HALMSTAD UNIVERSITY

Learning Systems (DT8008)

Unsupervised Learning: Clustering

Dr. Mohamed-Rafik Bouguelia
mohamed-rafik.bouguelia@hh.se

Halmstad University

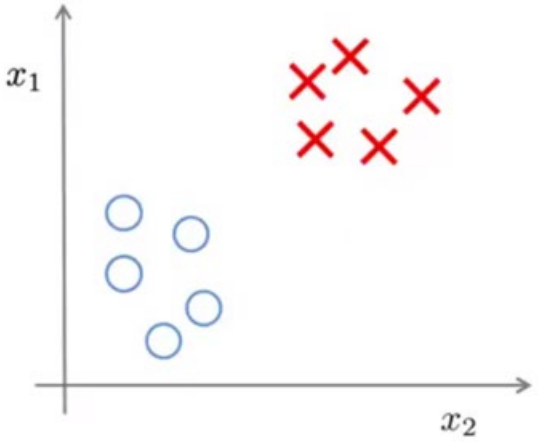
Classification ≠ Clustering

Supervised

Unsupervised

Supervised vs. Unsupervised Learning

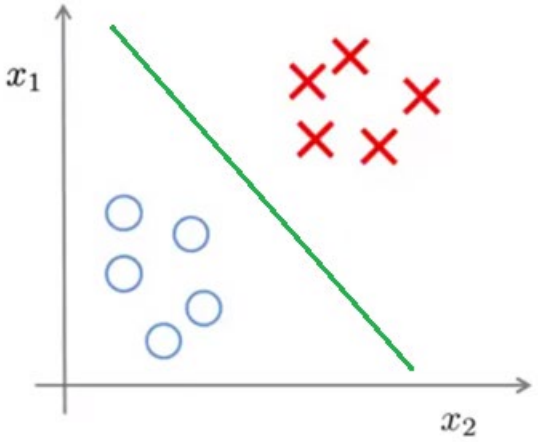
Supervised learning



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Supervised vs. Unsupervised Learning

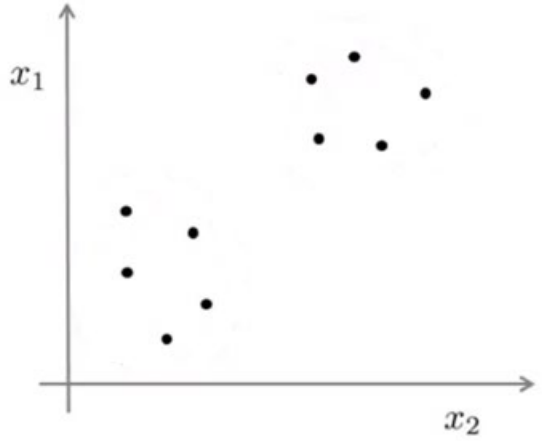
Supervised learning



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Supervised vs. Unsupervised Learning

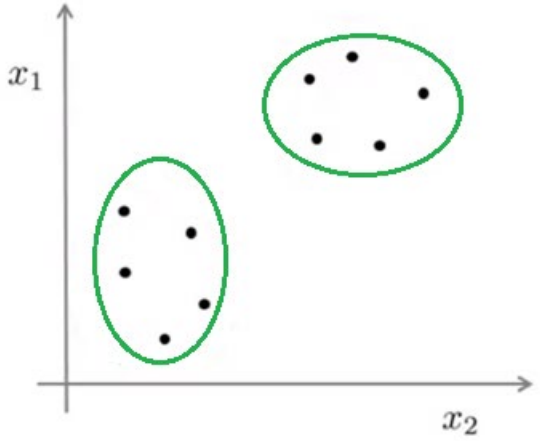
Unsupervised learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Supervised vs. Unsupervised Learning

Unsupervised learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

Supervised vs. Unsupervised Learning

Of the following examples, which would you address using an unsupervised learning algorithm? (Check all that apply.)

- ☐ Given email labeled as spam/not spam, learn a spam filter.
- ☐ Given a set of news articles found on the web, group them into set of articles about the same story.
- ☐ Given a database of customer data, automatically discover market segments and group customers into different market segments.
- ☐ Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

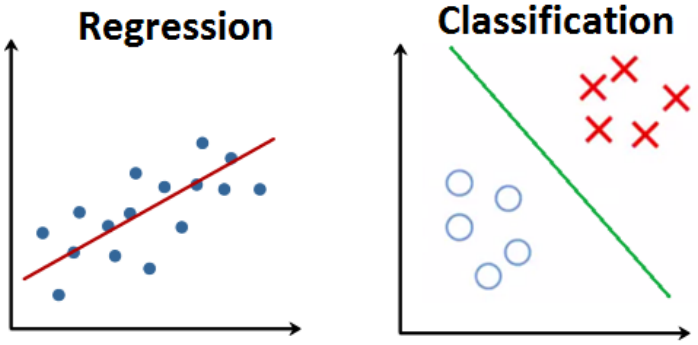
Supervised vs. Unsupervised Learning

Of the following examples, which would you address using an unsupervised learning algorithm? (Check all that apply.)

- ☐ Given email labeled as spam/not spam, learn a spam filter.
- ☒ Given a set of news articles found on the web, group them into set of articles about the same story.
- ☒ Given a database of customer data, automatically discover market segments and group customers into different market segments.
- ☐ Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

Supervised vs. Unsupervised Learning

- Supervised learning uses labeled data pairs $(x^{(i)}, y^{(i)})$ to learn some function $\boldsymbol{h}$: $\boldsymbol{X} \rightarrow \boldsymbol{y}$
  - e.g. Regression, Classification



- What if we don't have labels?

- Unsupervised learning ➔ without labels (without supervision from human)
- Semi-supervised learning ➔ few labelled instances and lot of unlabeled instances.

- **Clustering** is the unsupervised grouping of data-points.
  - e.g. for exploring the data or for knowledge discovery …

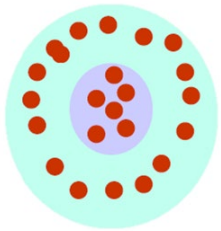
Supervised vs. Unsupervised Learning

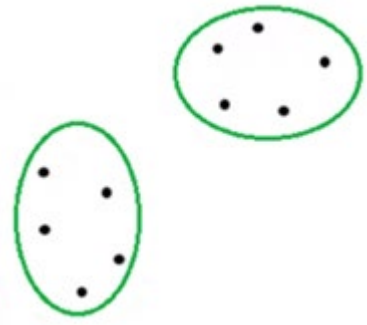
- Example of unsupervised learning tasks:
  - Clustering
  - Outlier and Anomaly detection
  - Dimensionality Reduction (e.g. with PCA)
  - Unsupervised feature learning (e.g. with Autoencoders).

- We want to explore the data to find some intrinsic structures in them.
  - **Clustering** is when the clusters are **not known**
  - If the clusters were **known** (i.e. they are known classes) and the problem was to place a new instance into the proper class, then this is **classification** (not clustering)

- It is often easier to obtain unlabeled data than labeled data (which requires human intervention).

Clustering



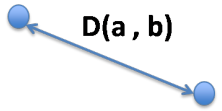


- Clustering is grouping data points such that
  - Instances within each cluster are similar to each other (**minimize intra-clusters distance**)
  - Instances from different clusters are dissimilar (**maximize inter-clusters distance**)

- Goal is to discover interesting structures (unknown subgroups in the data).

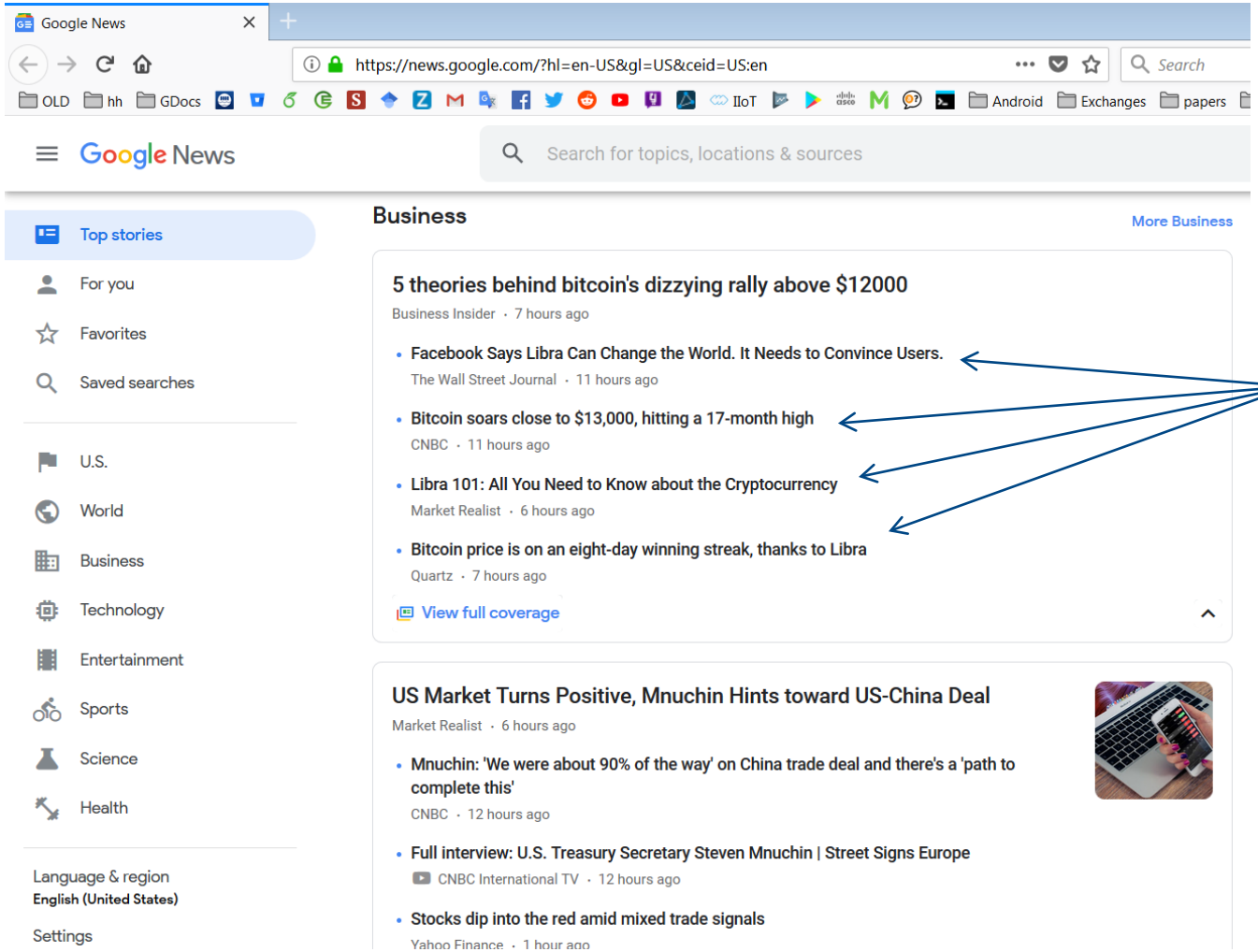
- The dissimilarity is defined using a distance measure
  - e.g. The Euclidean distance

$$D(a,b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2}$$



- Examples:
  - Groups of breast cancer patients grouped by their gene expression measurements
  - Groups of shoppers characterized by their browsing and purchase histories
  - Movies grouped by the ratings assigned by movie viewers.
  - … etc.

