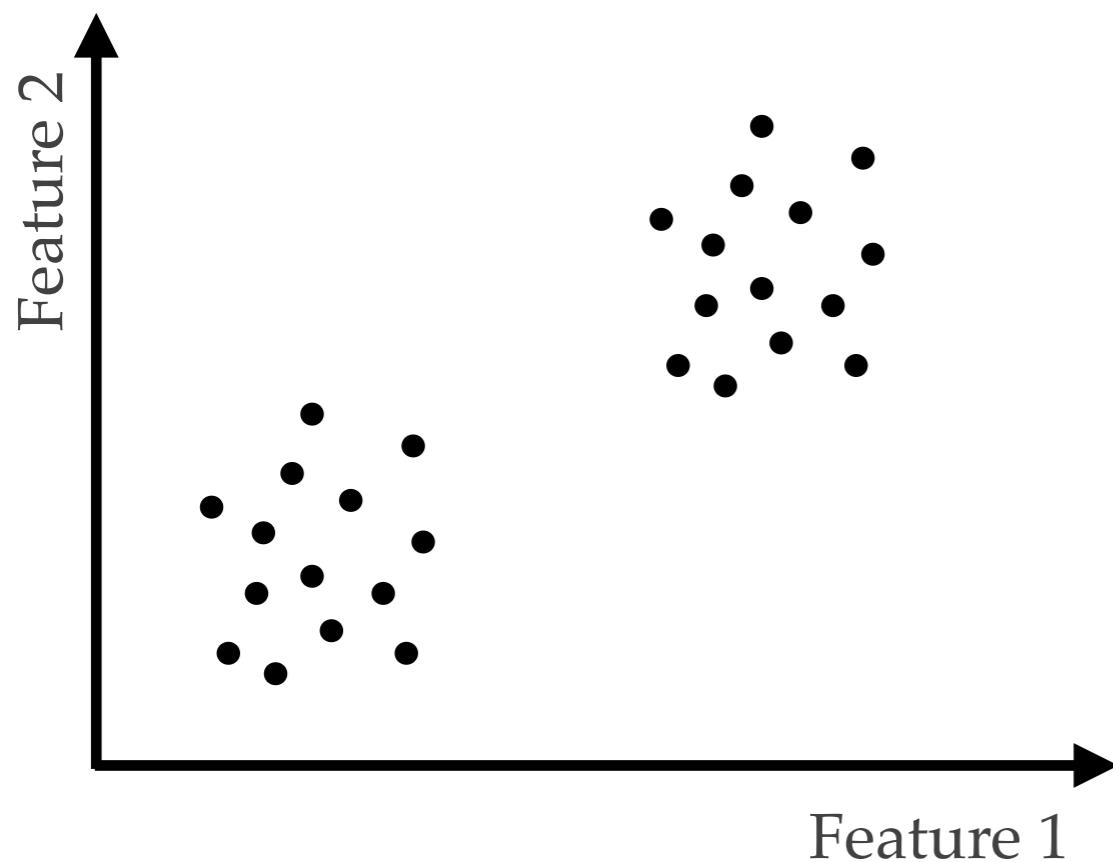


The anatomy of K-means

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

Input dataset: a list of objects with measured features.

For which datasets should we use K-means?

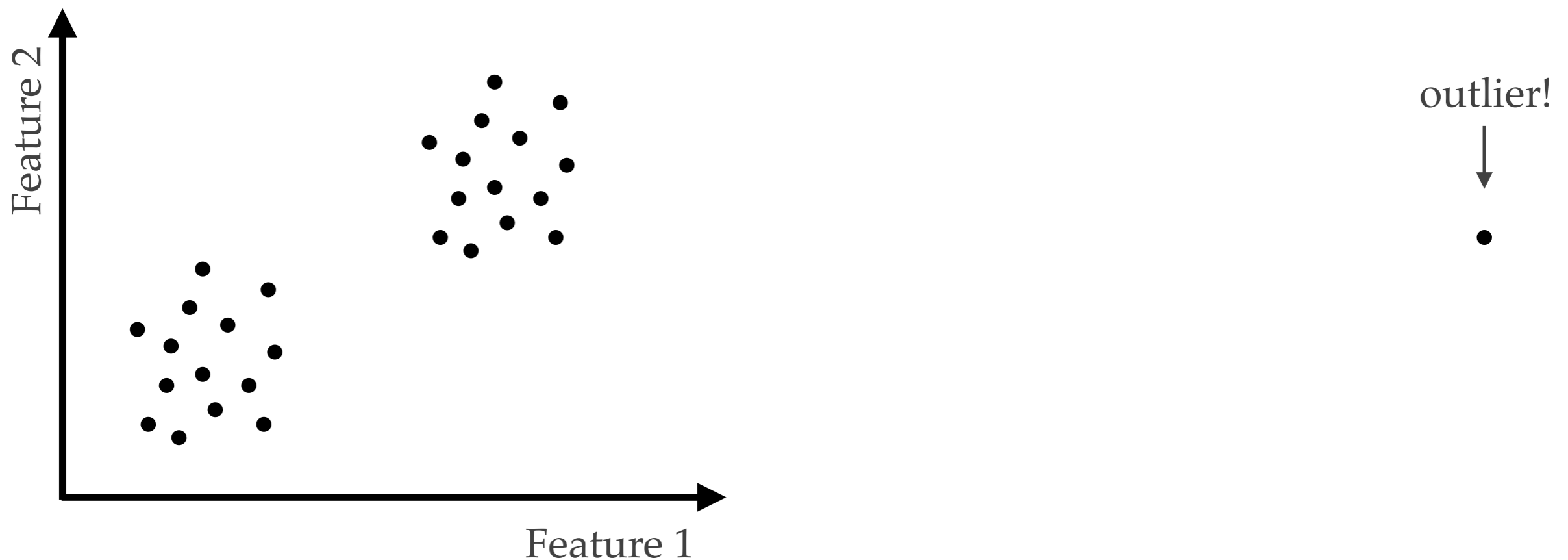


The anatomy of K-means

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

Input dataset: a list of objects with measured features.

What happens when we have an outlier in the dataset?

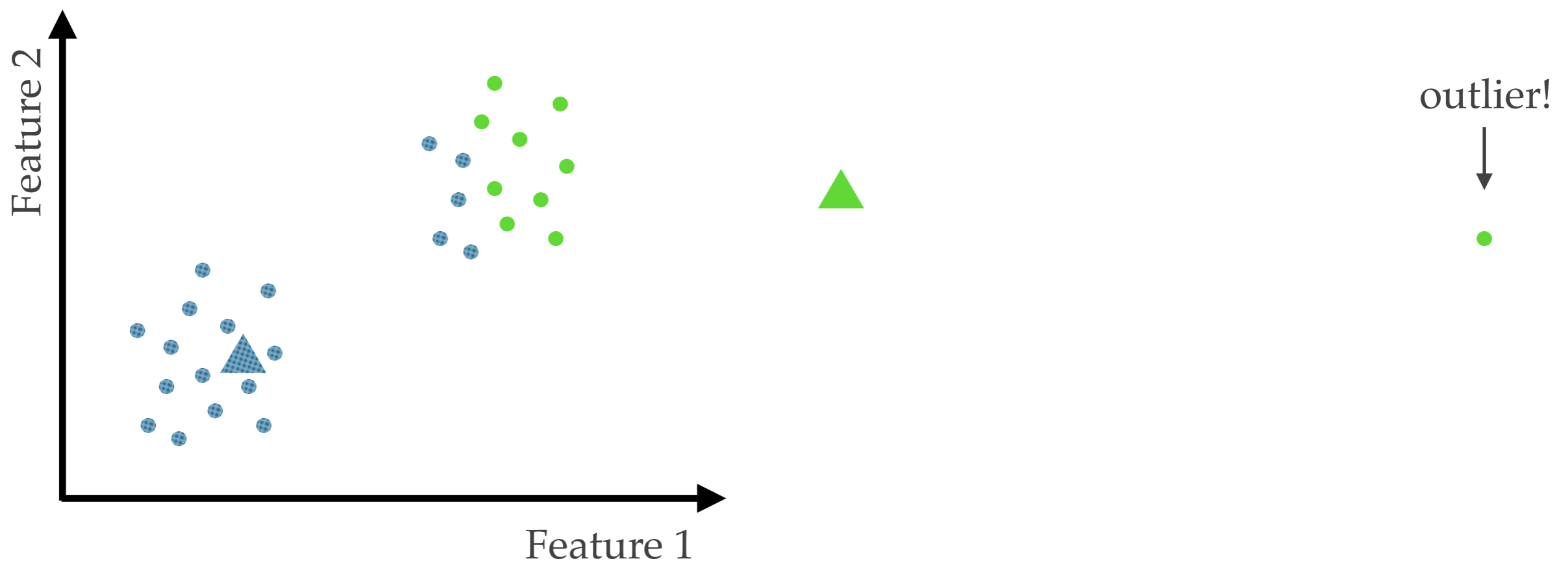


The anatomy of K-means

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

Input dataset: a list of objects with measured features.

What happens when we have an outlier in the dataset?

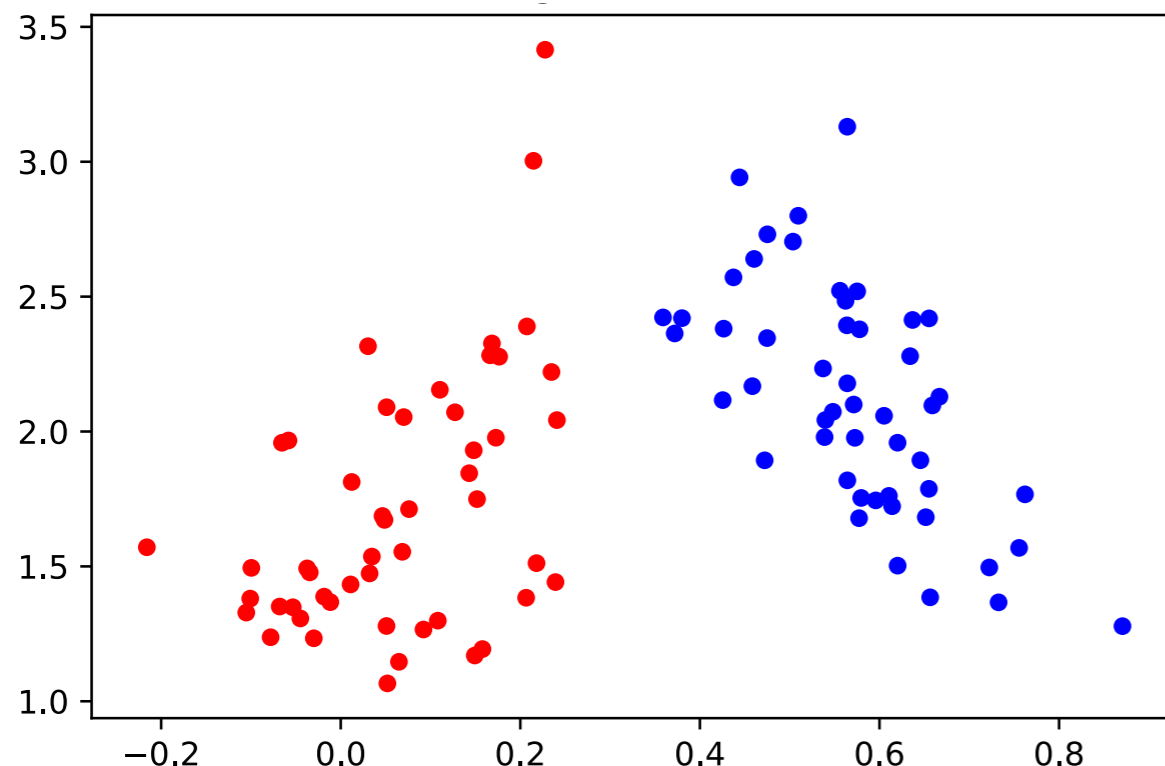


The anatomy of K-means

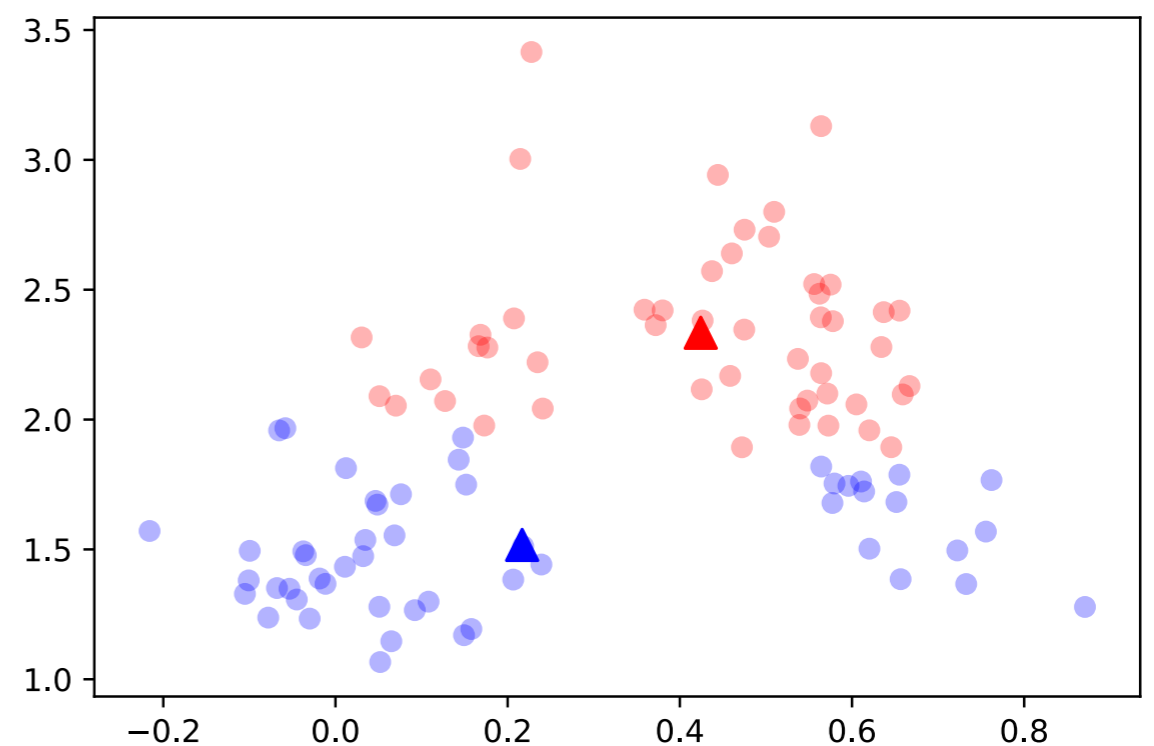
$$f(\overrightarrow{X}, \{a_1, a_2, \dots\}) = \overrightarrow{y}$$