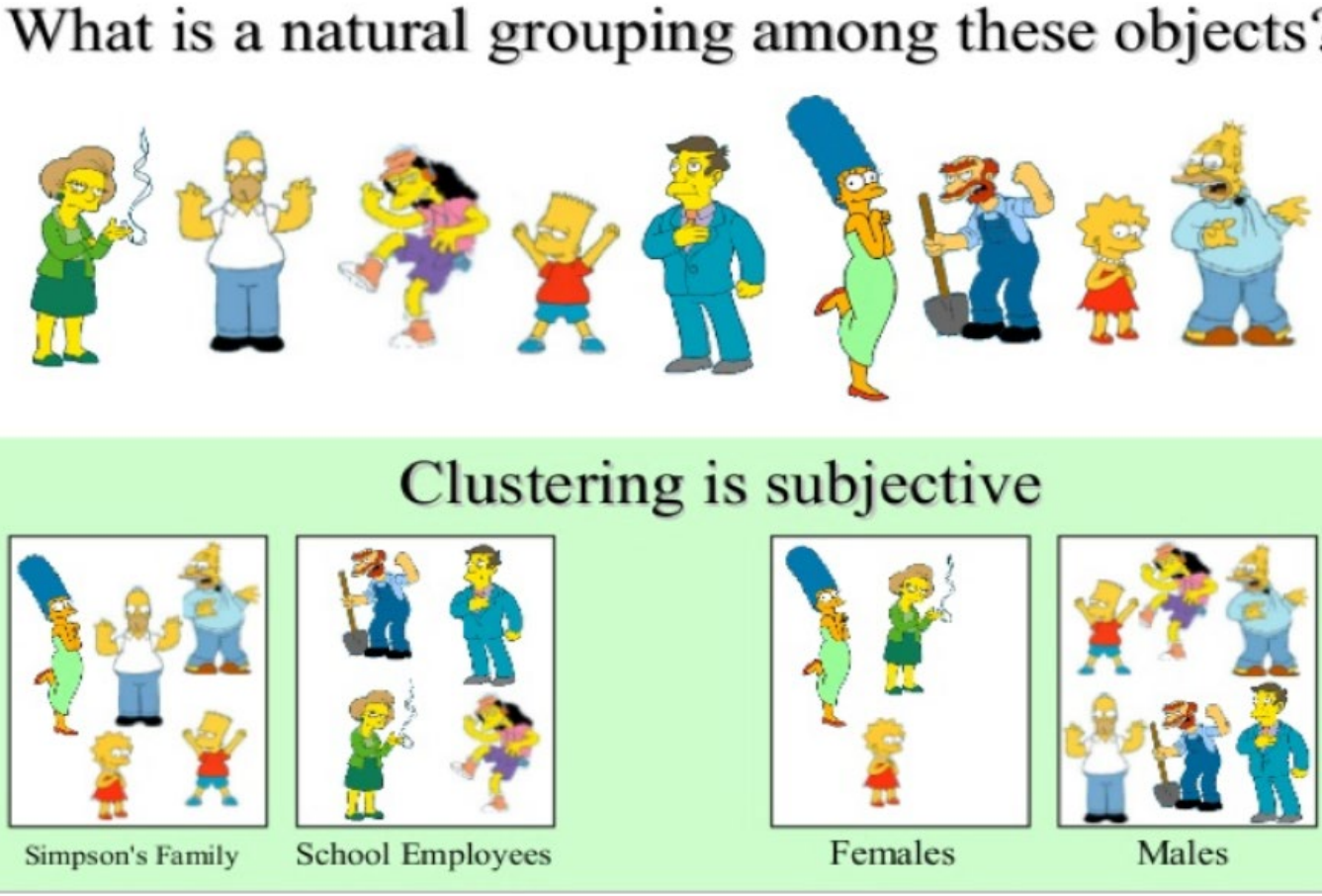
Example application of clustering



Automatically grouping together the stories (news articles on the Web) that talk about the same topic.

Clustering is "subjective"

- **Unsupervised** learning (e.g clustering) is more subjective than **supervised** learning.



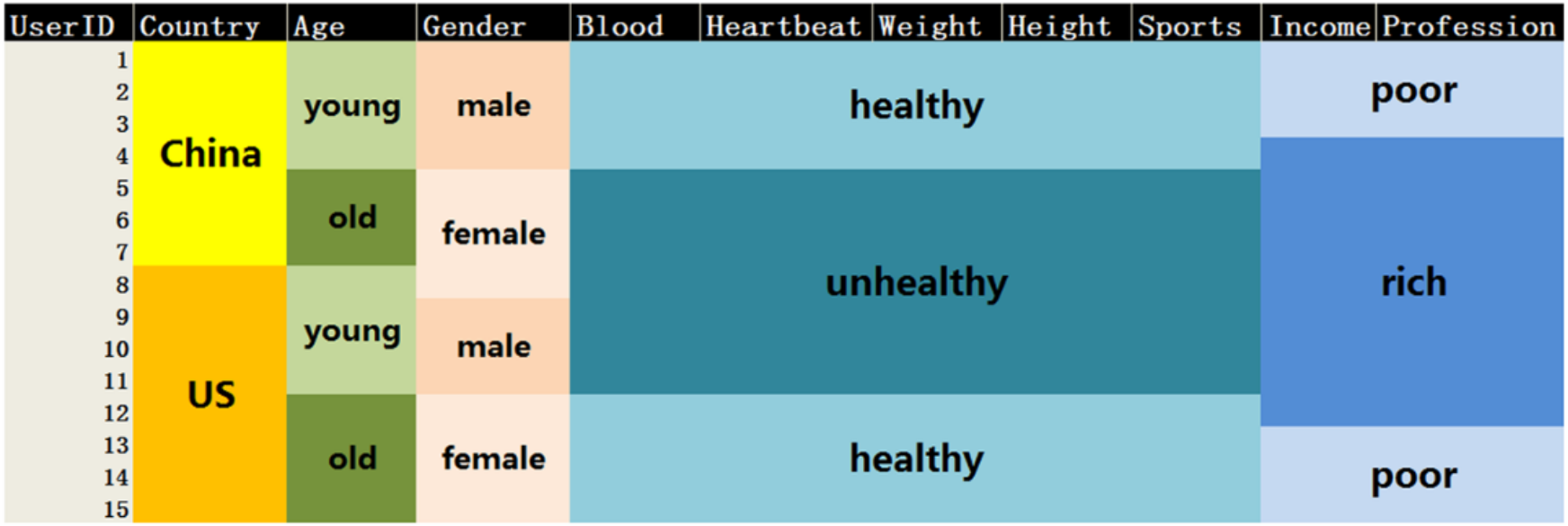
One possible clustering (with two clusters).

Another possible clustering (with two clusters)

Clustering is "subjective"

- **Unsupervised** learning (e.g clustering) is more subjective than **supervised** learning.

People can be clustered by different criteria



Distance metrics

Distance functions

Note: a "*distance*" (i.e. dissimilarity) is the inverse of "*similarity*".

- Let $x \in R^d$ and $z \in R^d$ be two data-points (vectors of dimension d).
  - Minkowski distance (a general distance function) :

$$\|x - z\|_p = \left[\sum_{i=1}^{d} (x_i - z_i)^p\right]^{\frac{1}{p}}$$

  - Euclidean distance (i.e. when $p = 2$) :

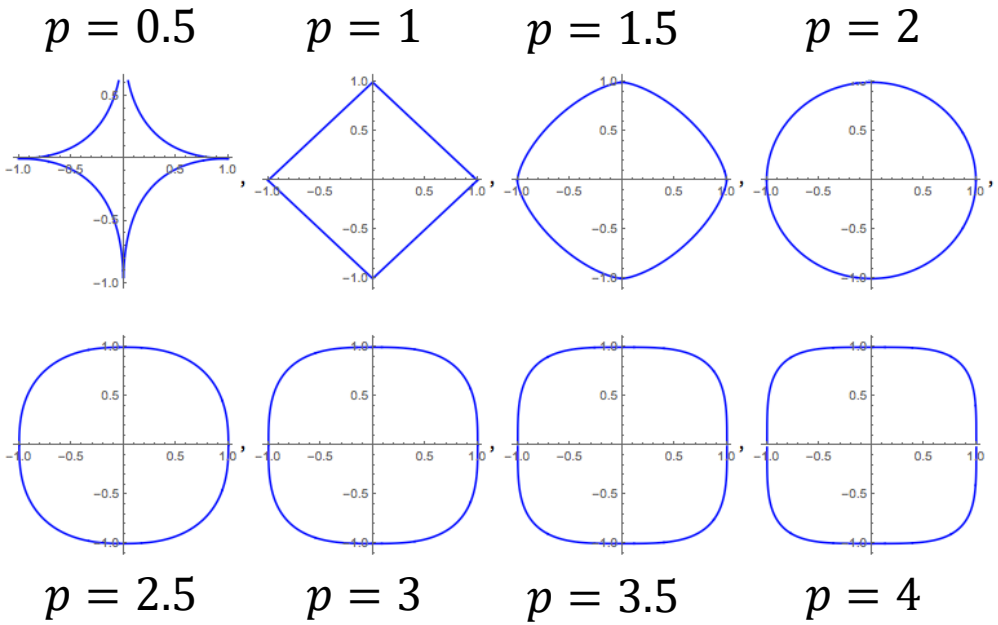
$$\|x - z\|_2 = \sqrt{\sum_{i=1}^{d} (x_i - z_i)^2}$$

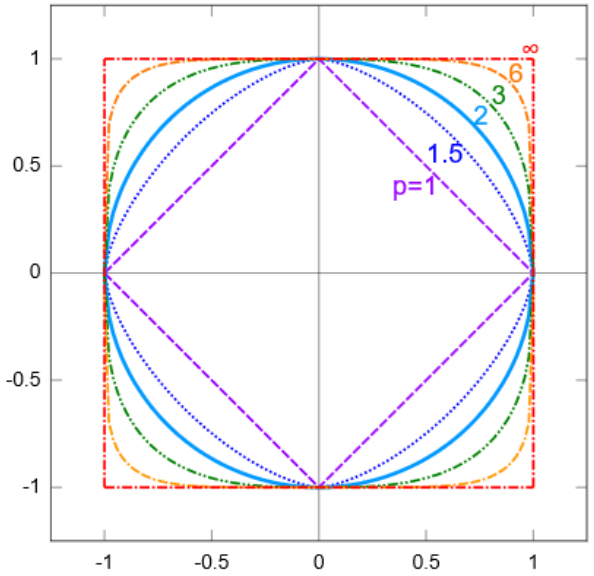
  - Manhattan distance (i.e. when $p = 1$) :

$$\|x - z\|_1 = \sum_{i=1}^{d} |x_i - z_i|$$

Norm of a vector

- Consider the vector $v = x - z$. The distance $\|x - z\|_p$ is just the p-norm of the vector v.



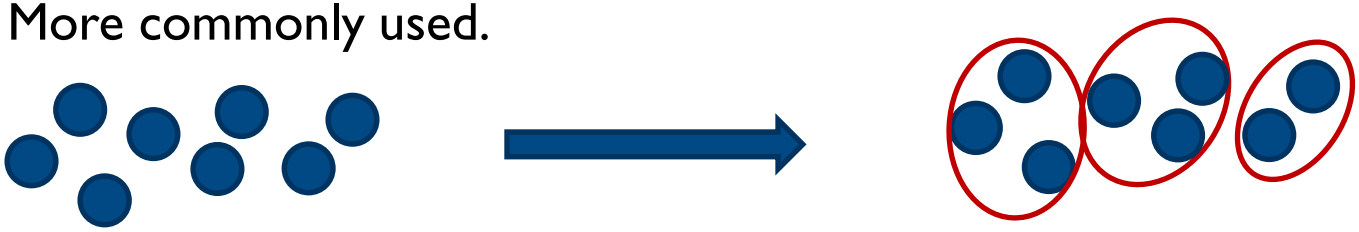


All the points on the blue curve have the same p-norm $\|v\|_p$ (Minkowski distance from v to the origin).