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

input dataset



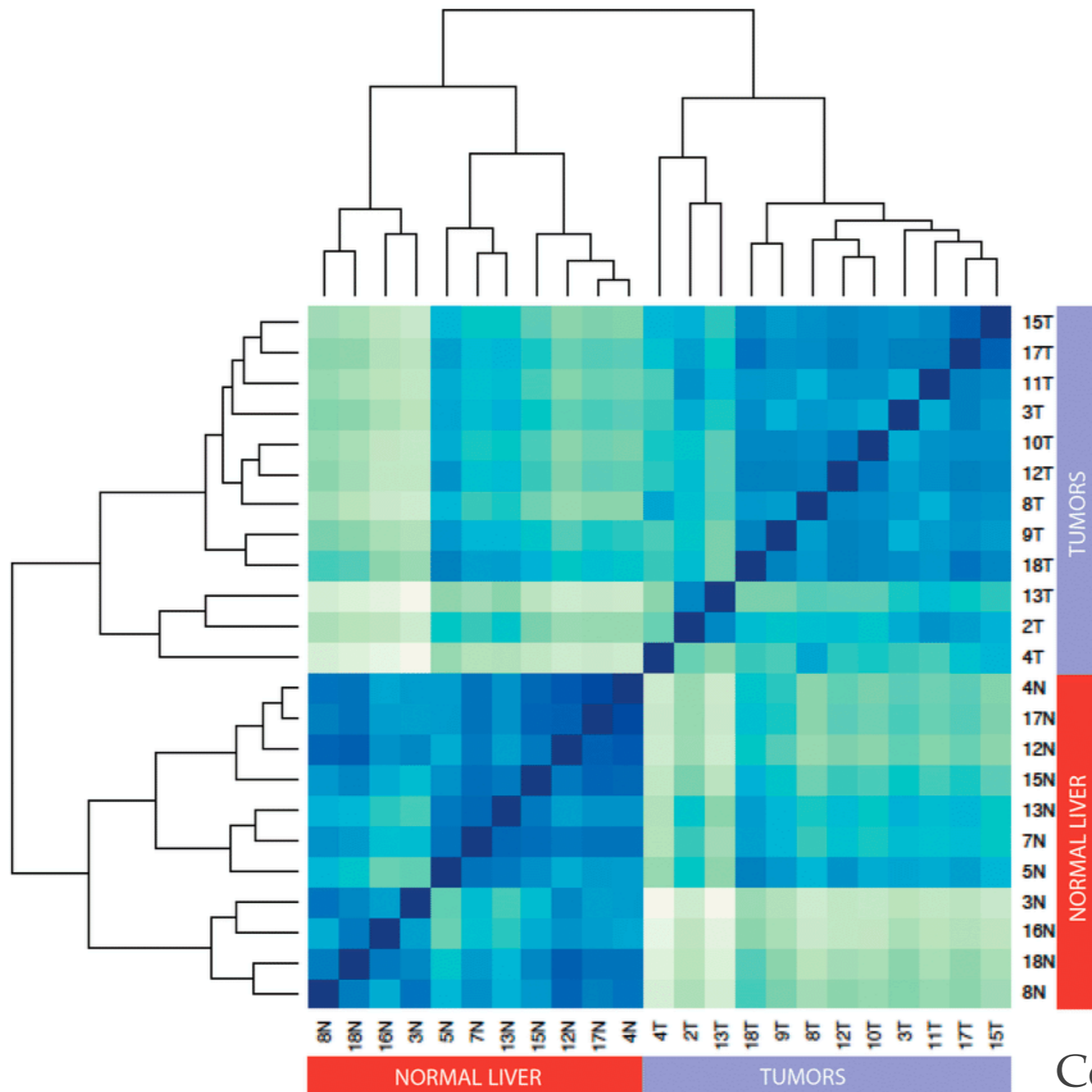
K-means output



Questions?

Hierarchical Clustering

or, how to visualize complicated similarity measures

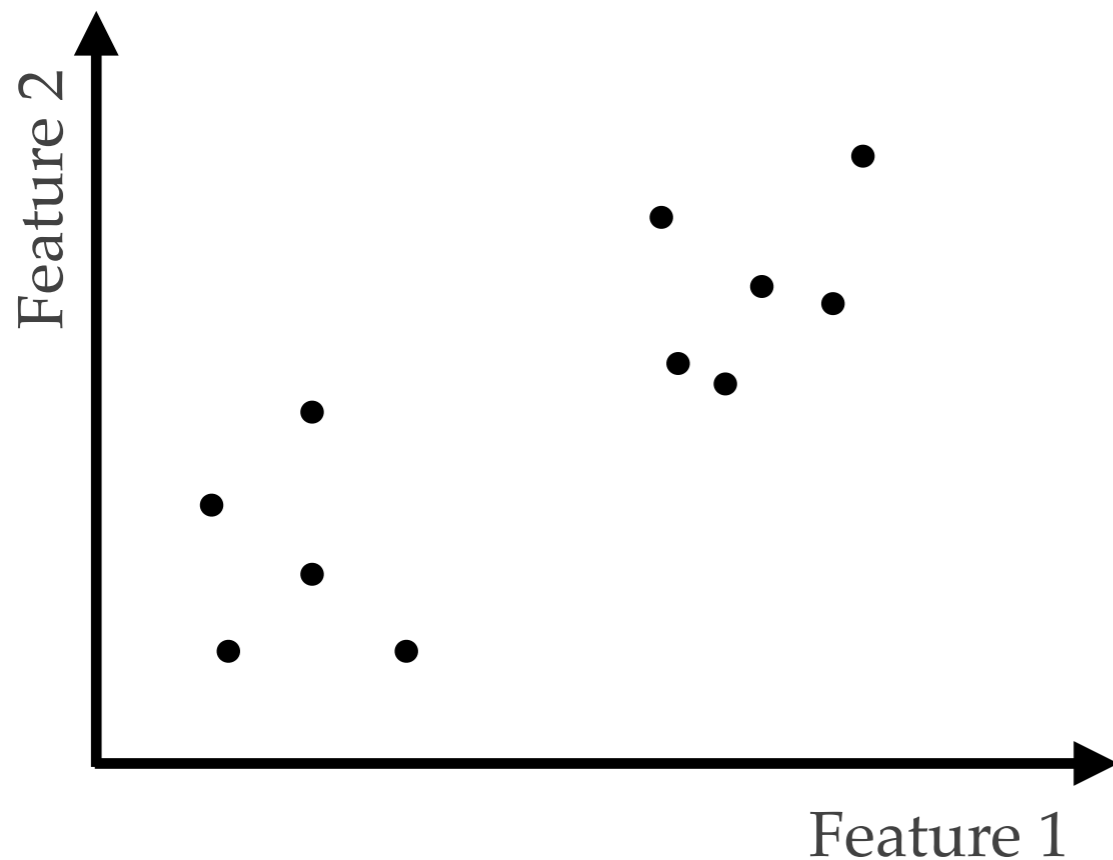


Correa-Gallego+ 2016

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.

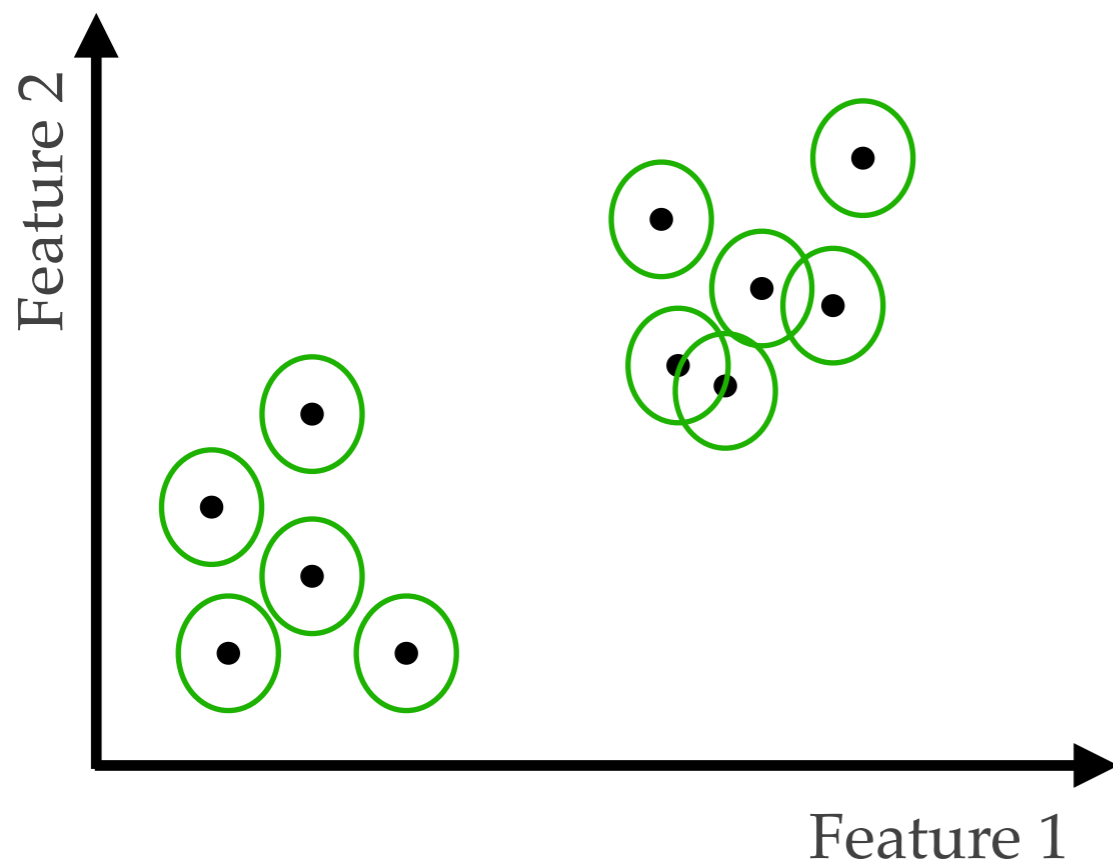


Hierarchal Clustering

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.

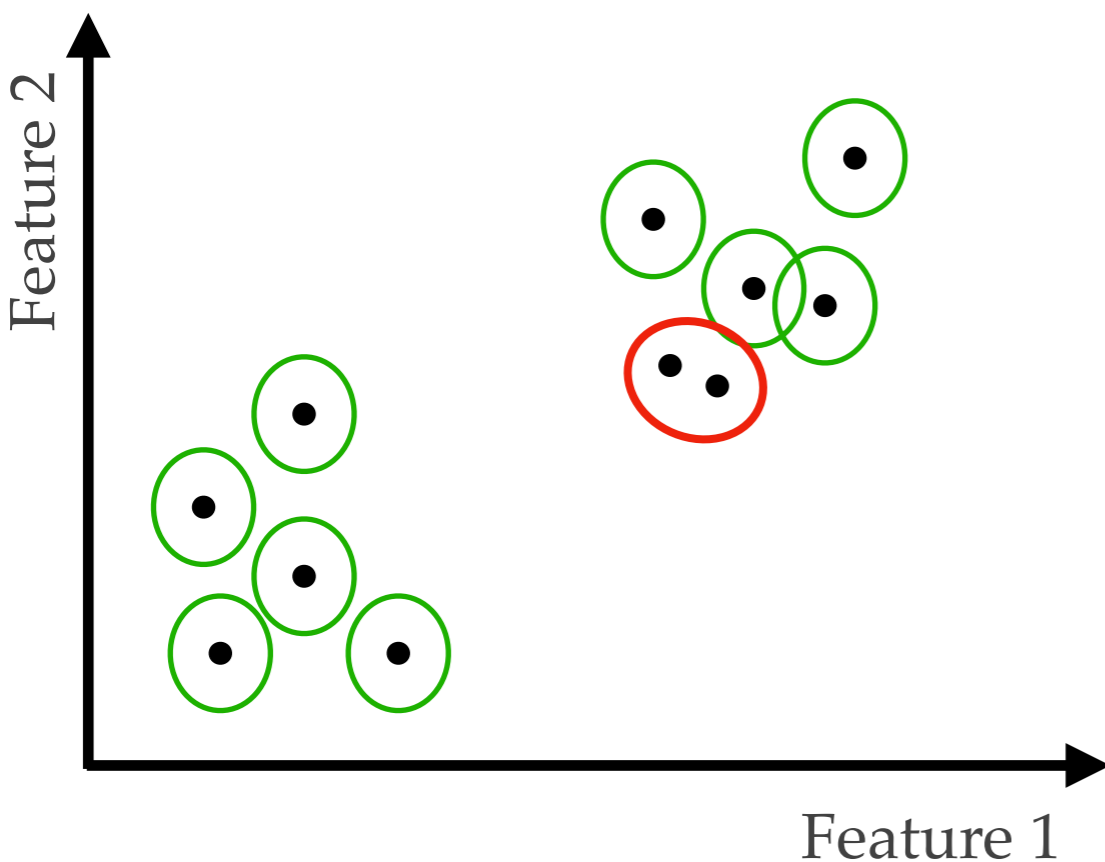
Next: the algorithm merges the two closest clusters into a single cluster. Then, the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



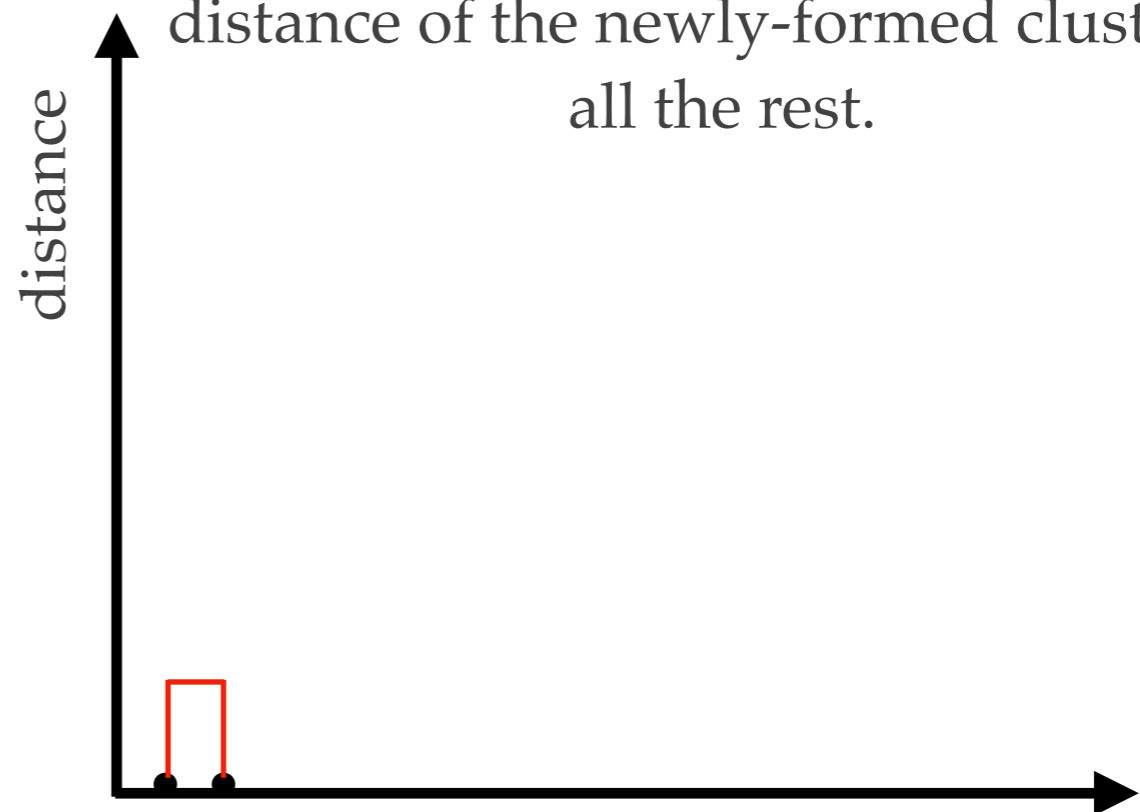
Hierarchical Clustering

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.