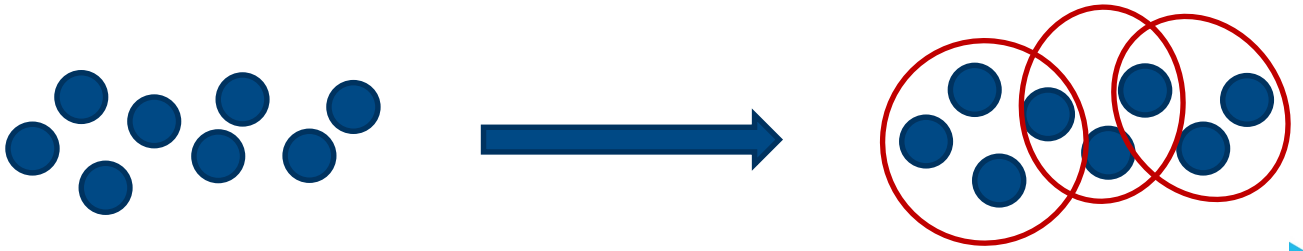
Some types of clustering techniques

Types of clustering techniques

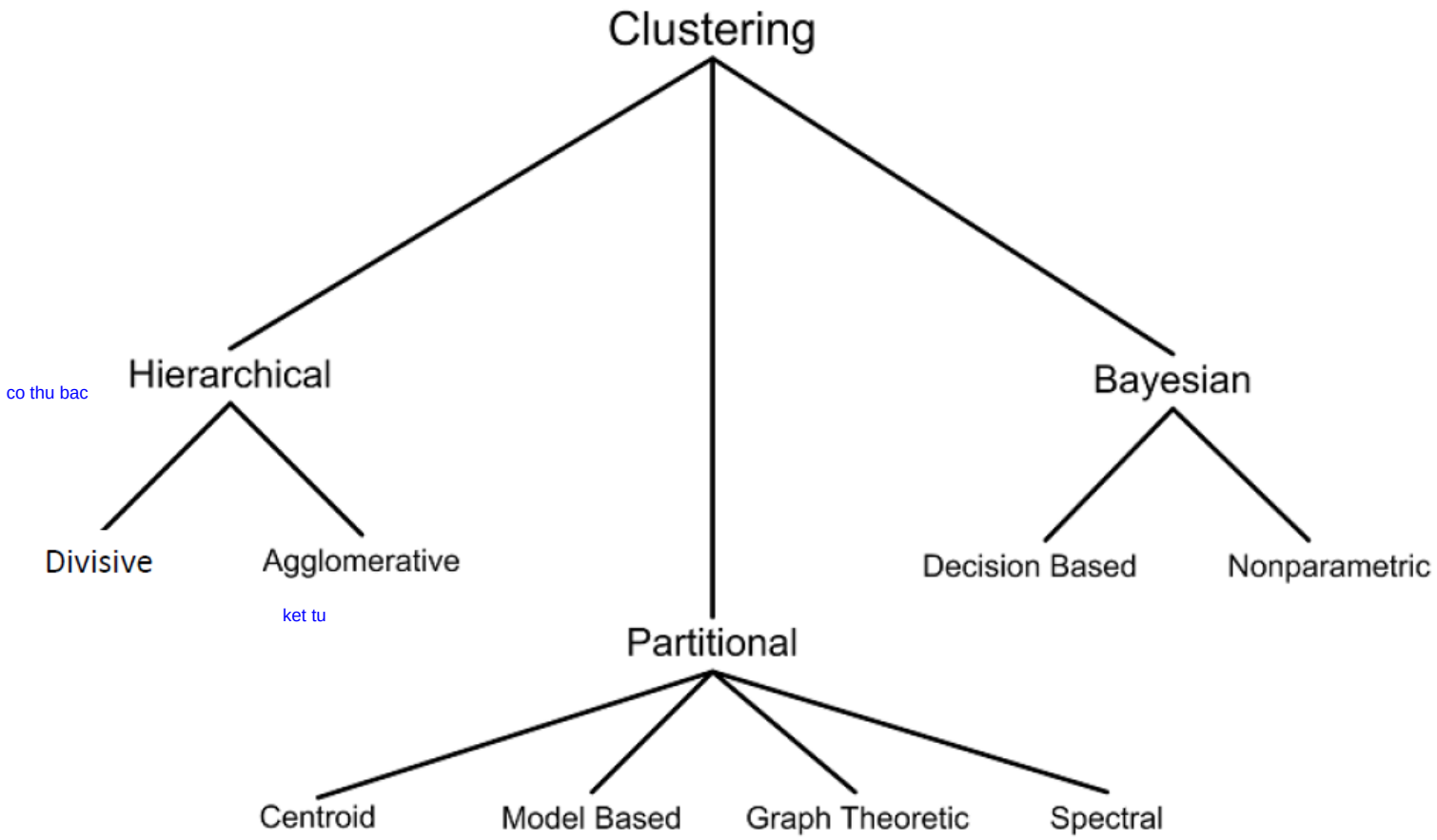
- **Hard clustering** (focus of this lecture): each data belongs to only one cluster.
  - More commonly used.



- **Soft clustering**: a data can belong to more than one cluster.
  - Example: some documents belongs to multiple topics
  - Membership to clusters is modeled as a probability distribution



Types of clustering techniques



co thu bac

ket tu

Types of clustering techniques

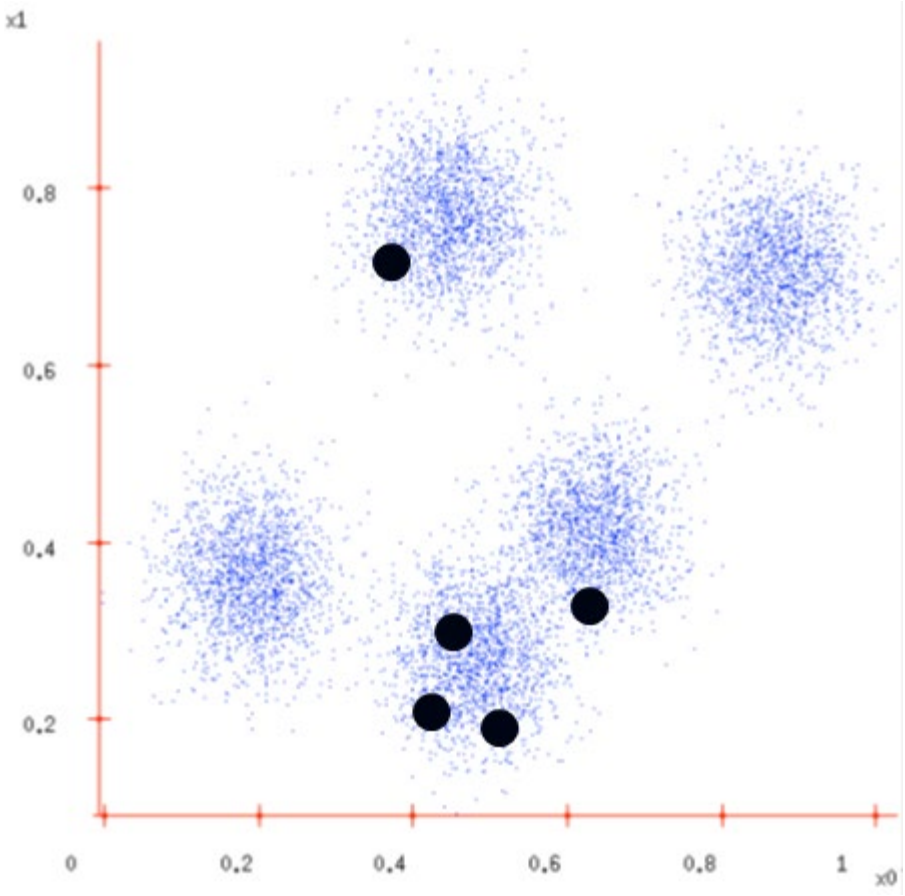
- **Partitional** algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.

- **Hierarchical** algorithms find successive clusters using previously established clusters. These algorithms can be either agglomerative ("*bottom-up*") or divisive ("*top-down*"):
  1. Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters;
  2. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

The K-means clustering method

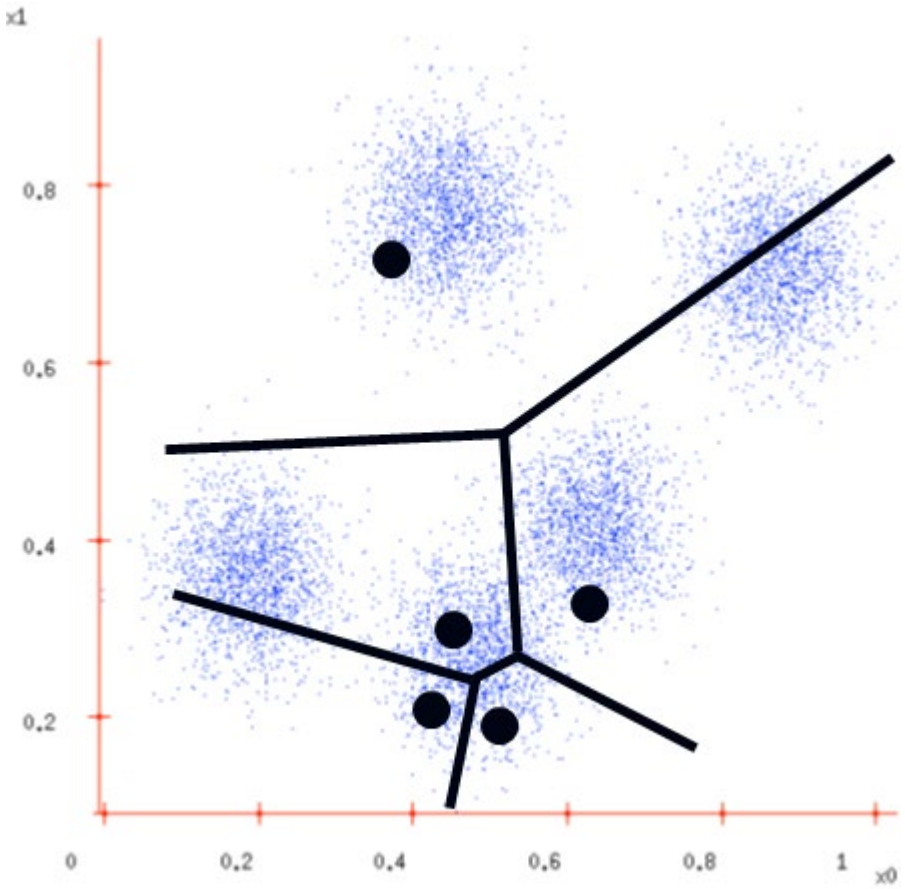
K-Means (algorithm)

- Input: data, K
  - Randomly choose K cluster centers (centroids).
  - Loop until convergence
    - Assign each point to the cluster of the closest centroid.
    - Re-estimate the cluster centroids based on the data assigned to each.



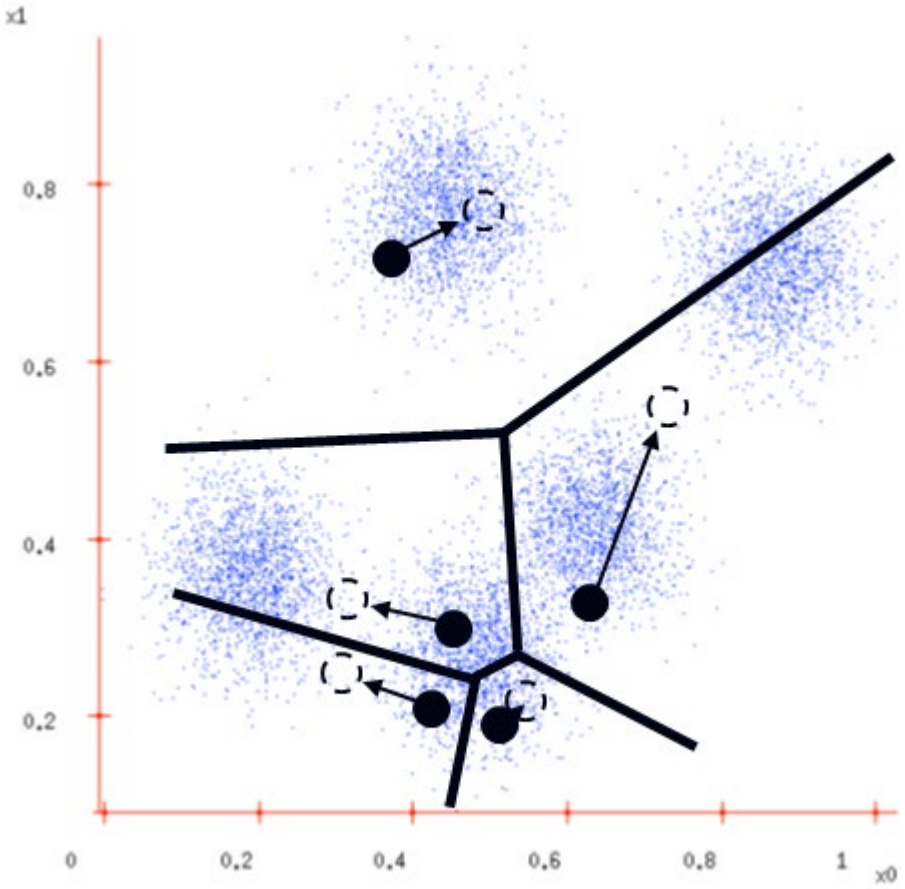
K-Means (algorithm)

- Input: data, K
  - Randomly choose K cluster centers (centroids).
  - Loop until convergence
    - Assign each point to the cluster of the closest centroid.
    - Re-estimate the cluster centroids based on the data assigned to each.



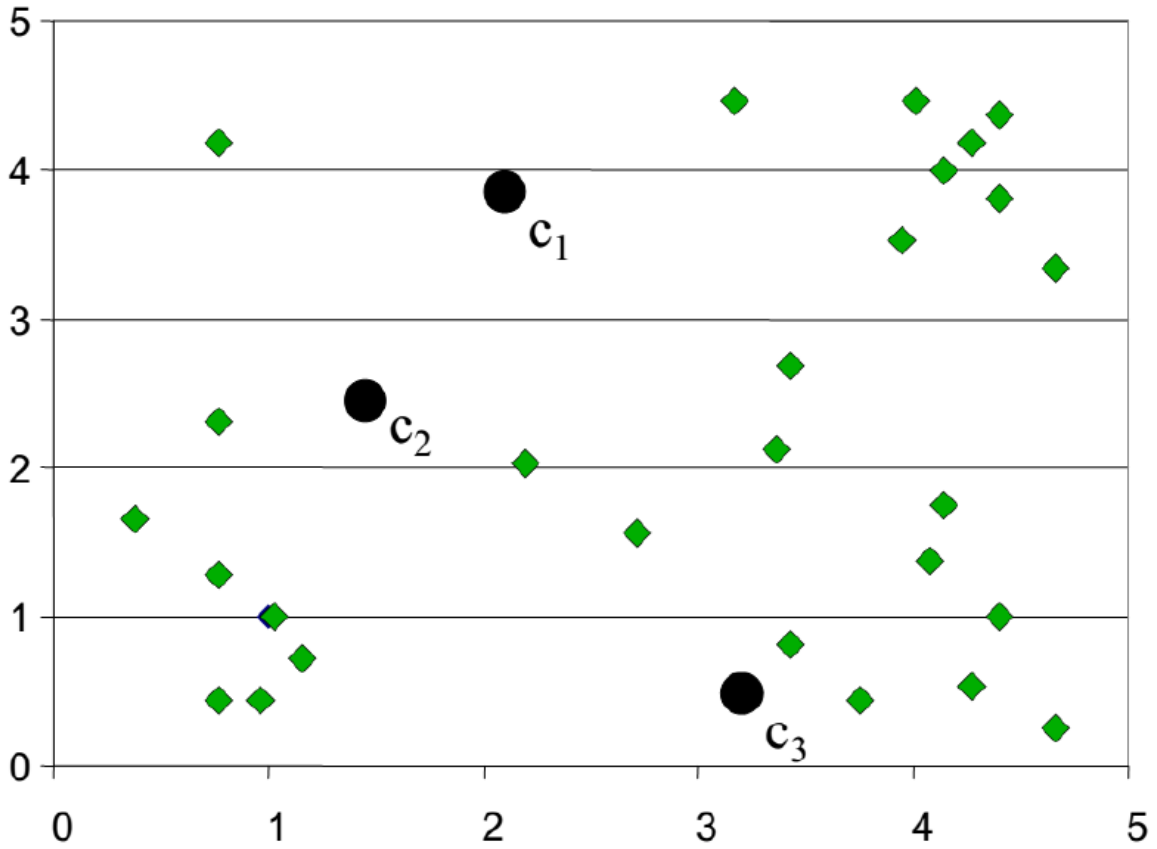
K-Means (algorithm)

- Input: data, K
  - Randomly choose K cluster centers (centroids).
  - Loop until convergence
    - Assign each point to the cluster of the closest centroid.
    - Re-estimate the cluster centroids based on the data assigned to each.

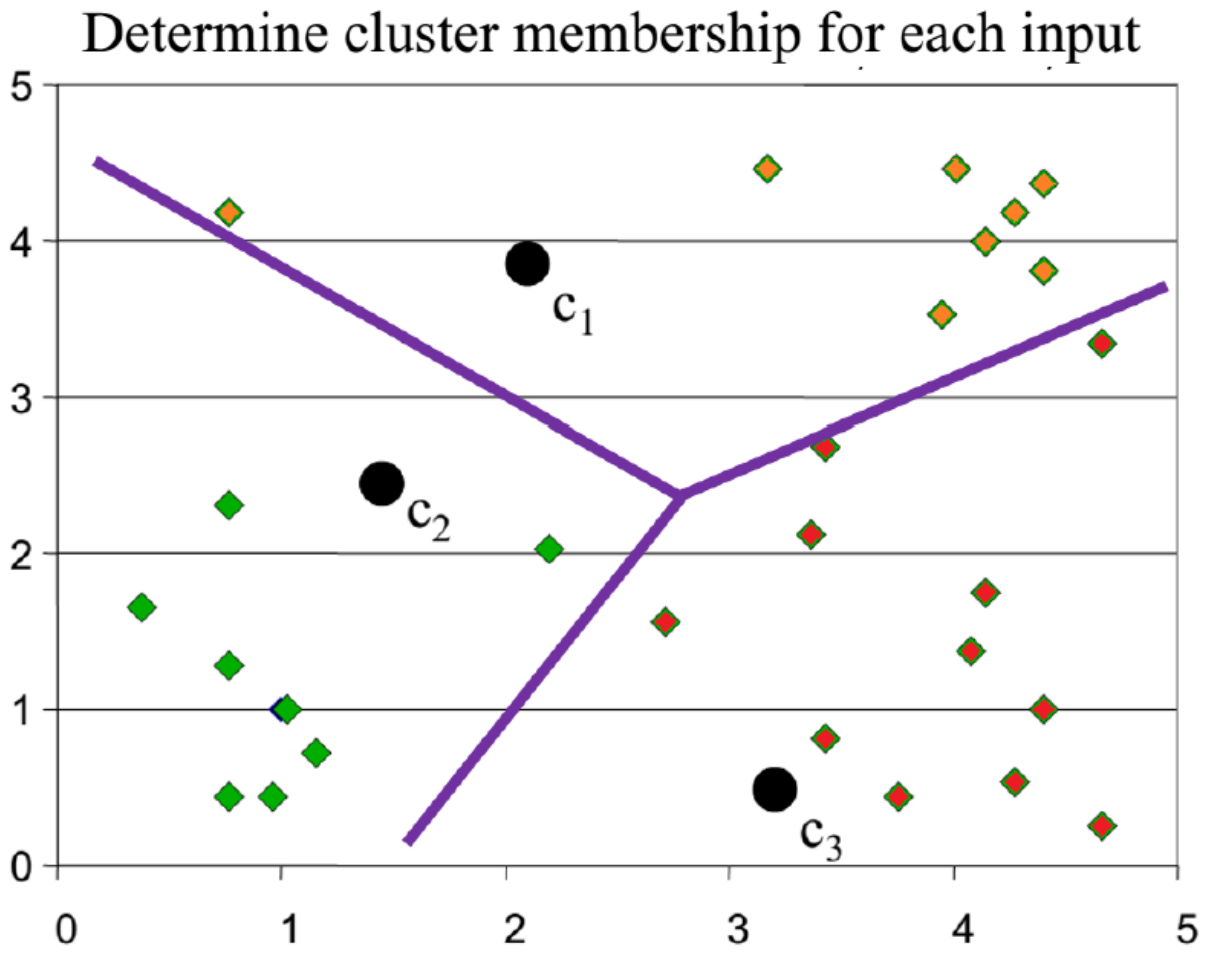


K-Means (example)

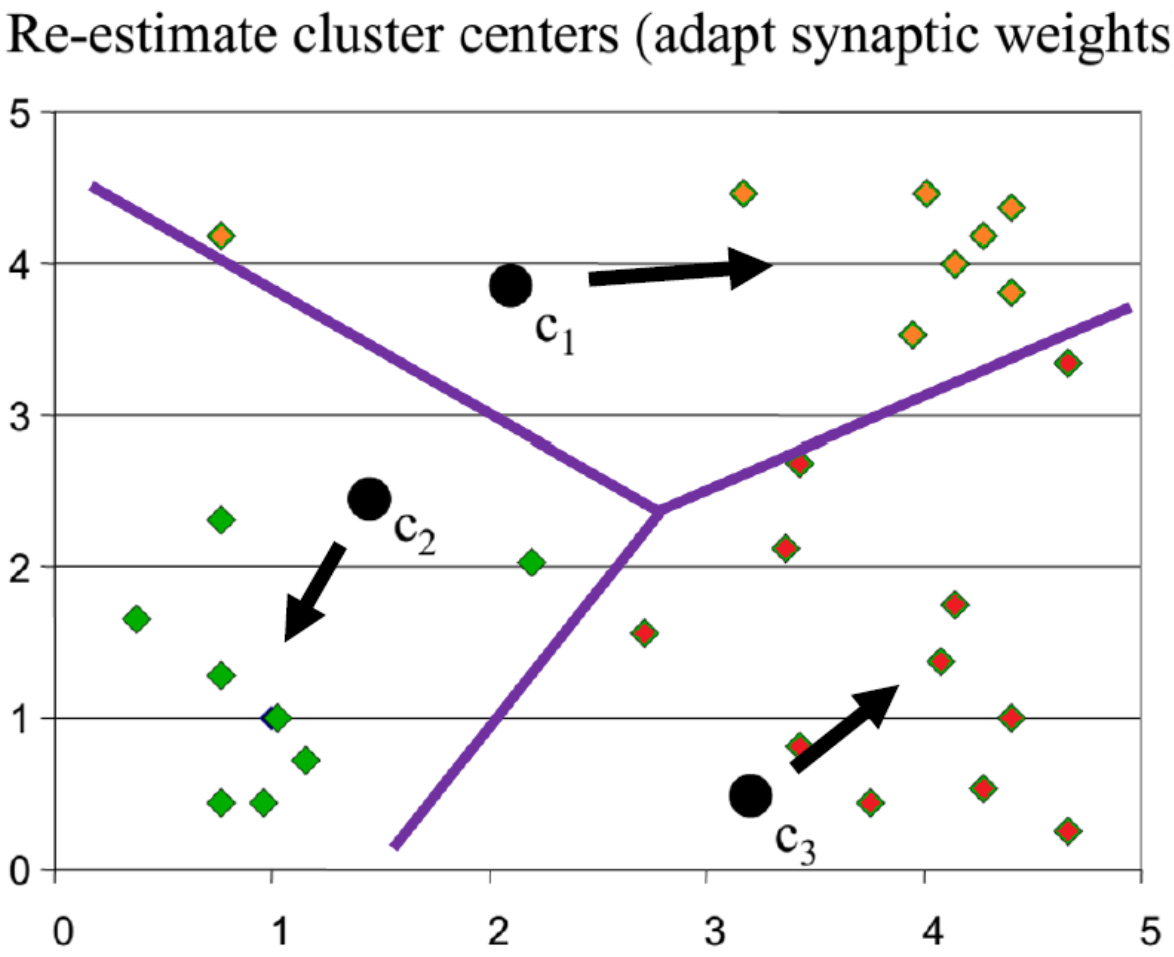
Randomly initialize the cluster centers (synaptic weights)