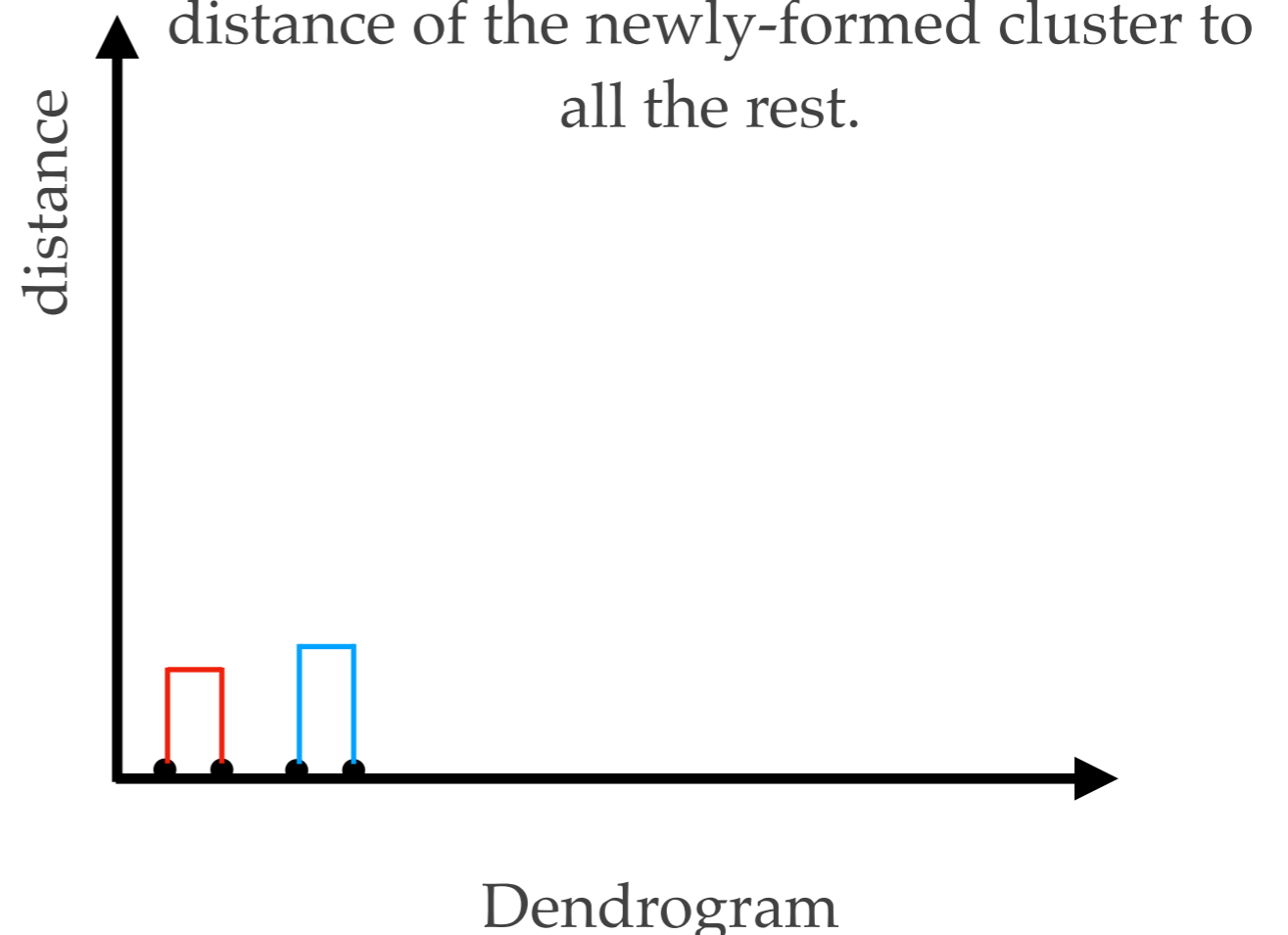
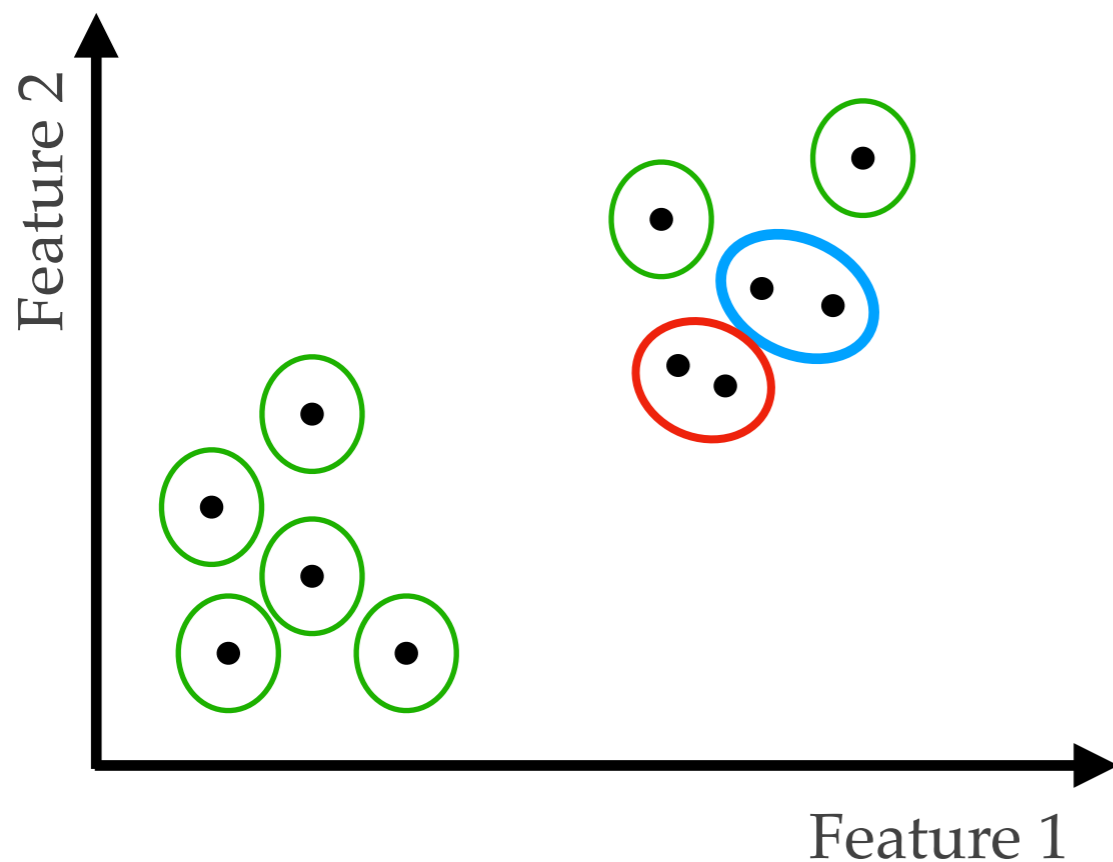
Dendrogram

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.

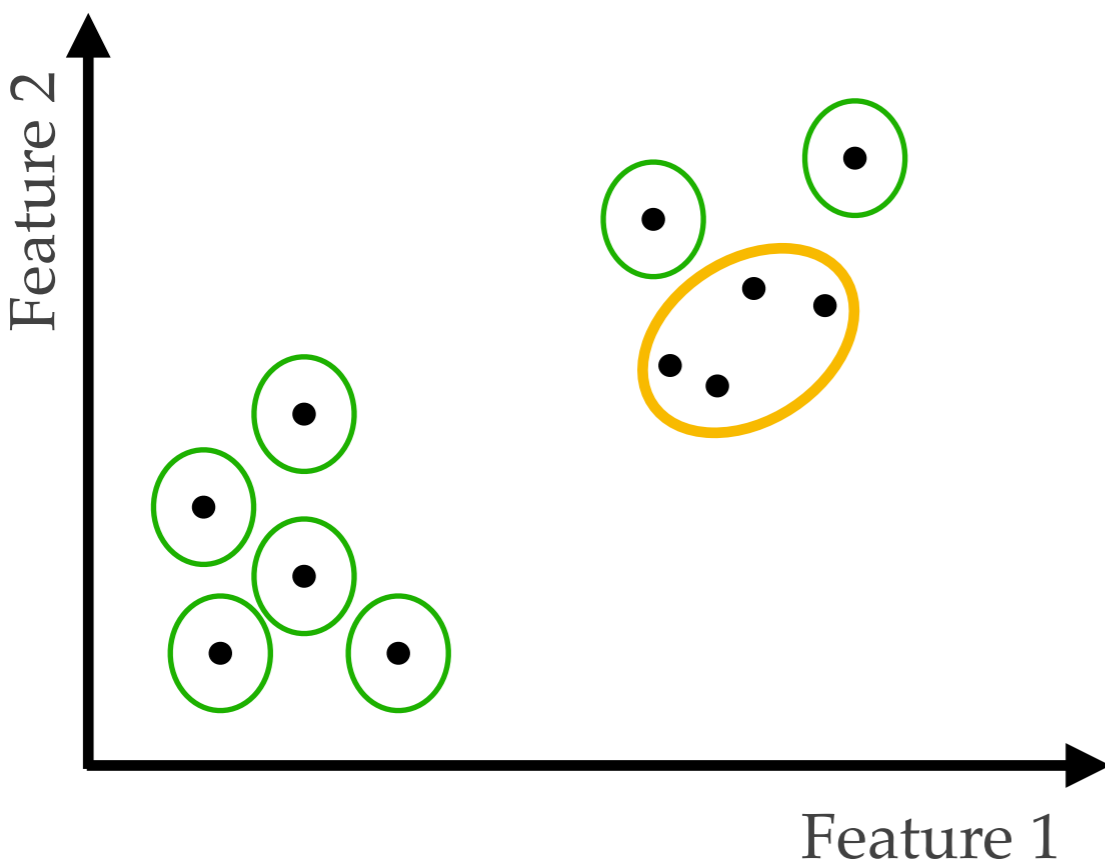
Next: the algorithm merges the two closest clusters into a single cluster. Then, the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



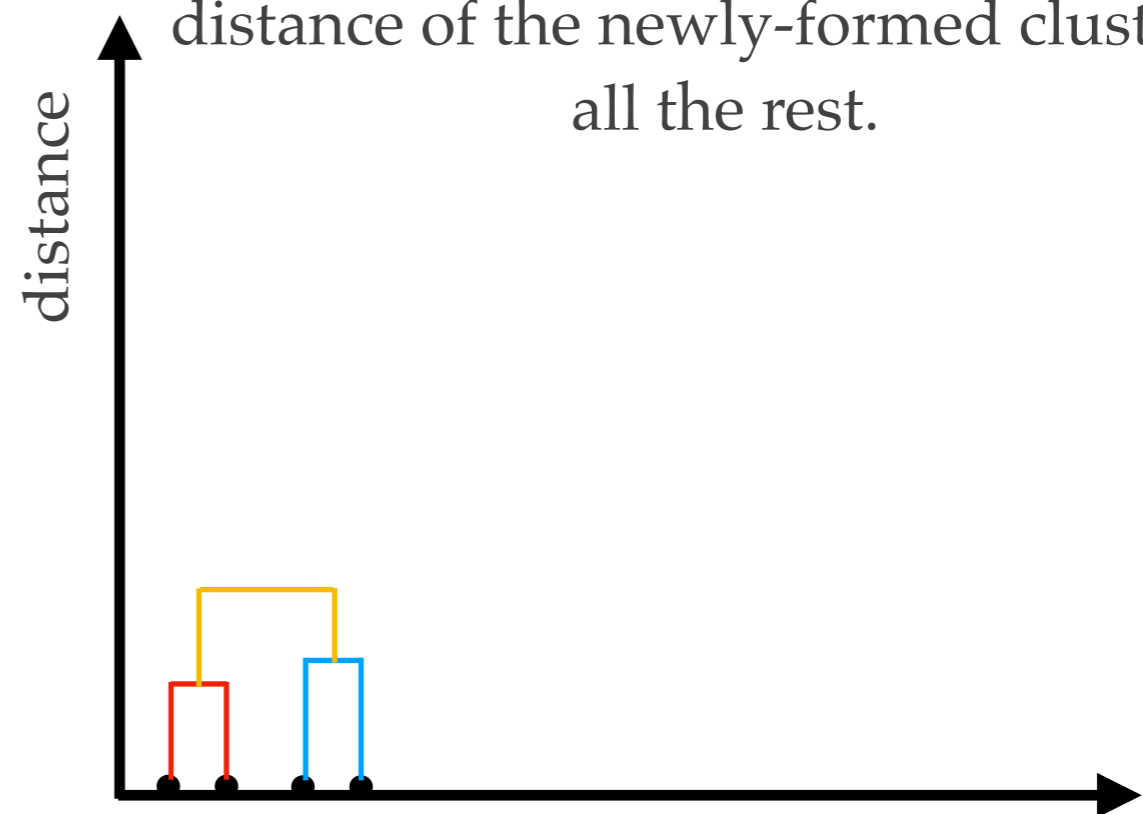
Hierarchical Clustering

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.