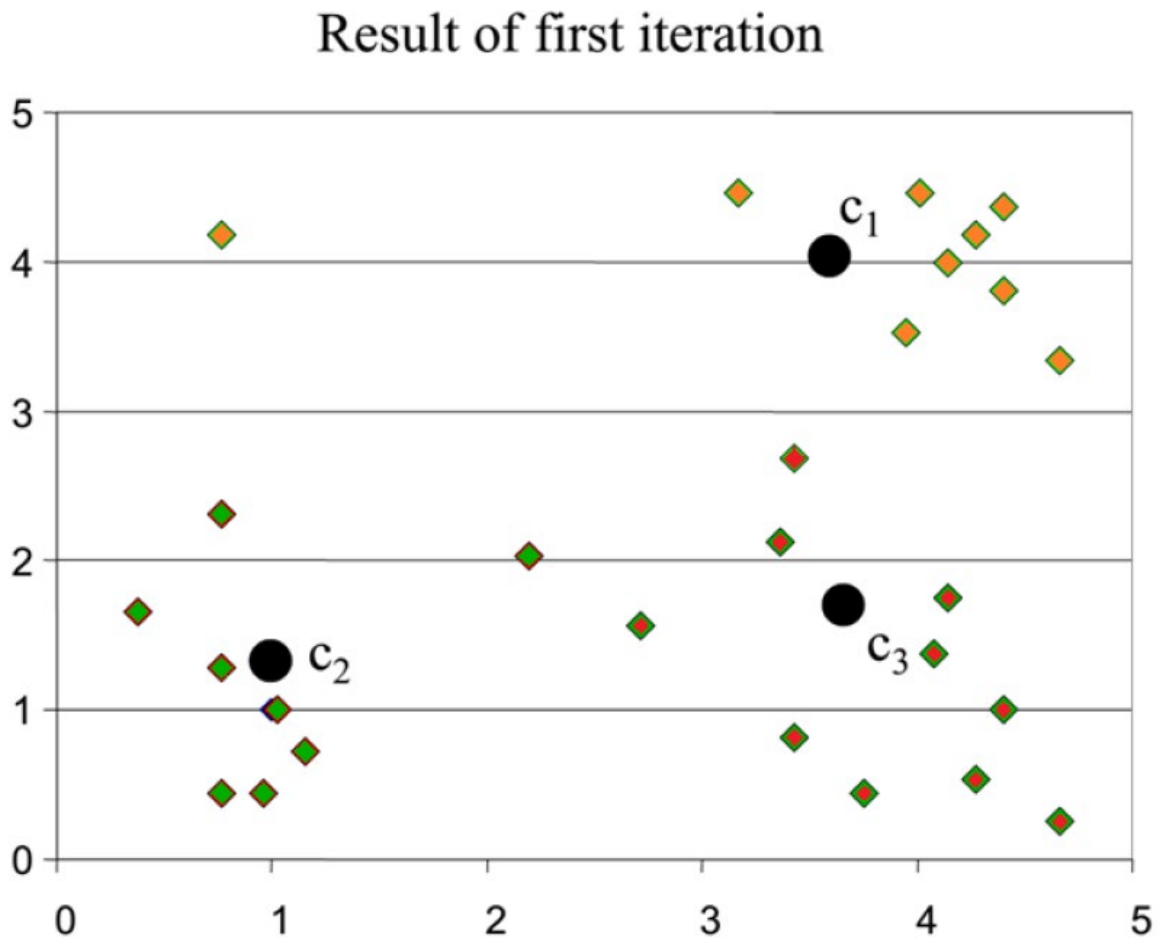
K-Means (example)



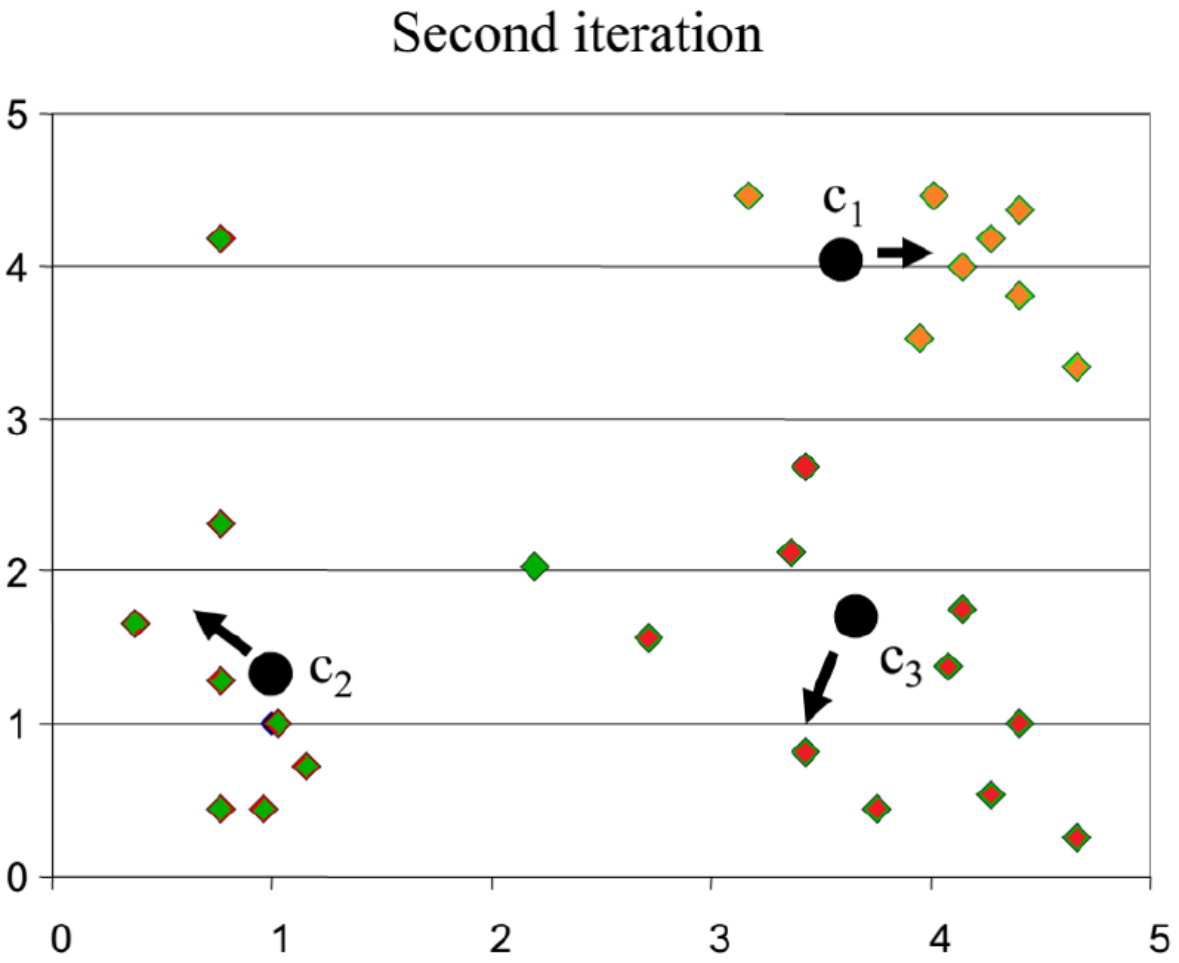
K-Means (example)



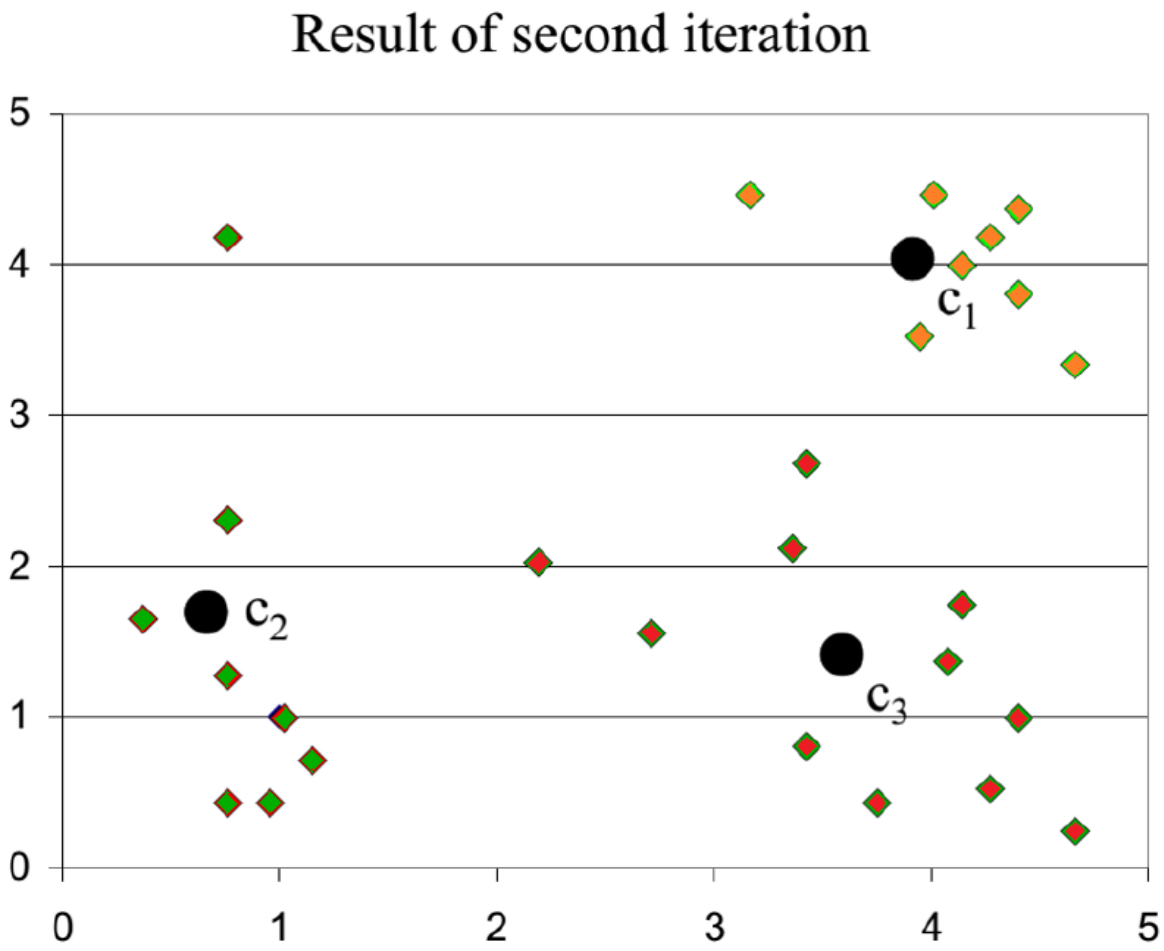
K-Means (example)



K-Means (example)



K-Means (example)



K-Means (objective/cost function)

$$\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \| \mathbf{x} - \boldsymbol{\mu}_i \|^2$$

Sum over clusters

Sum over samples in the cluster

Sample

Cluster center

K-Means (strengths)

- Simple: easy to understand and to implement
- K-means is the most popular clustering algorithm.

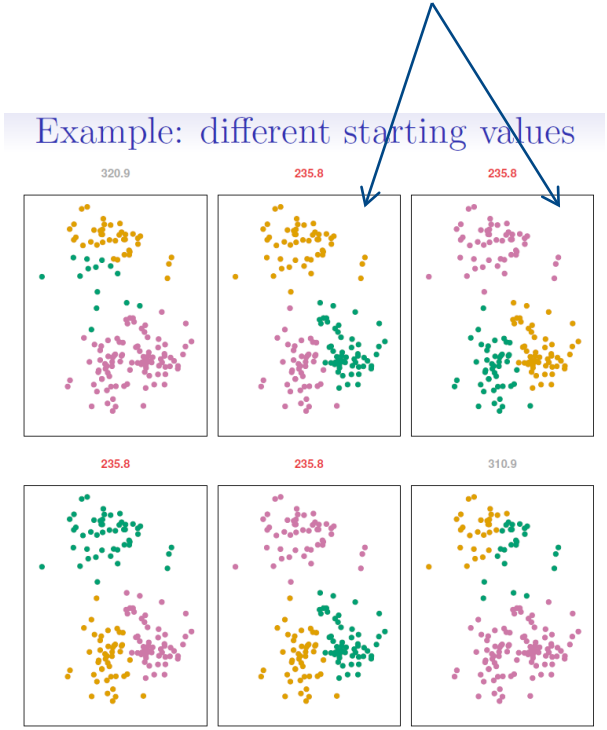
- Efficient: Time complexity: $O(tknd)$, where
  - n is the number of data points,
  - d is the number of features (dimensionality),
  - k is the number of clusters,
  - t is the number of iterations.

  ➔ Fast enough, since d, k and t are usually small.

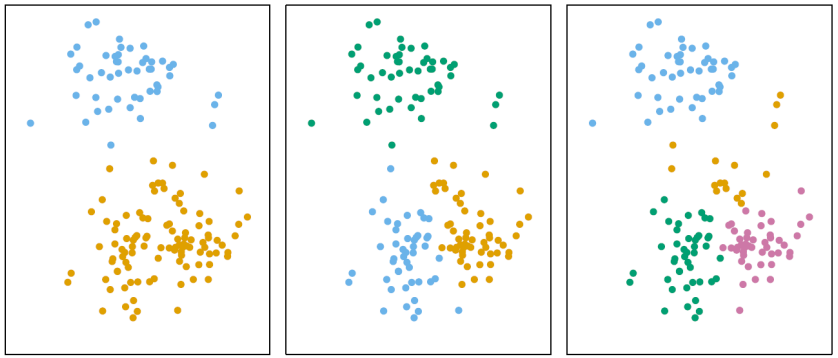
- Note:
  - It can end-up at a local optimum (not a convex optimization problem).

K-Means (weaknesses)

These two are the same results. Colors don't matter here (unsupervised).



- Very sensitive to the initial centroids.
  - To address this, we can:
    - Run k-means several times; each time with different initial centroids.
    - Seed the centroids using a better method than random (e.g. choose them to be initially as far as possible from each others).
- User must manually choose k.
- Sensitive to outliers (why?)
  - Points very far away from the others.

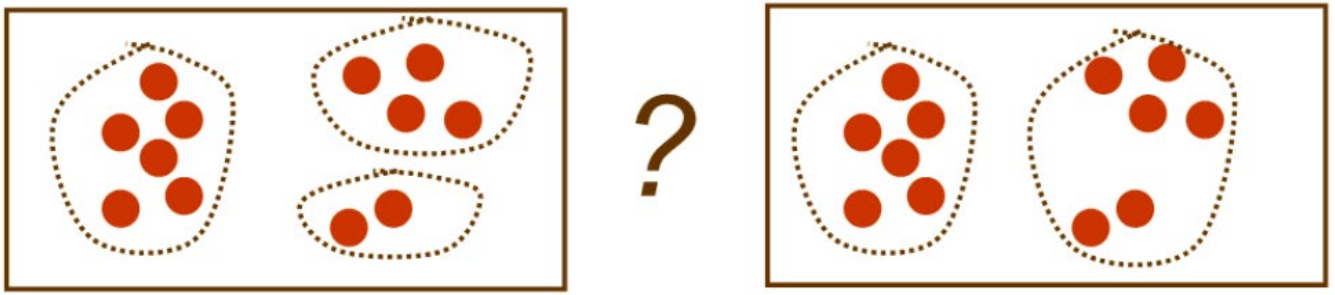


$k = 2$ $k = 3$ $k = 4$

Hierarchical clustering

Hierarchical clustering

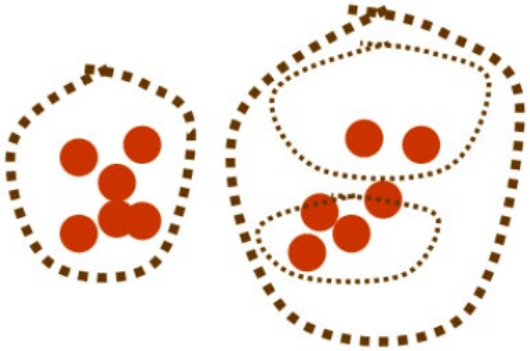
Up to now, considered "flat" clustering



It's difficult to decide on the number of clusters beforehand.

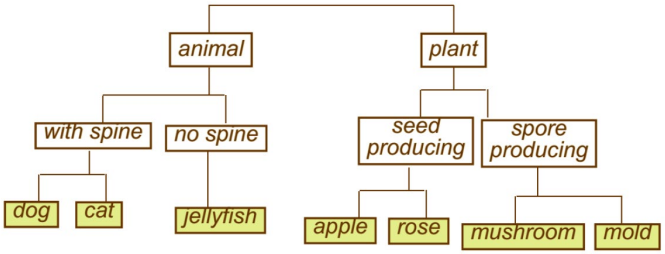
For some data, hierarchical clustering is more appropriate than "flat" clustering

Hierarchical clustering



Hierarchical clustering

- We do not know in advance how many clusters we want; in fact, we end up with a tree-like visual representation of the observations, called a **Dendrogram**, that allows us to view at once the clusterings obtained for each possible number of clusters (from **1** to **n**)

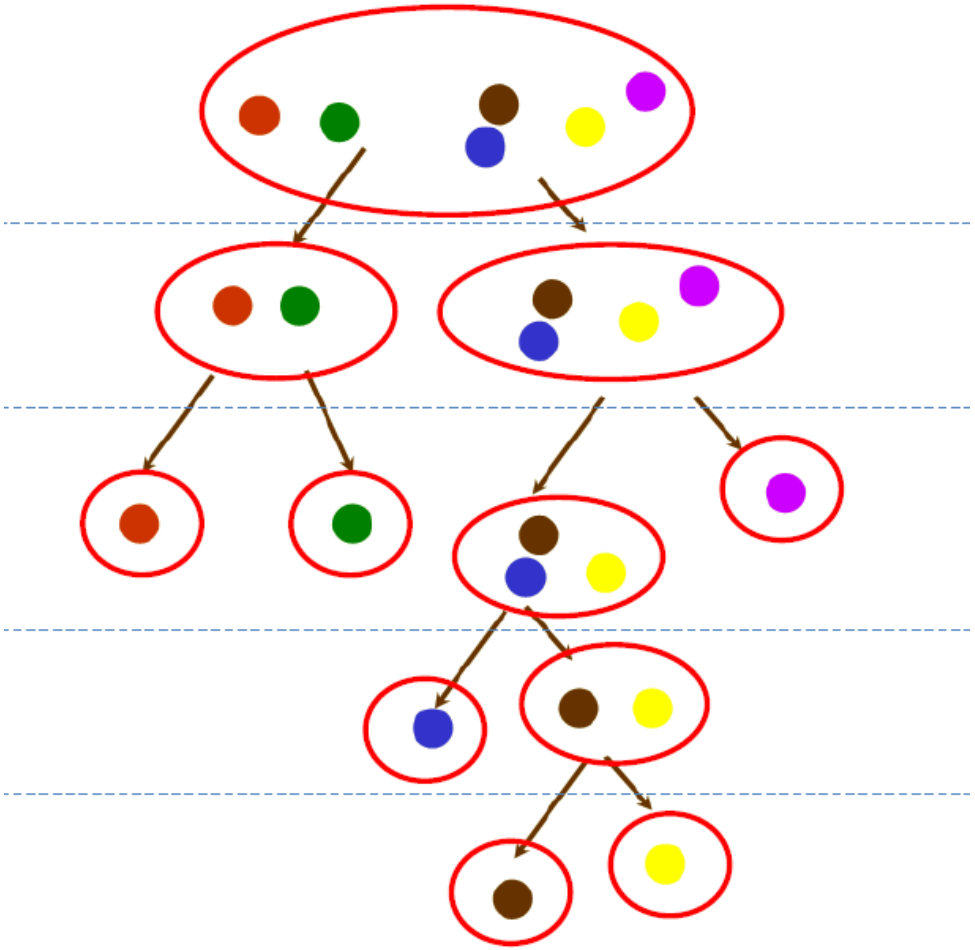


- Top-down approach:
  - start from a single cluster with all examples
  - recursively split clusters into subclusters
- Bottom-up approach:
  - start with n clusters of individual examples (singletons)
  - recursively aggregate pairs of clusters

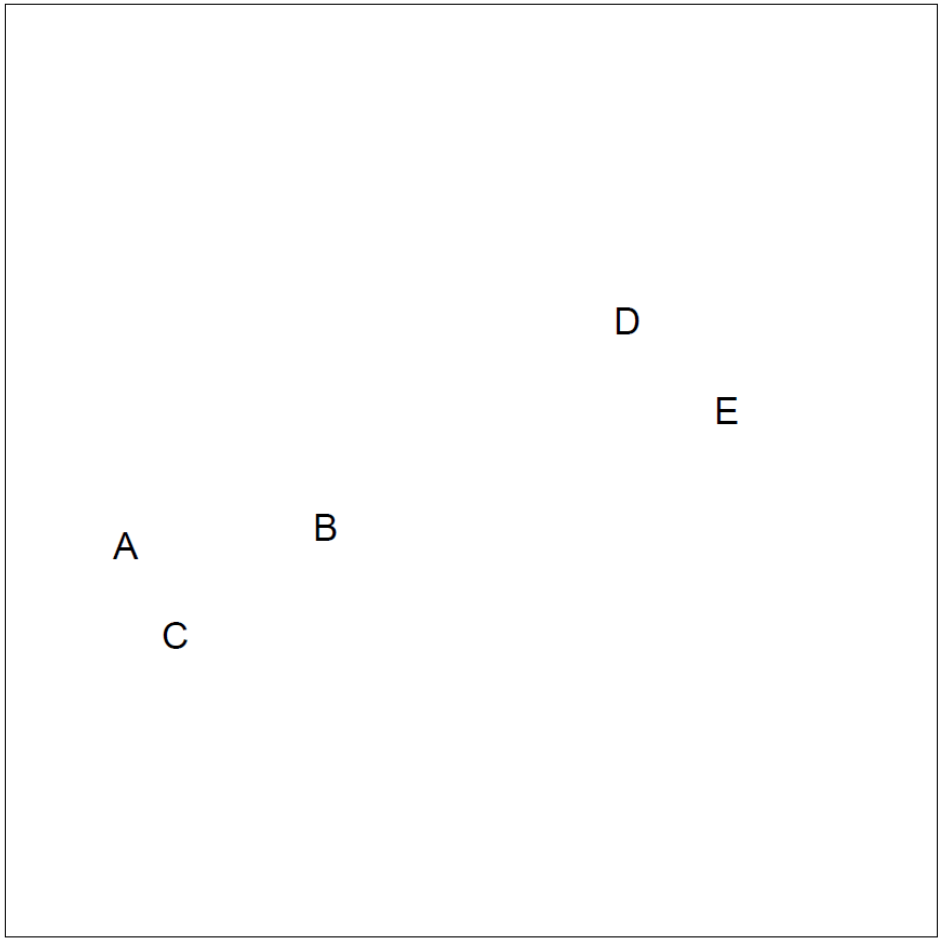
- Hierarchical (agglomerative) clustering
  - Initially, each point in cluster by itself.
  - Repeatedly combine the two "nearest" clusters into one.

Hierarchical clustering (divisive)

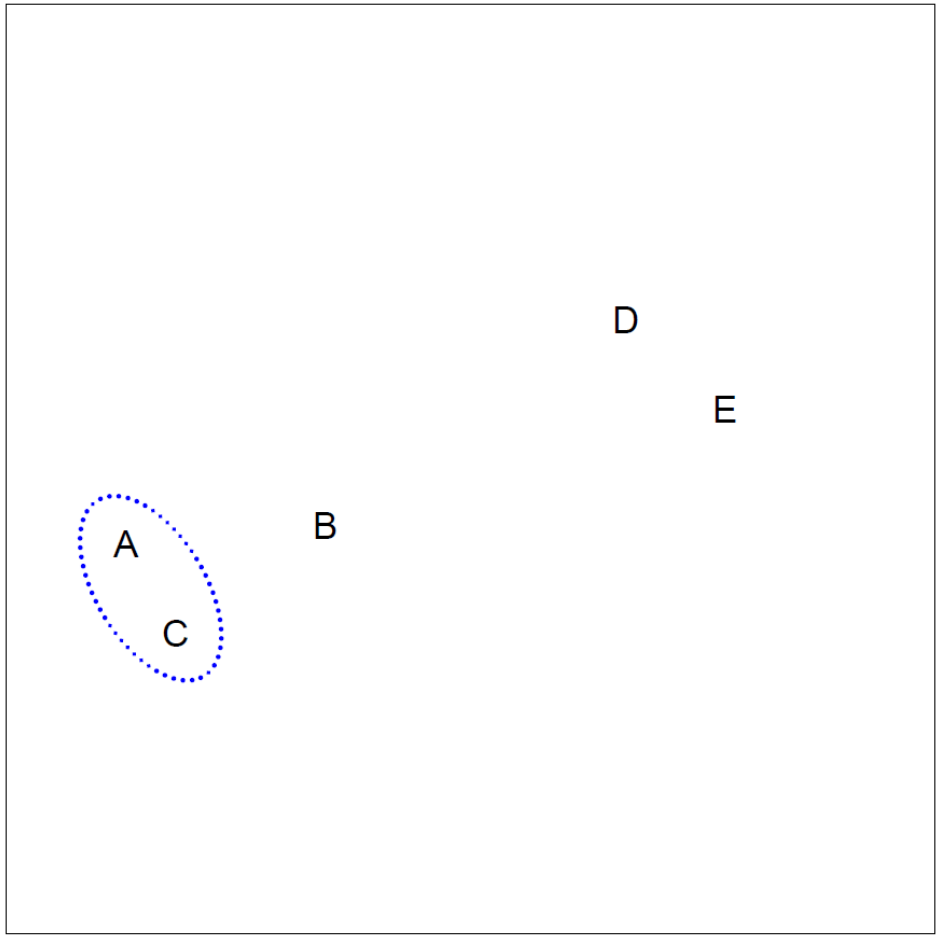
- Any flat clustering algo can be used
  - e.g. k-means with k=2 repeated 5 times.