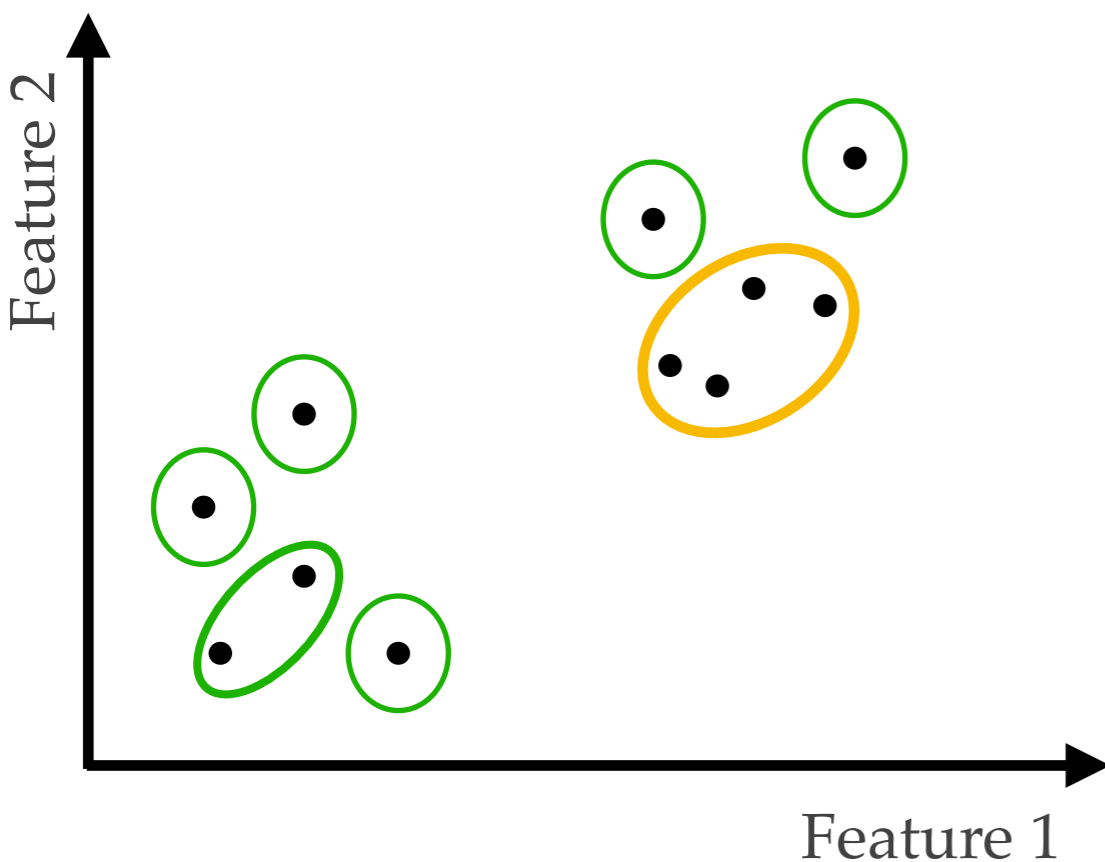


Dendrogram

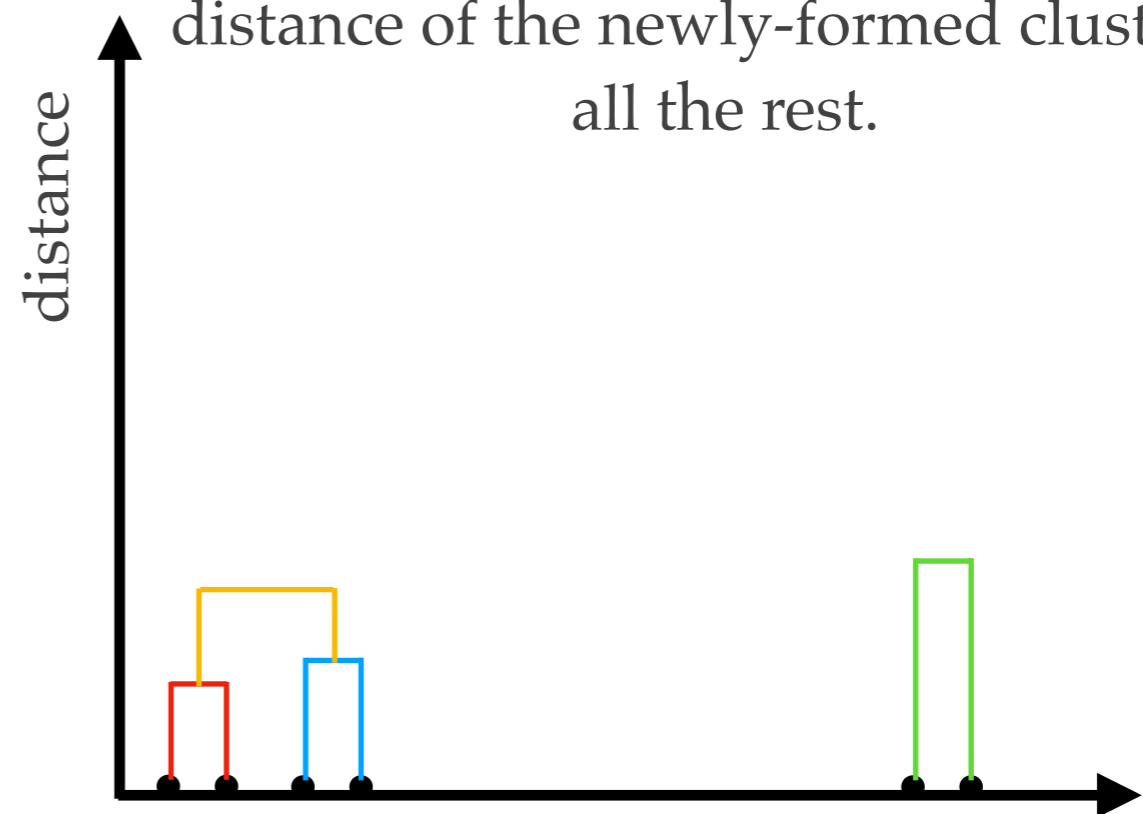
Hierarchical Clustering

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.

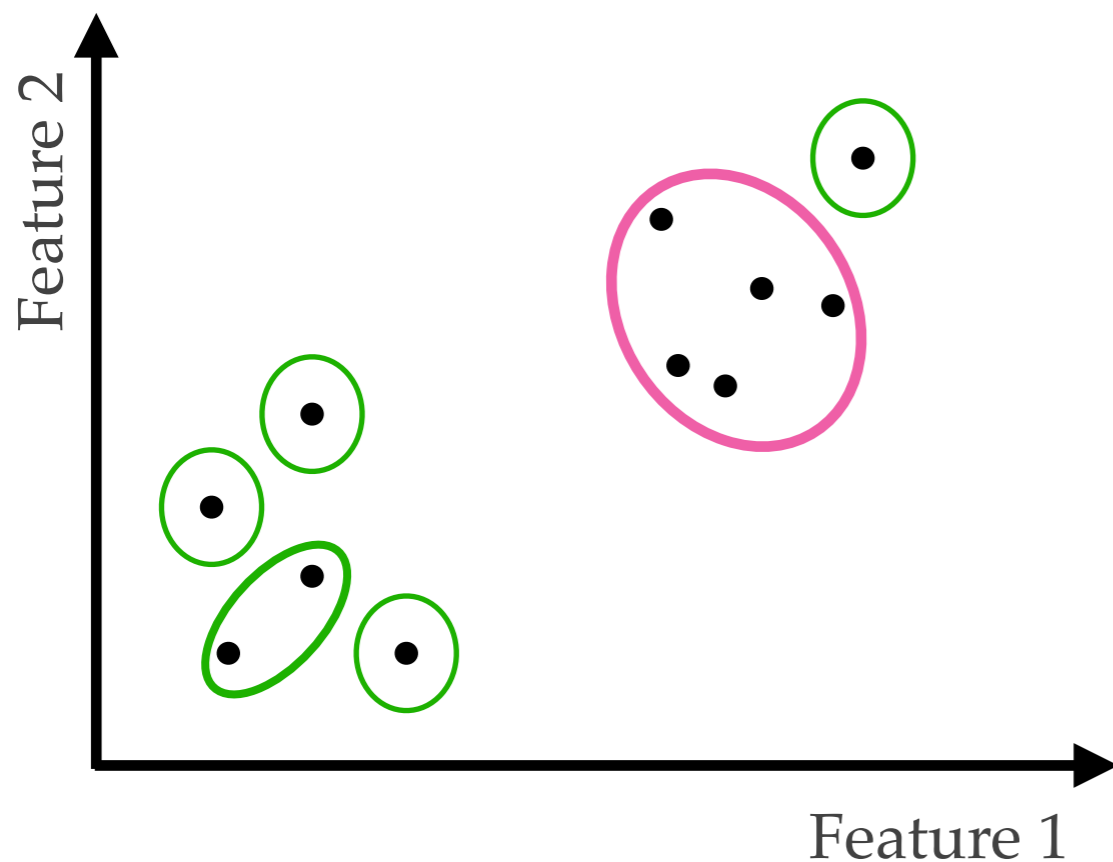


Dendrogram

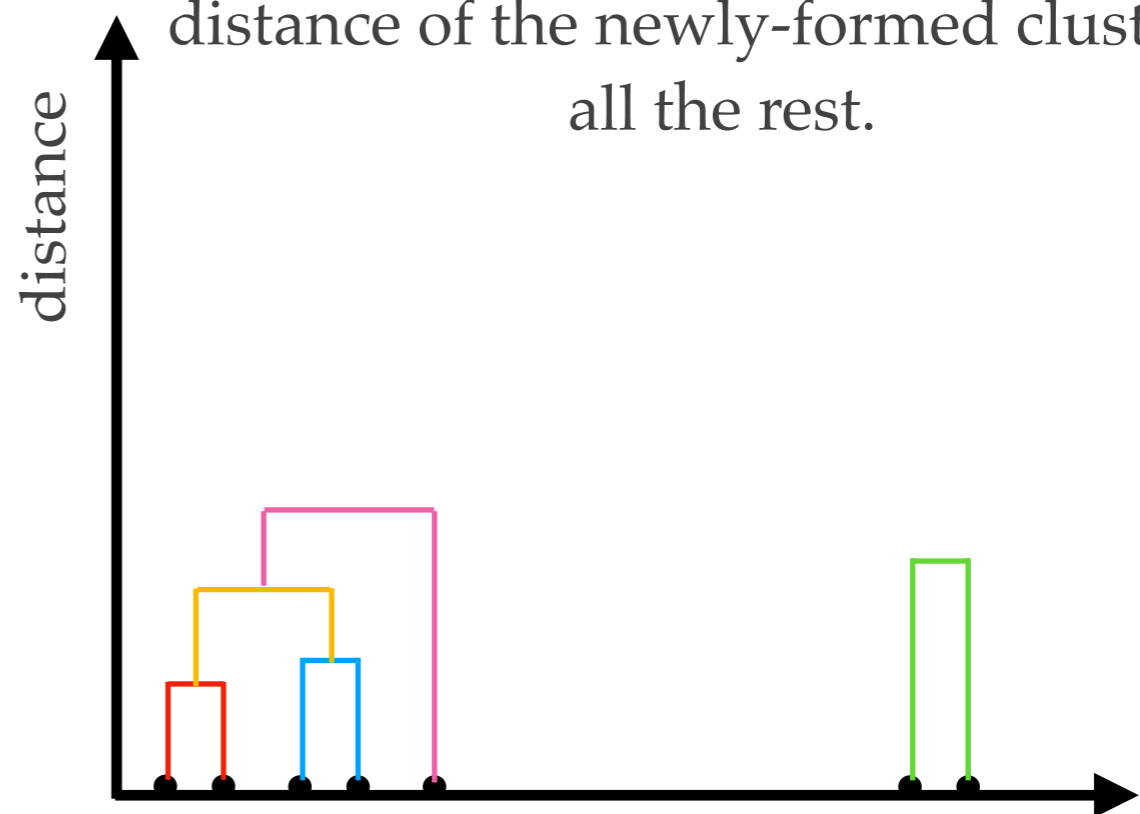
Hierarchical Clustering

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.