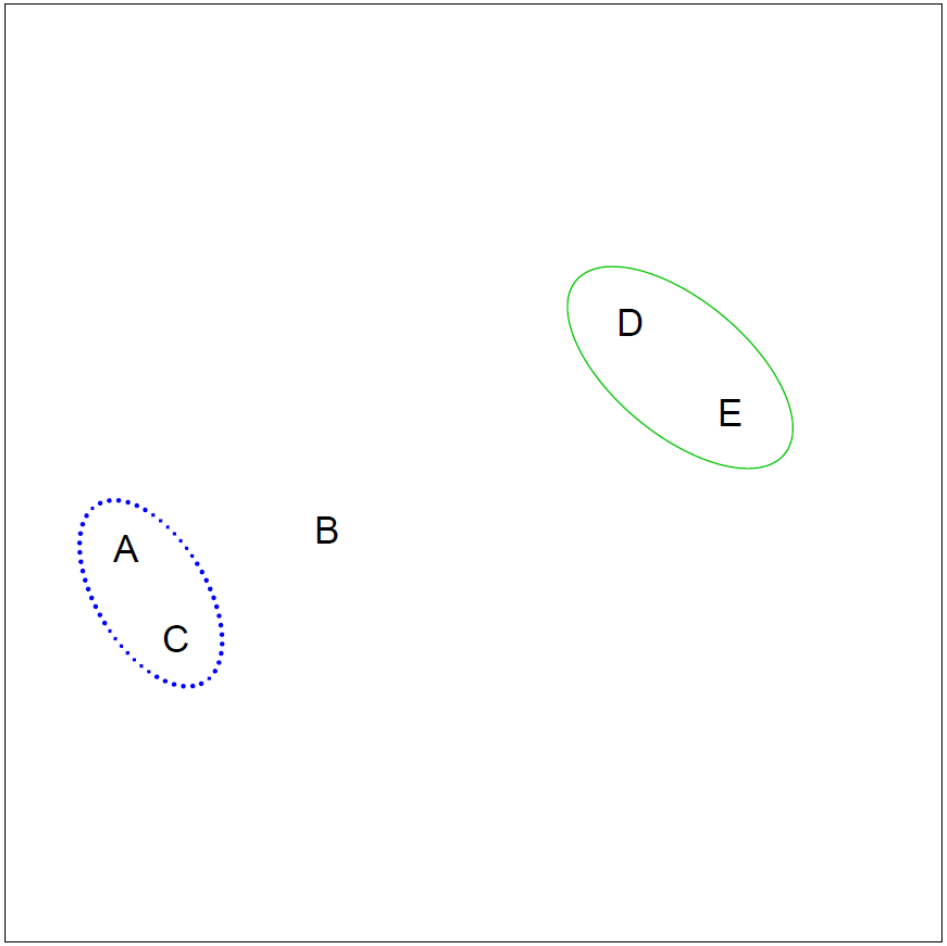
Hierarchical clustering (agglomerative)



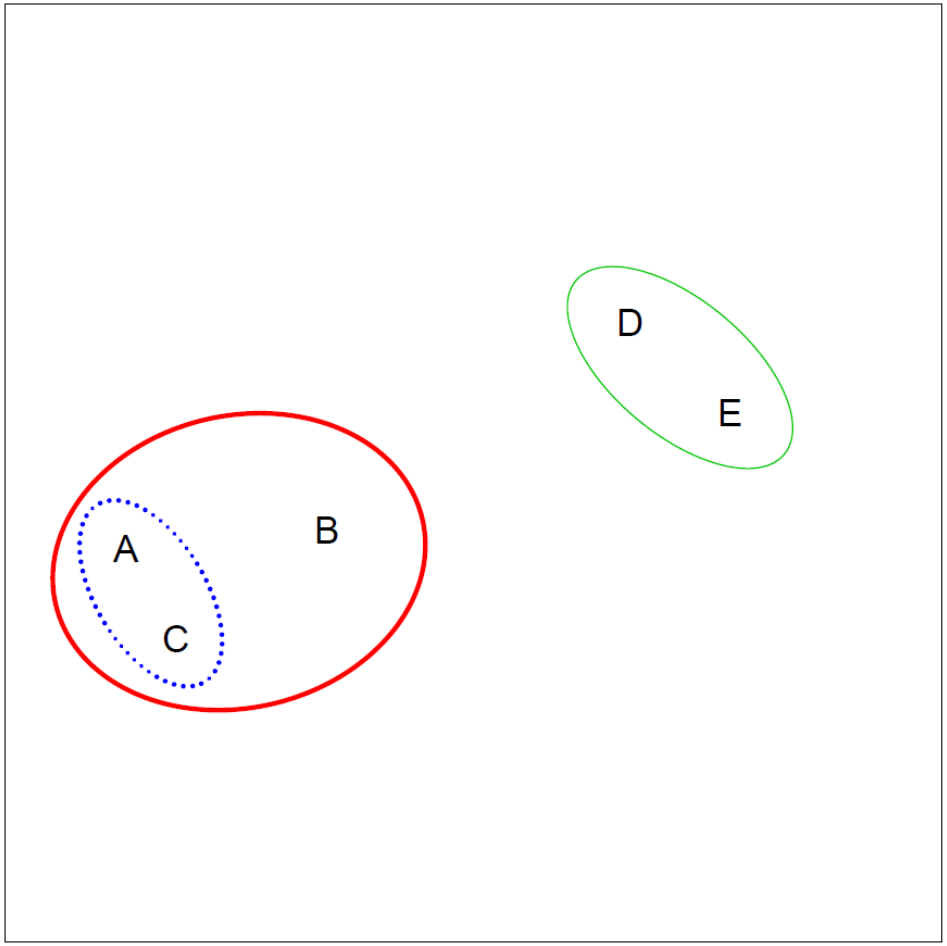
Hierarchical clustering (agglomerative)



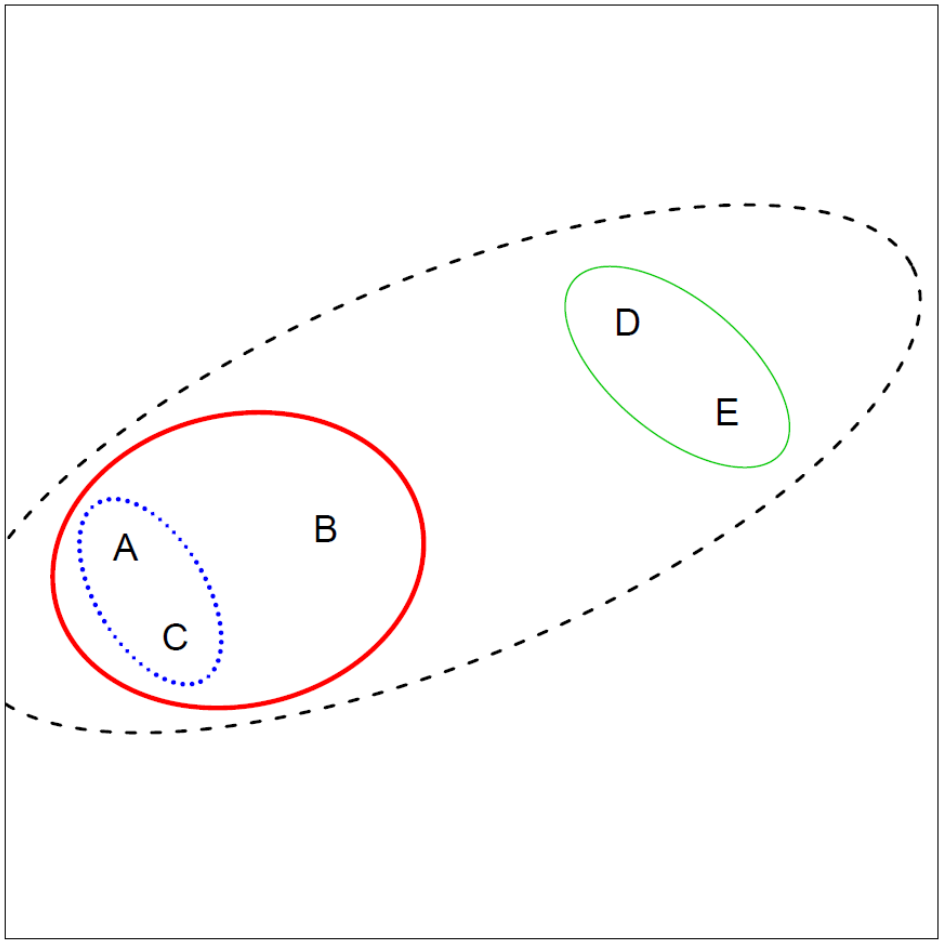
Hierarchical clustering (agglomerative)



Hierarchical clustering (agglomerative)

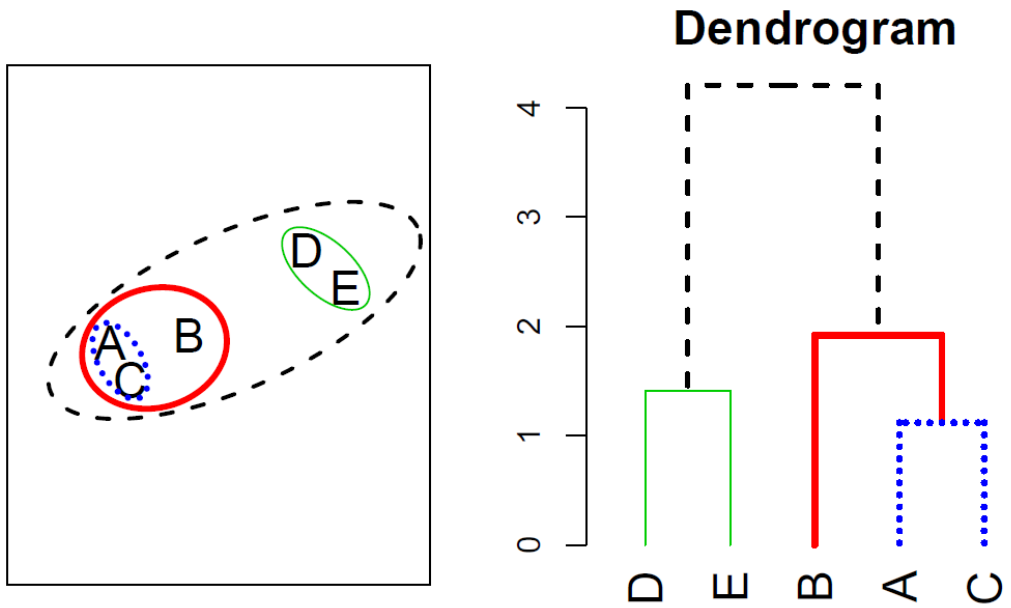


Hierarchical clustering (agglomerative)



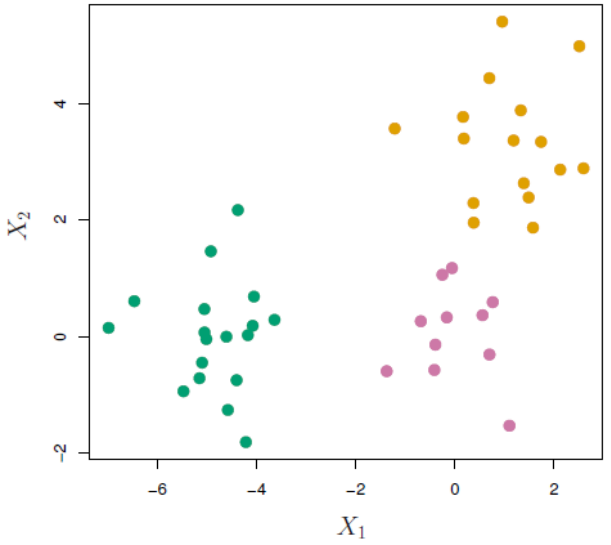
Hierarchical clustering (agglomerative)

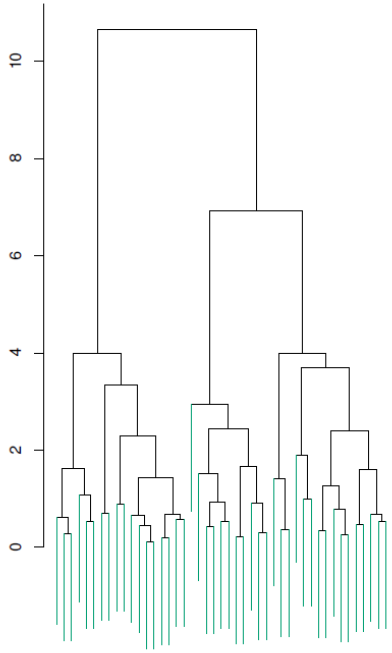
- Start with each point in its own cluster.
- Identify the **closest** two clusters and merge them.
- Repeat.
- Ends when all points are in a single cluster.