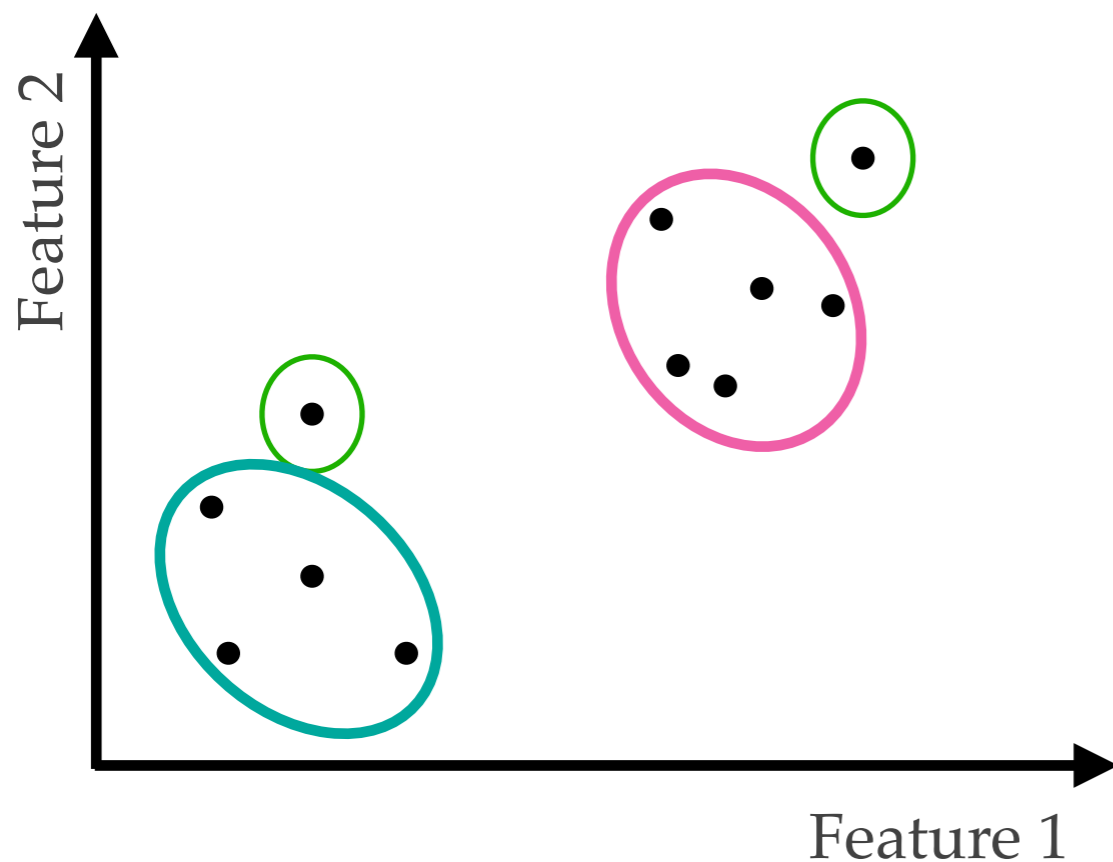


Dendrogram

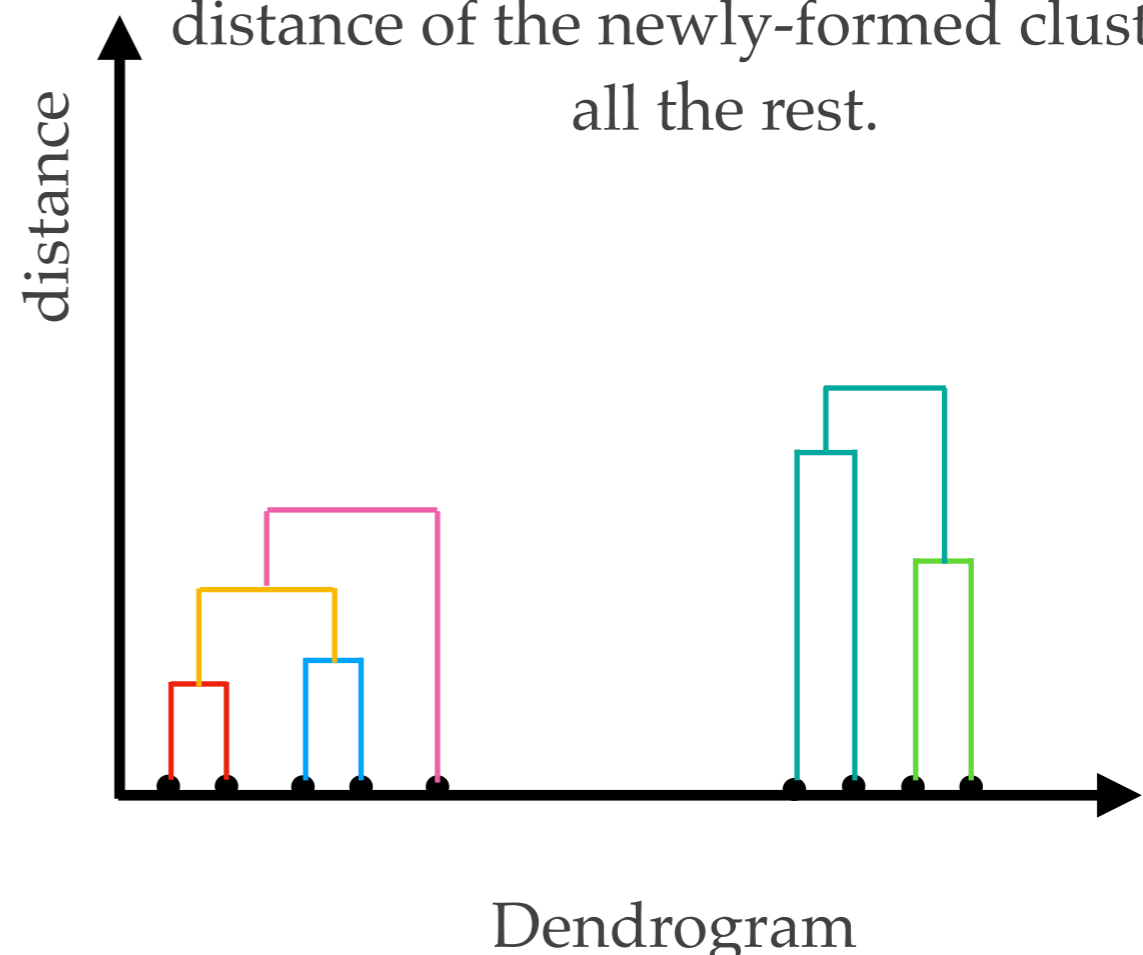
Hierarchical Clustering

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.



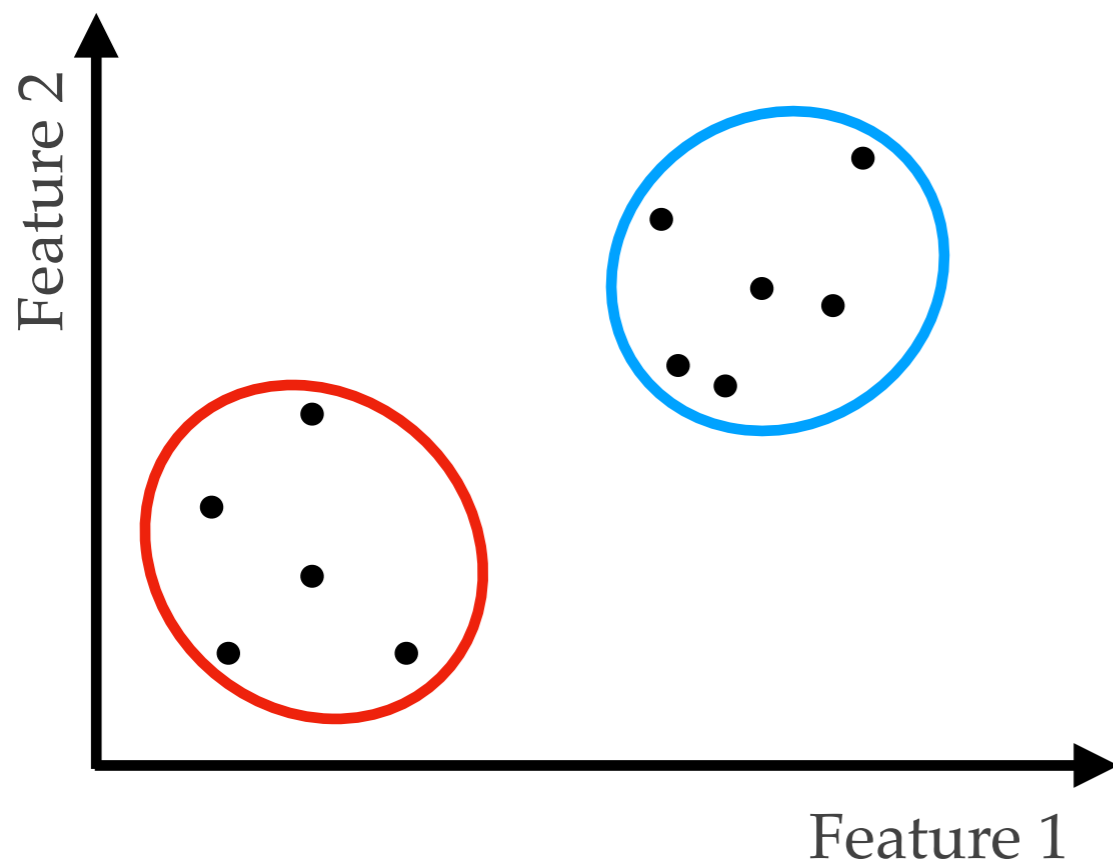
Next: the algorithm merges the two closest clusters into a single cluster. Then, the algorithm re-calculates the distance of the newly-formed cluster to all the rest.



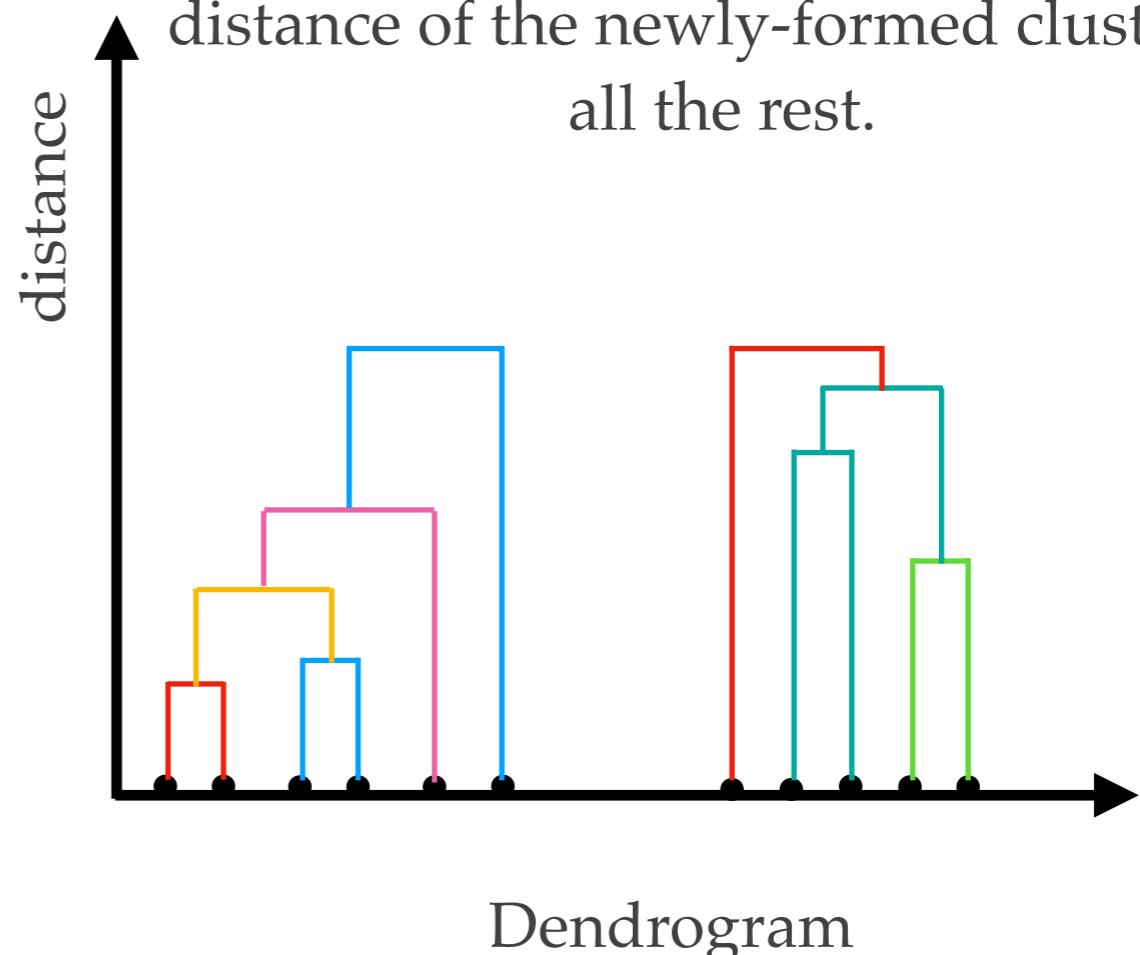
Hierarchical Clustering

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.

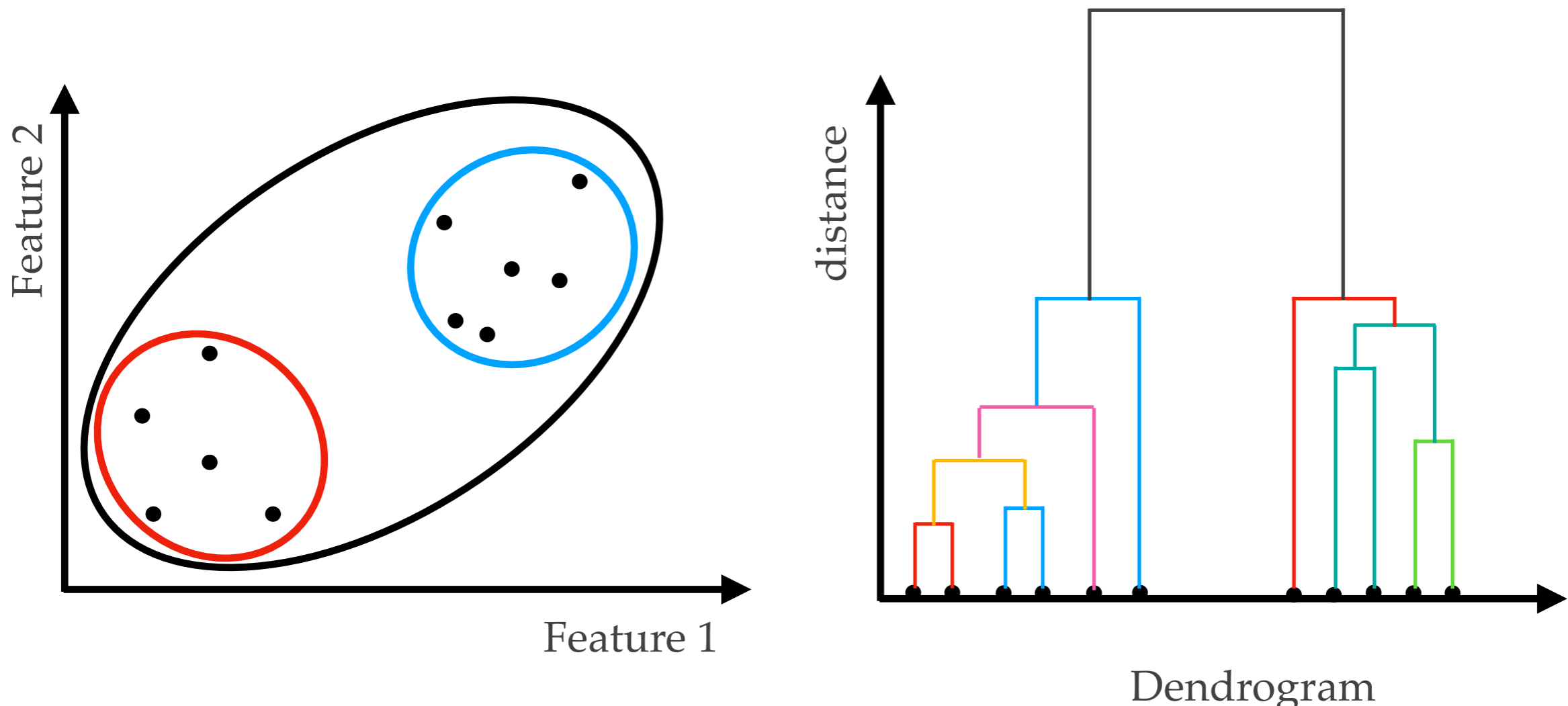


Hierarchical Clustering

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



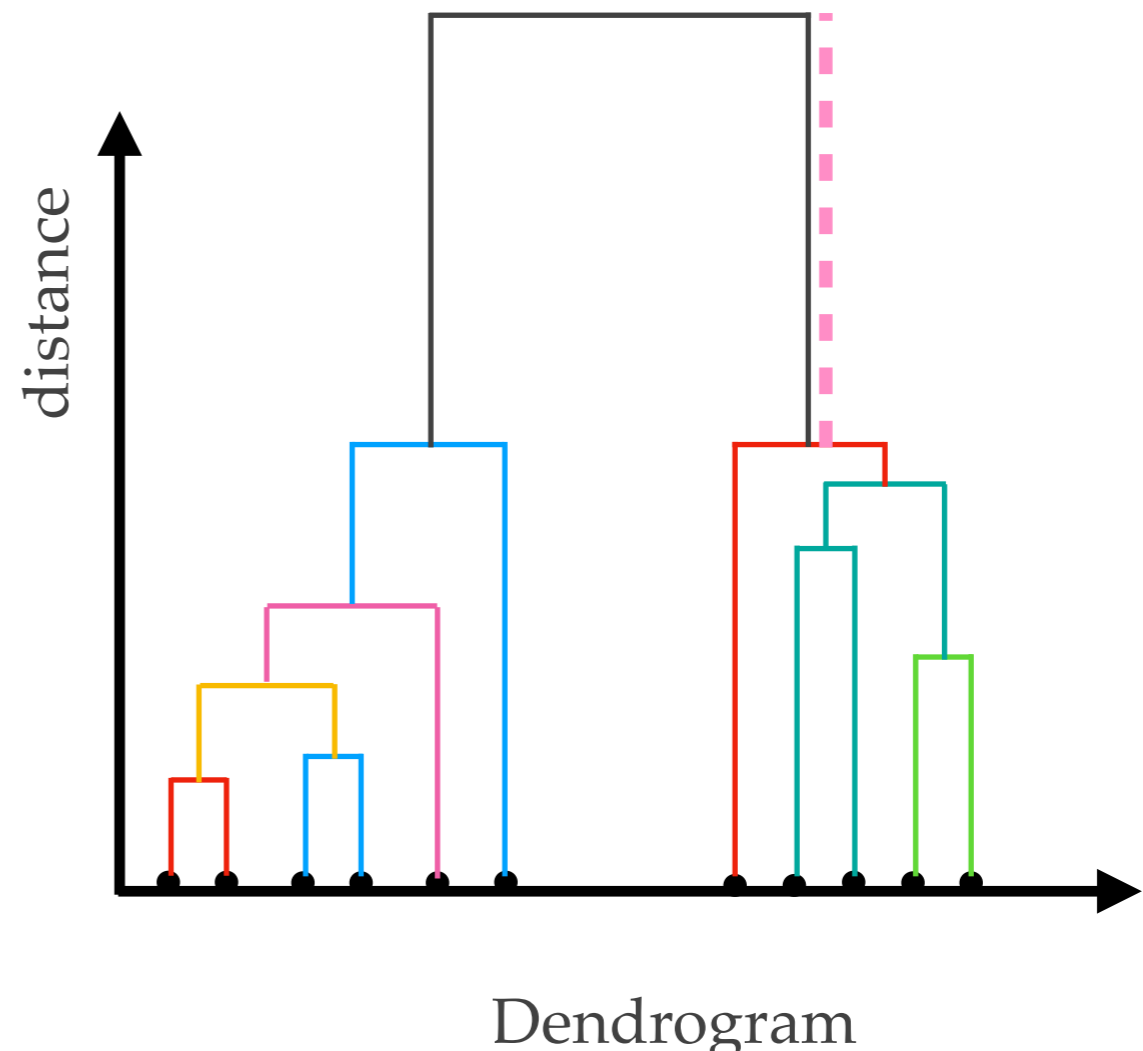
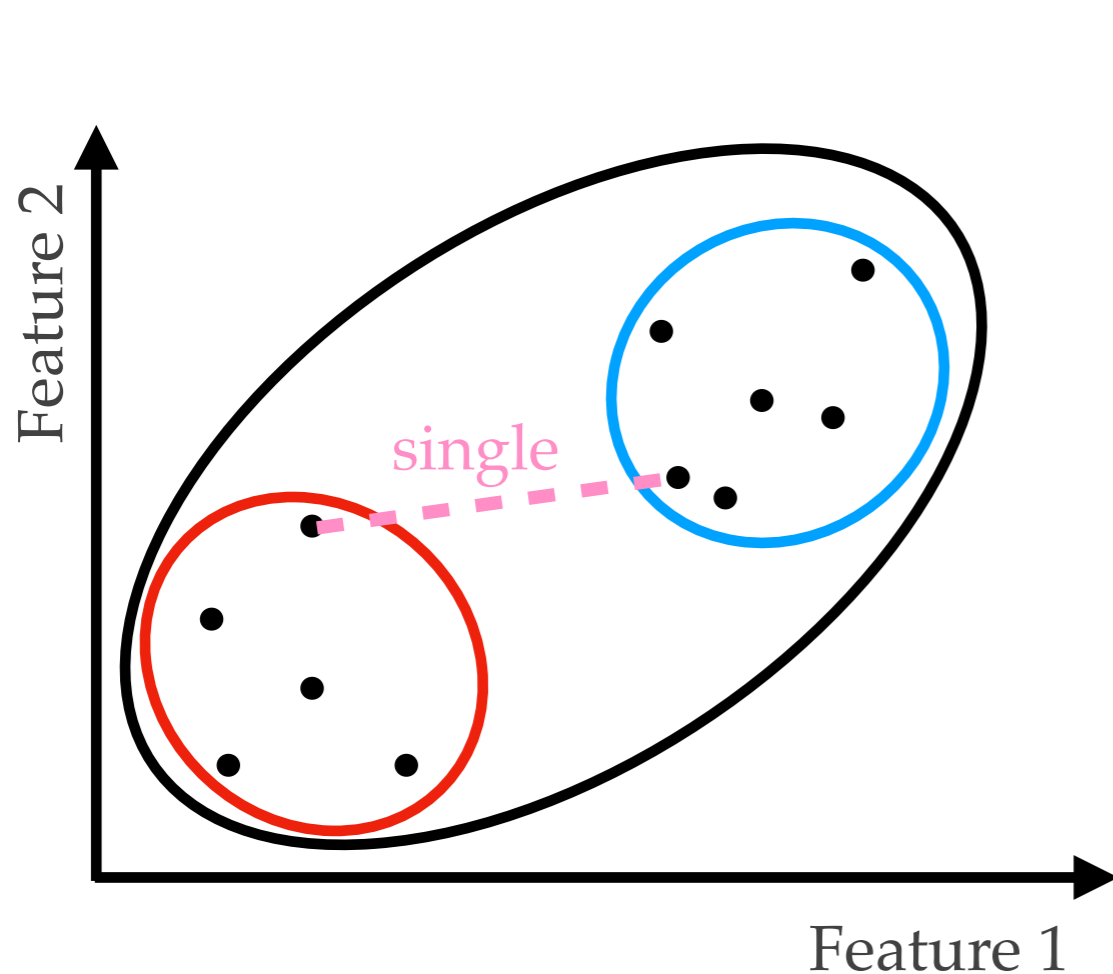
The anatomy of Hierarchical Clustering

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

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.



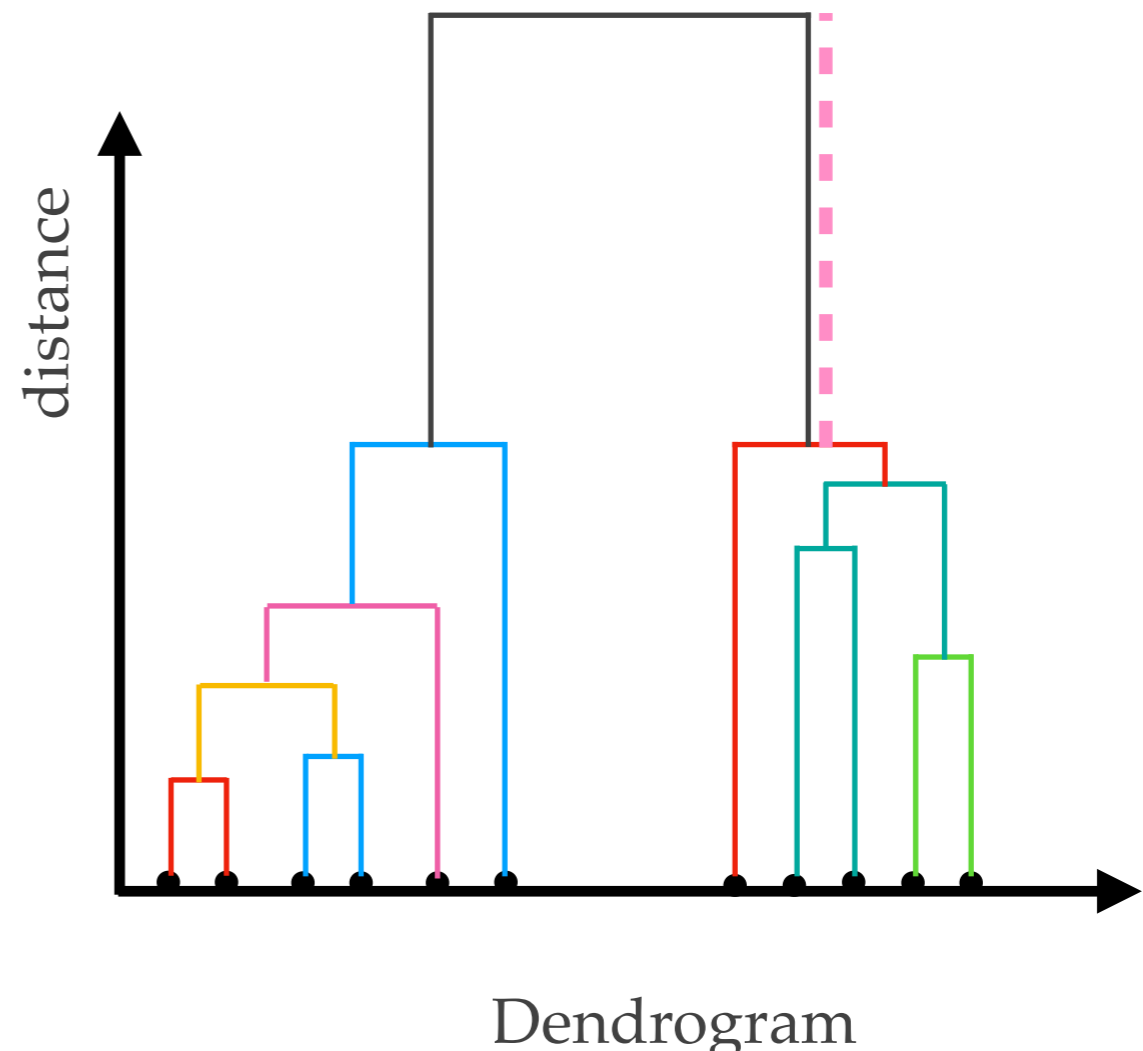
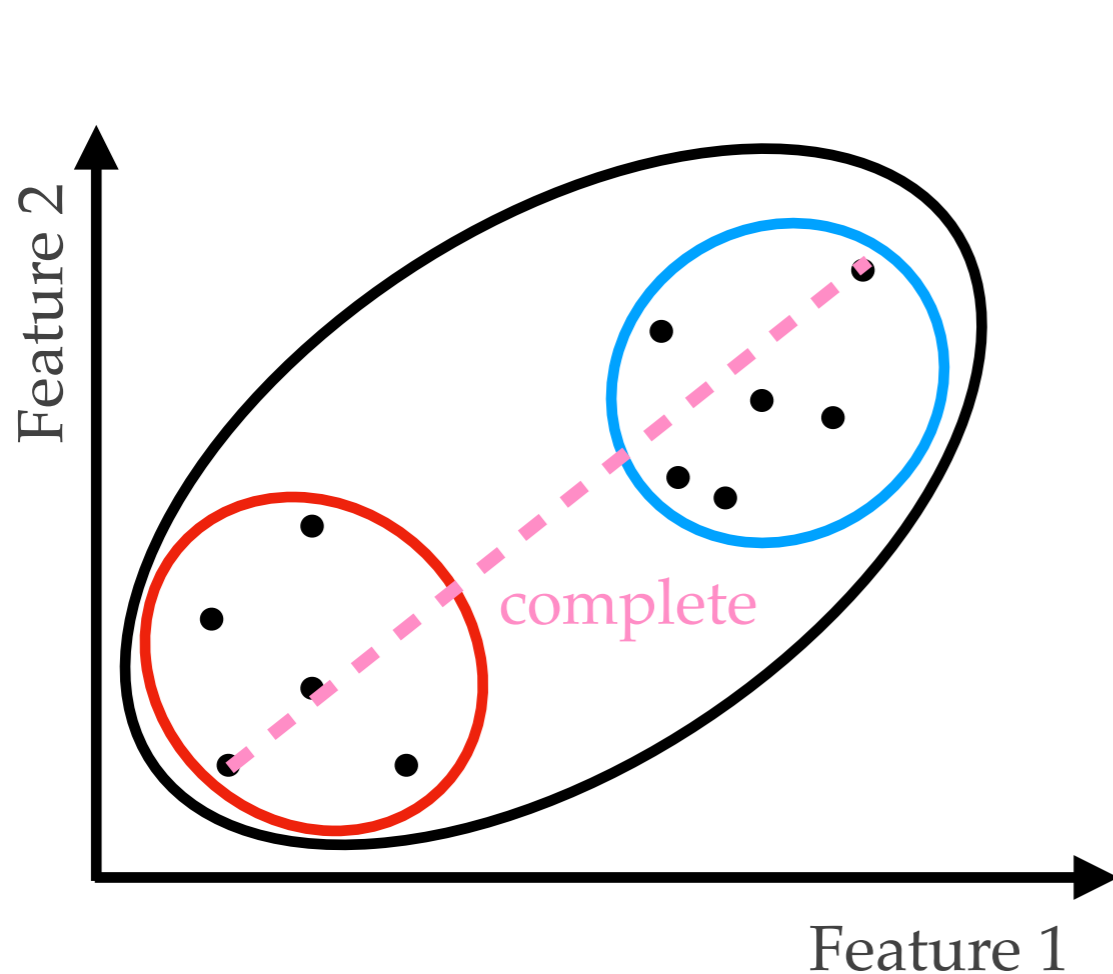
The anatomy of Hierarchical Clustering

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

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.



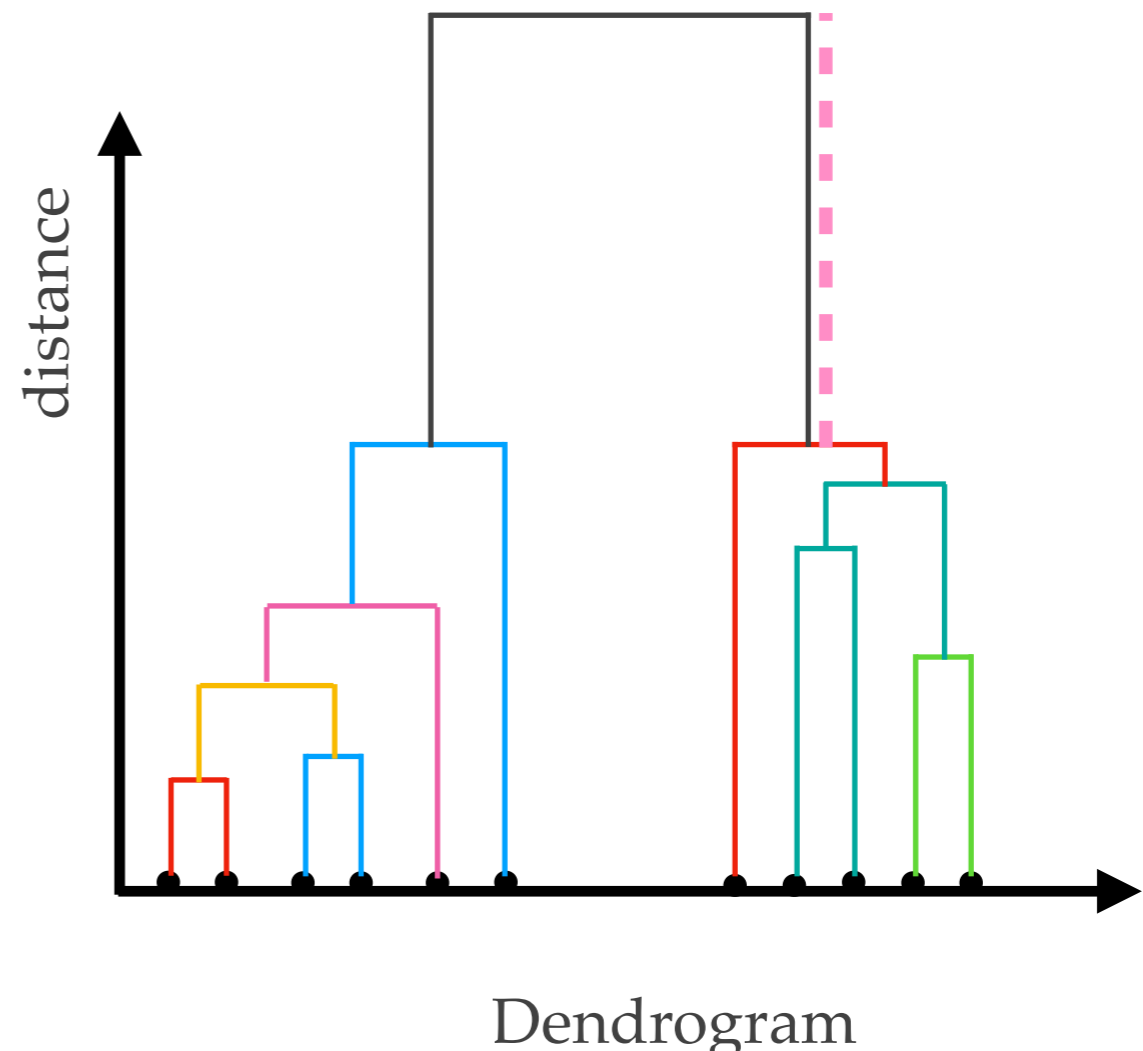
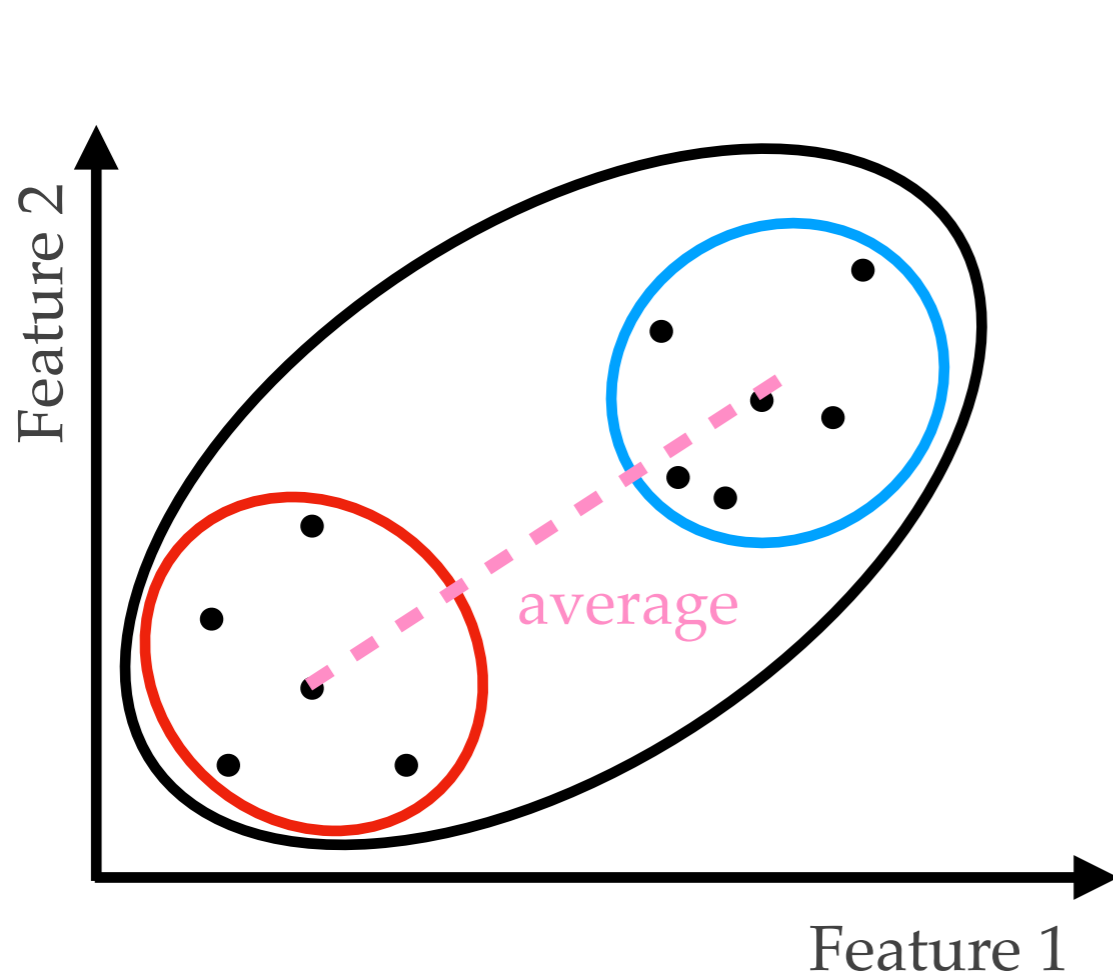
The anatomy of Hierarchical Clustering

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

Internal choices and/or internal cost function:

The linkage method is used to define a distance between two newly formed clusters.

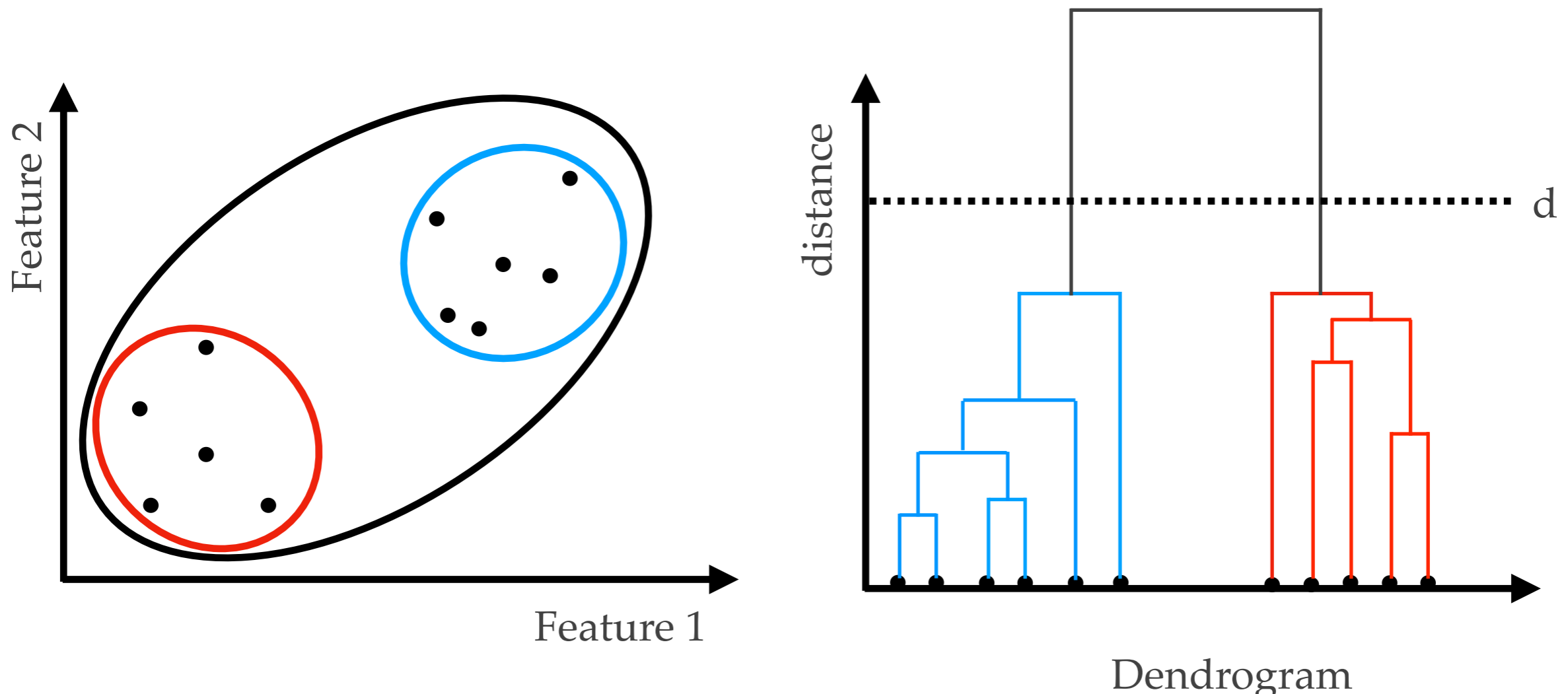
Methods include: single (minimal), complete (maximal), average, etc.



The anatomy of Hierarchical Clustering

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

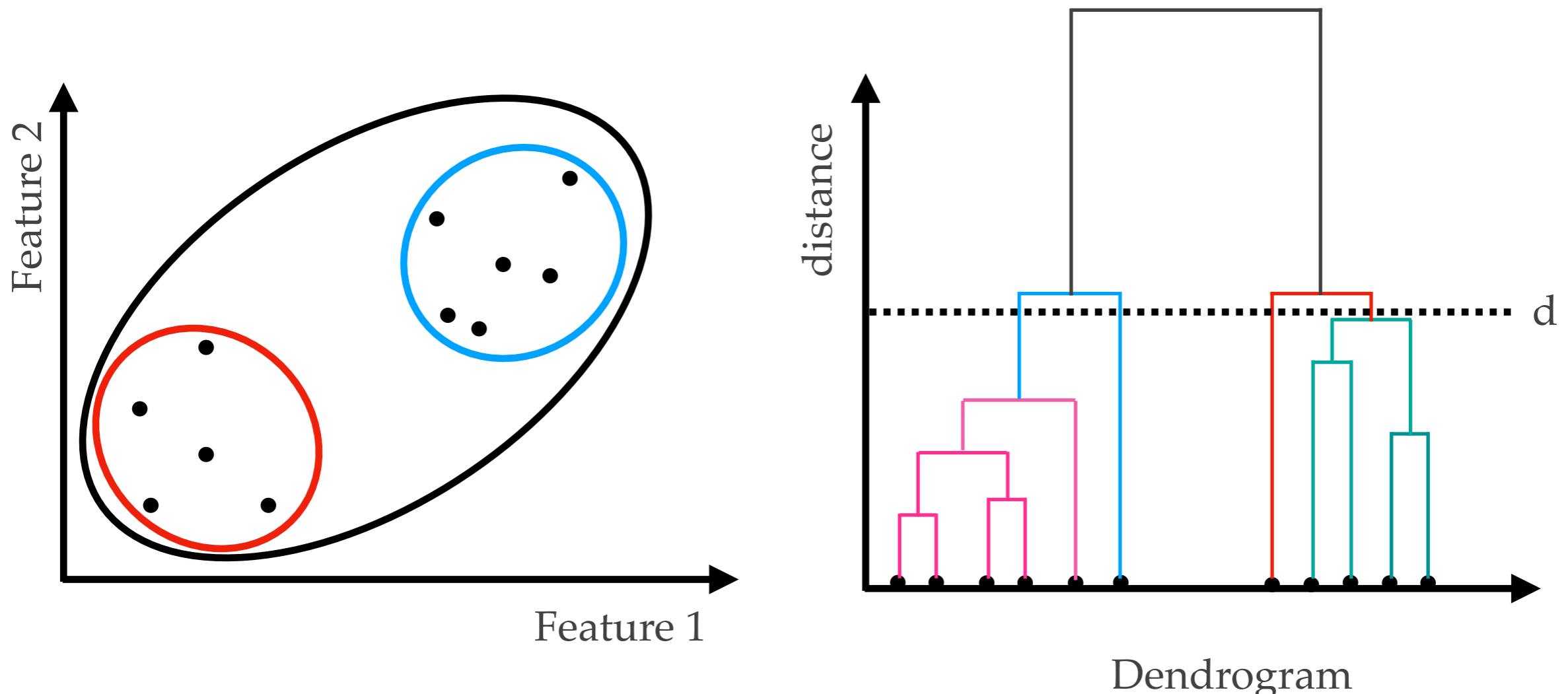
Hyper-parameters: clusters are defined beneath a threshold d . Alternatively, we can select a threshold d that corresponds to the desired number of clusters, k .



The anatomy of Hierarchical Clustering

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

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 .

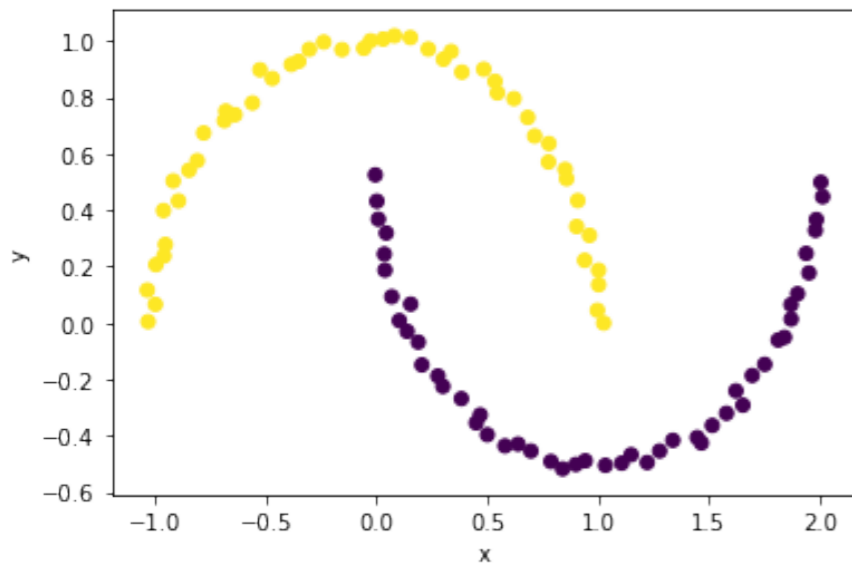


The anatomy of Hierarchical Clustering

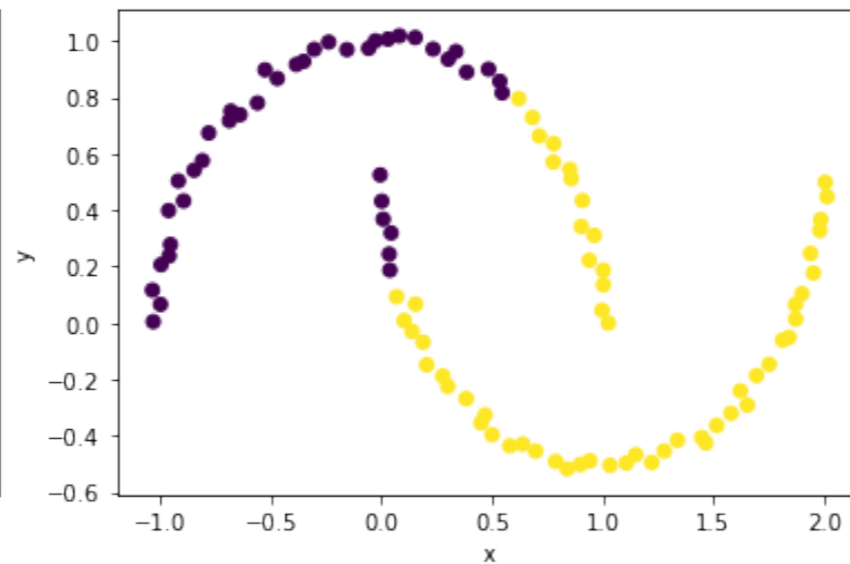
$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

Hyper-parameters: the linkage method is essentially a hyper-parameter of the algorithm. Different linkages will result in a different output.

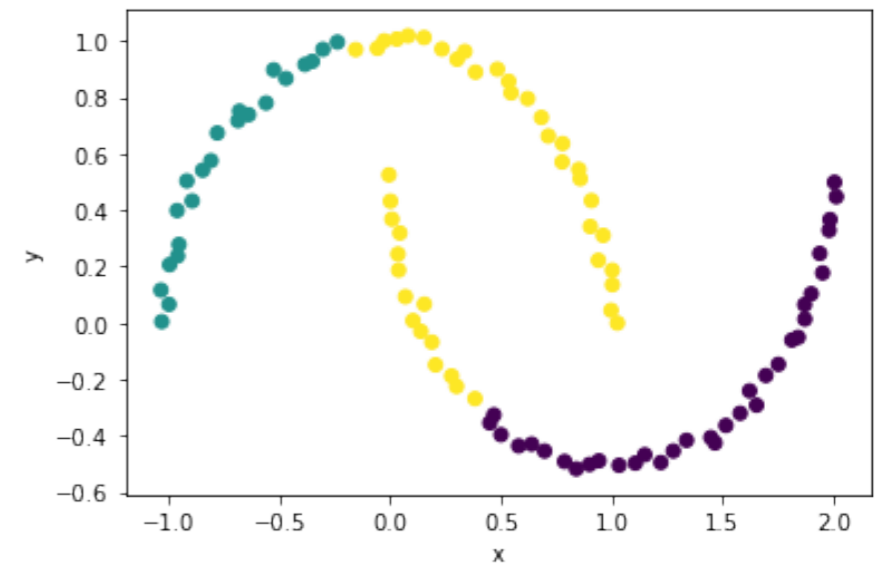
single linkage



complete linkage



average linkage



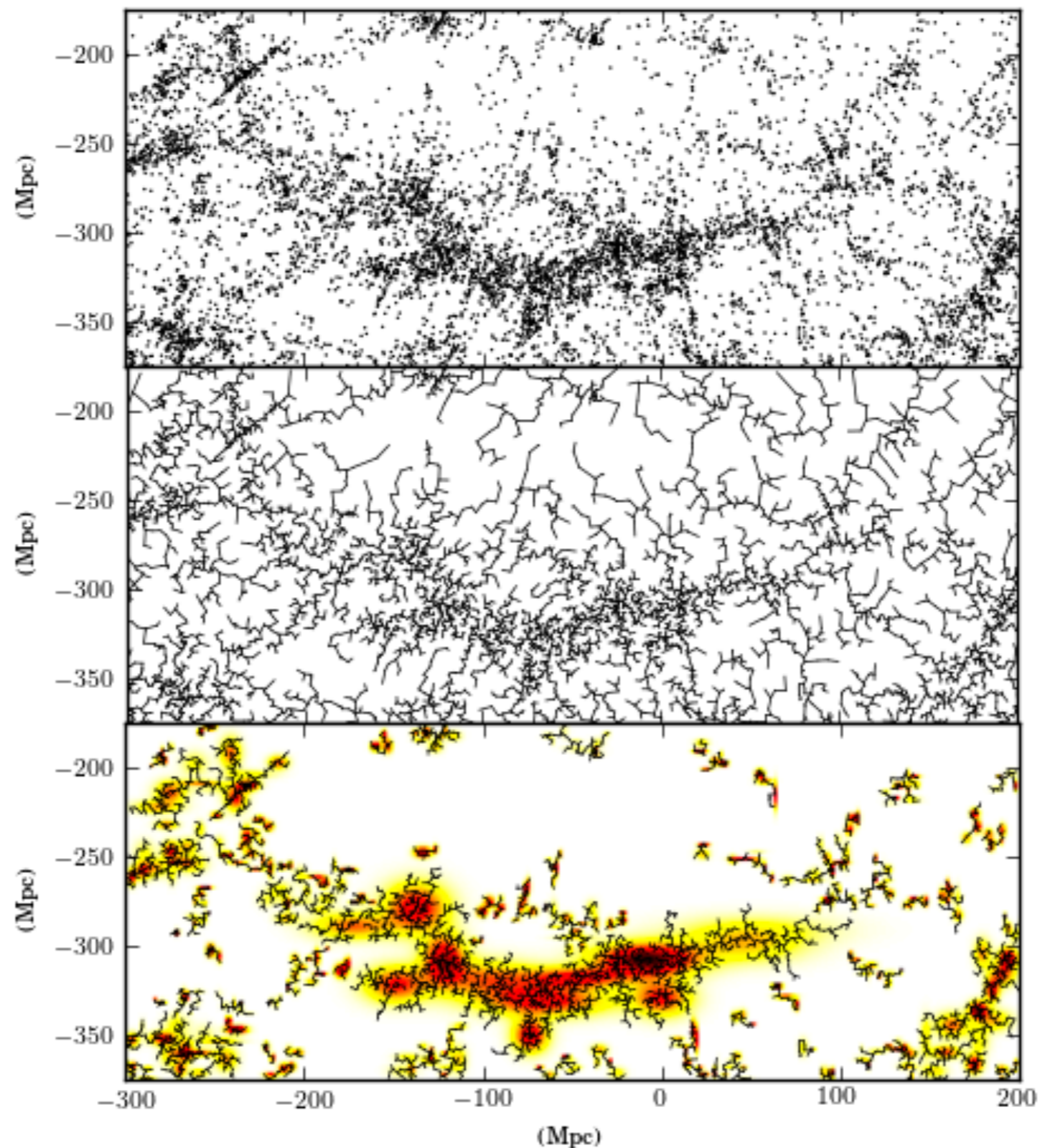
The anatomy of Hierarchical Clustering

$$f(\vec{X}, \{a_1, a_2, \dots\}) = \vec{y}$$

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

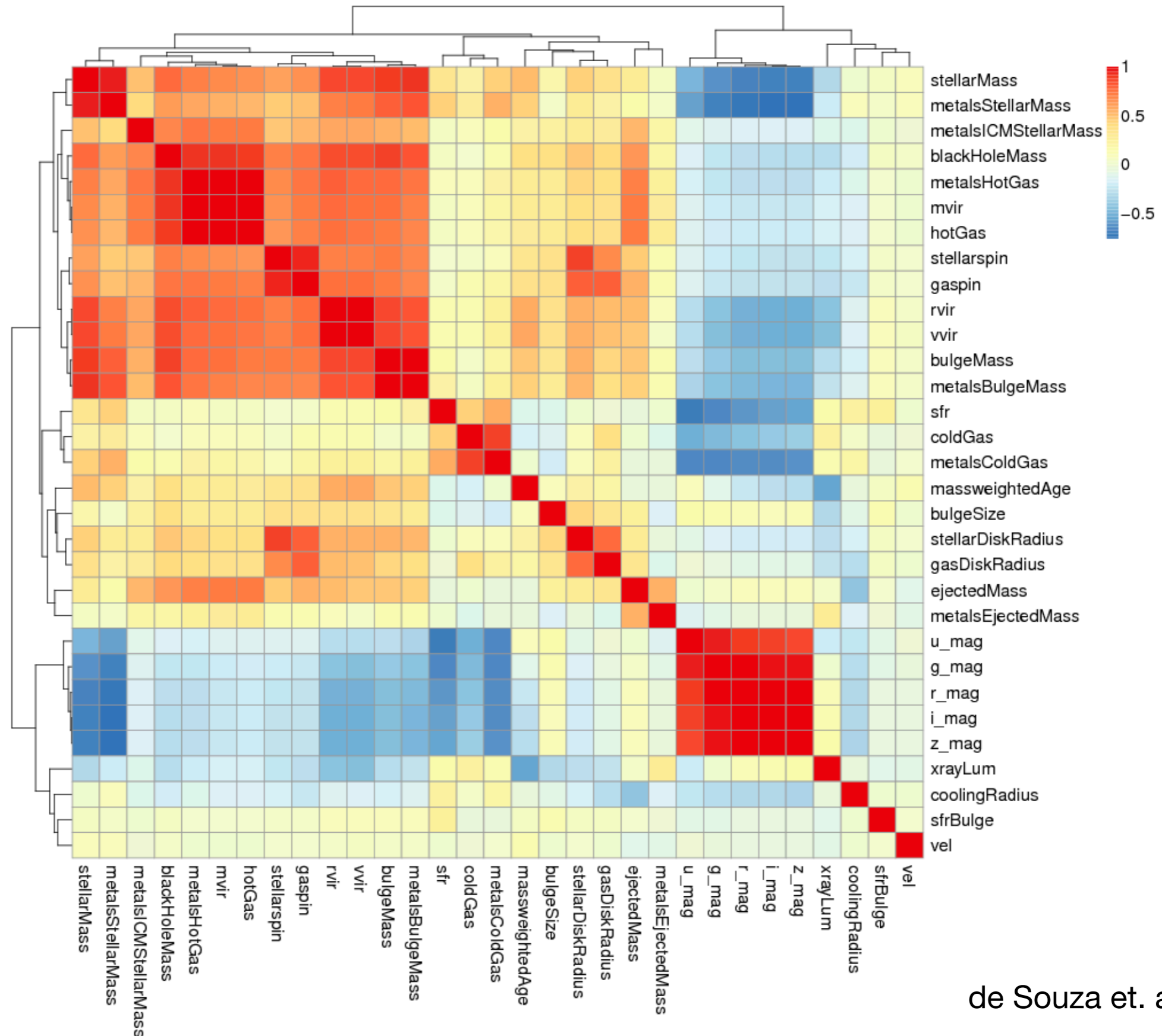
What happens if we have an outlier in the dataset?

Spatial clustering: example



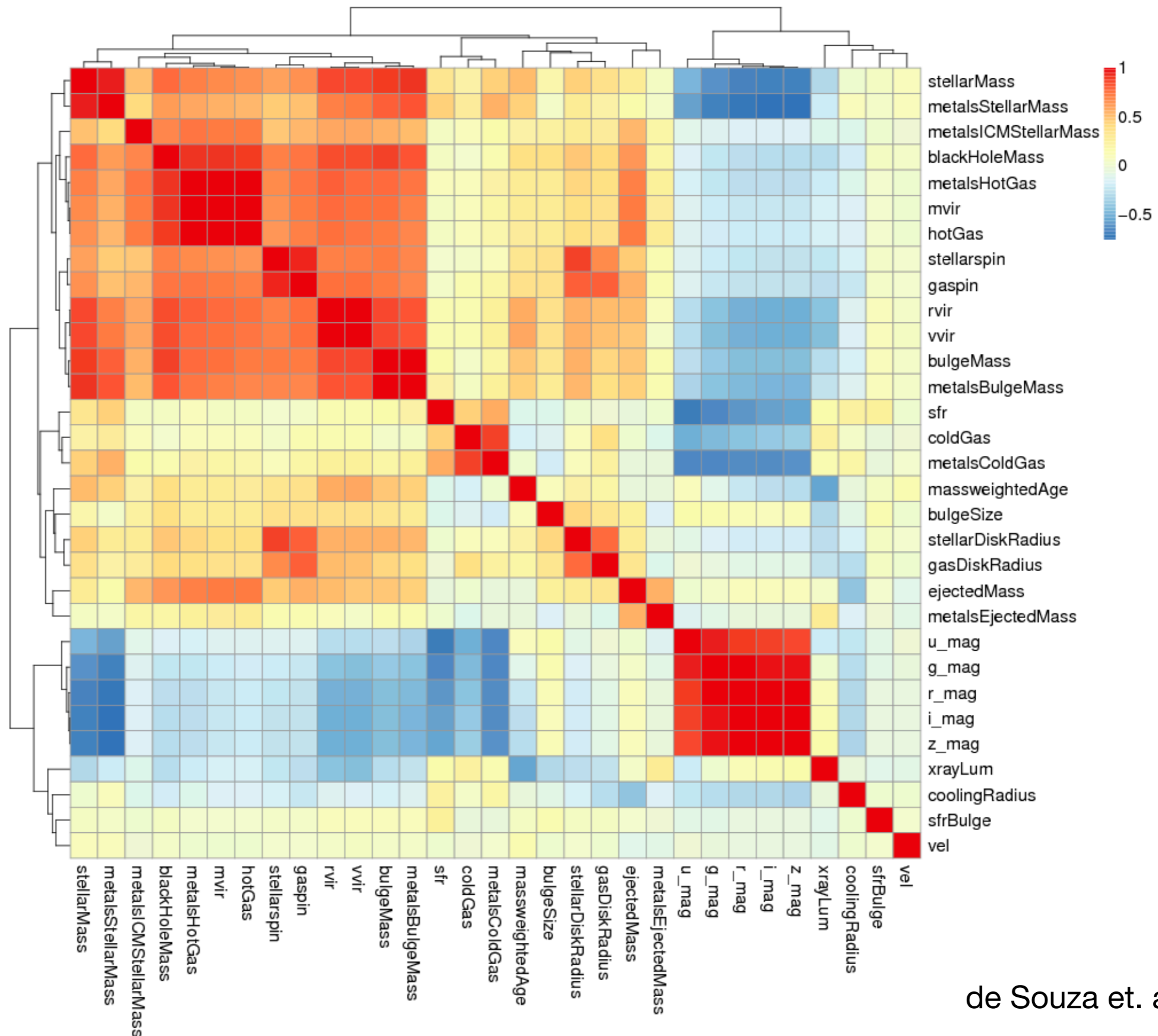
See code [here](#).

Visualizing correlations with Hierarchical Clustering



Visualizing correlations with Hierarchical Clustering

Note Viviana's
suggestion to
remove
correlated
features!



Questions?