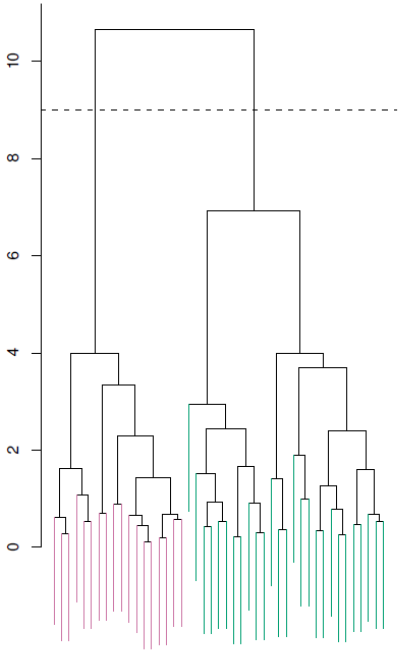


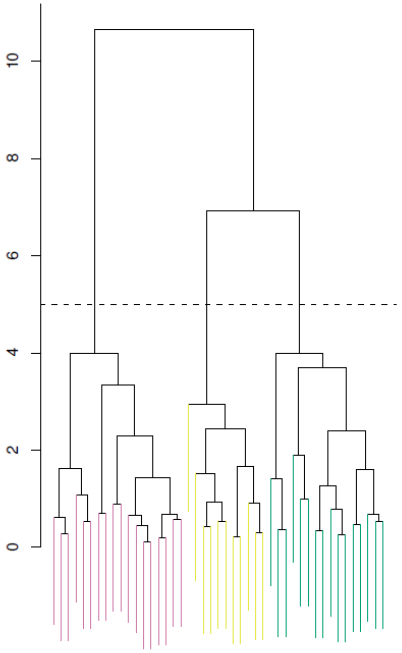
Hierarchical clustering

- Dendrogram



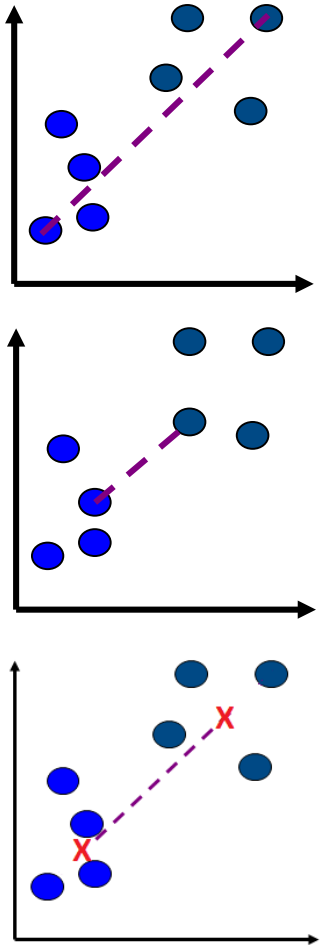






Hierarchical clustering

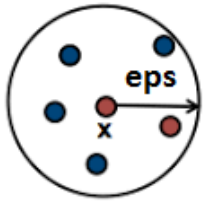
Linkage	*Description*
Complete	Maximal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities.
Average	Mean inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.



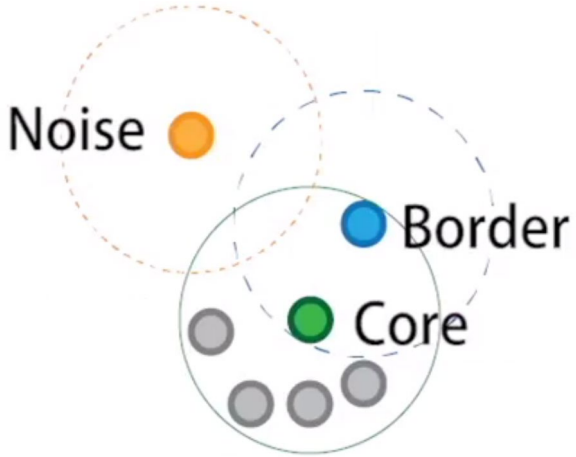
DBSCAN clustering method (quick overview)

DBSCAN (a density-based algorithm)

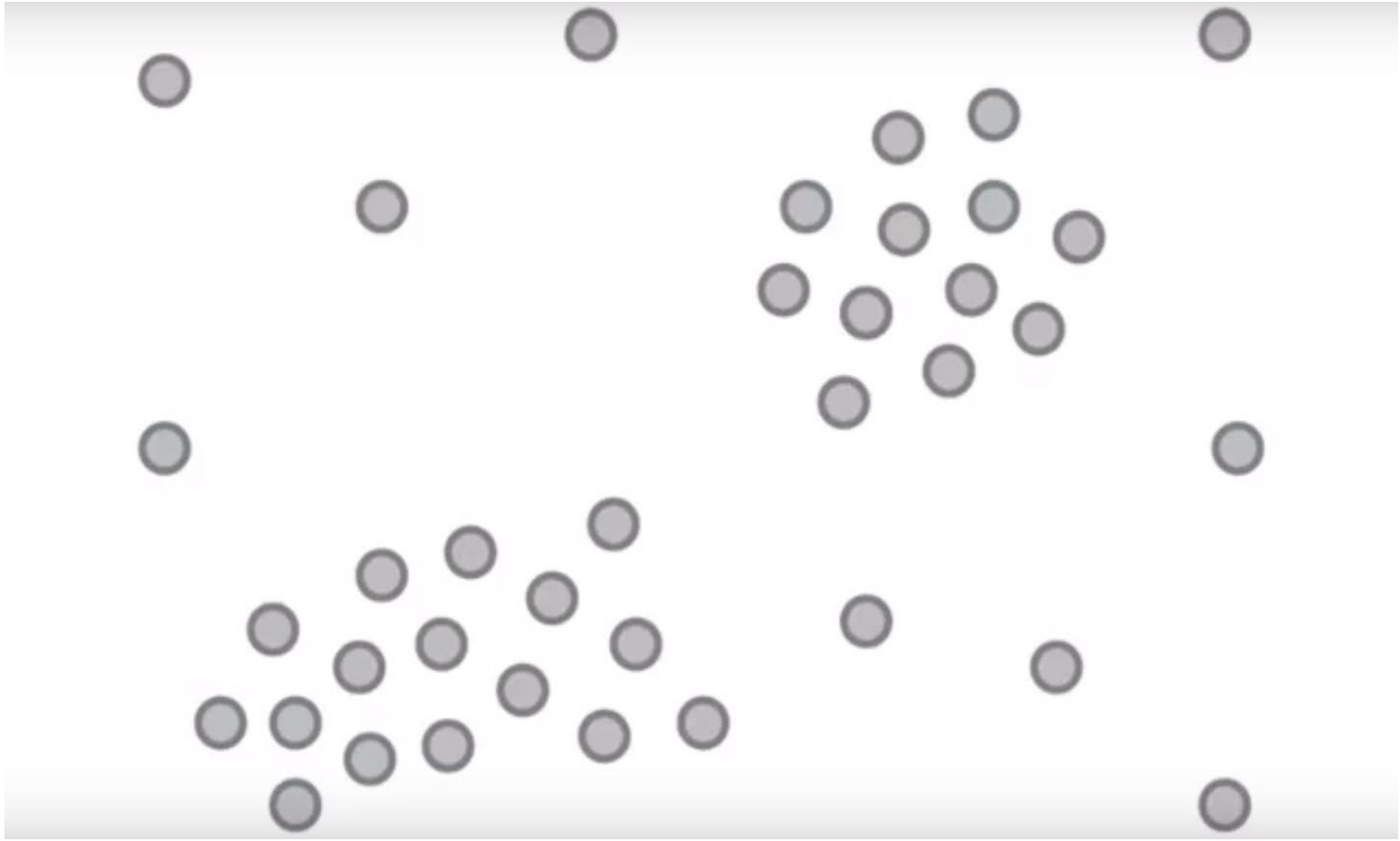
- A cluster is defined as a set of density-connected points.
- Two parameters:
  - $\boldsymbol{eps}$: defines the neighborhood
    - e.g. N(x) = $\{z \in X : dist(x, z) < \boldsymbol{eps}\}$
  - $\boldsymbol{minpts}$: minimum number of points the neighborhood.



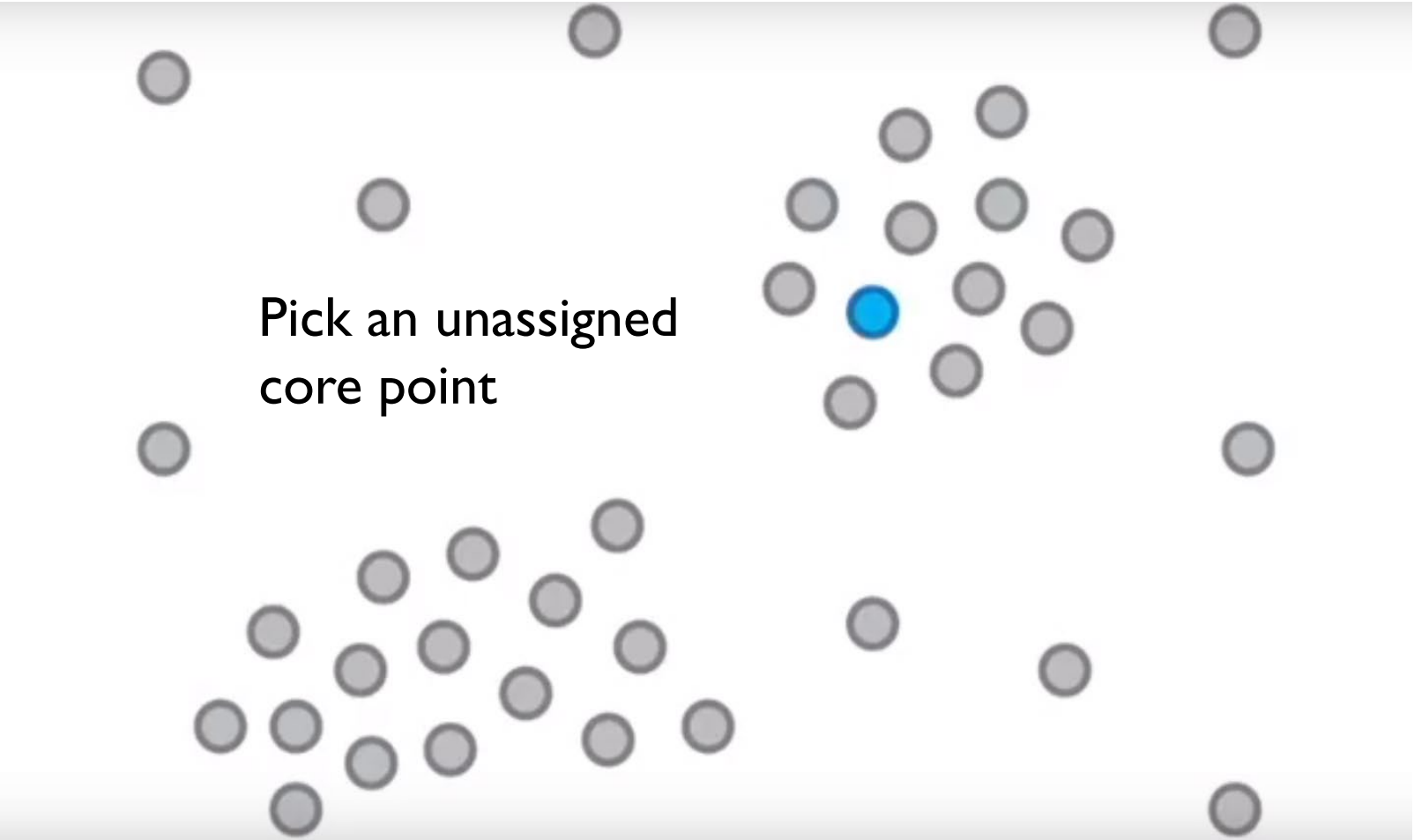
- Core points: Point x is a core point if $|\boldsymbol{N(x)}| > \boldsymbol{minpts}$
- Border points: Not a core point but has at least one core point in its neighborhood.
- Noise points: Not a core not a border point.



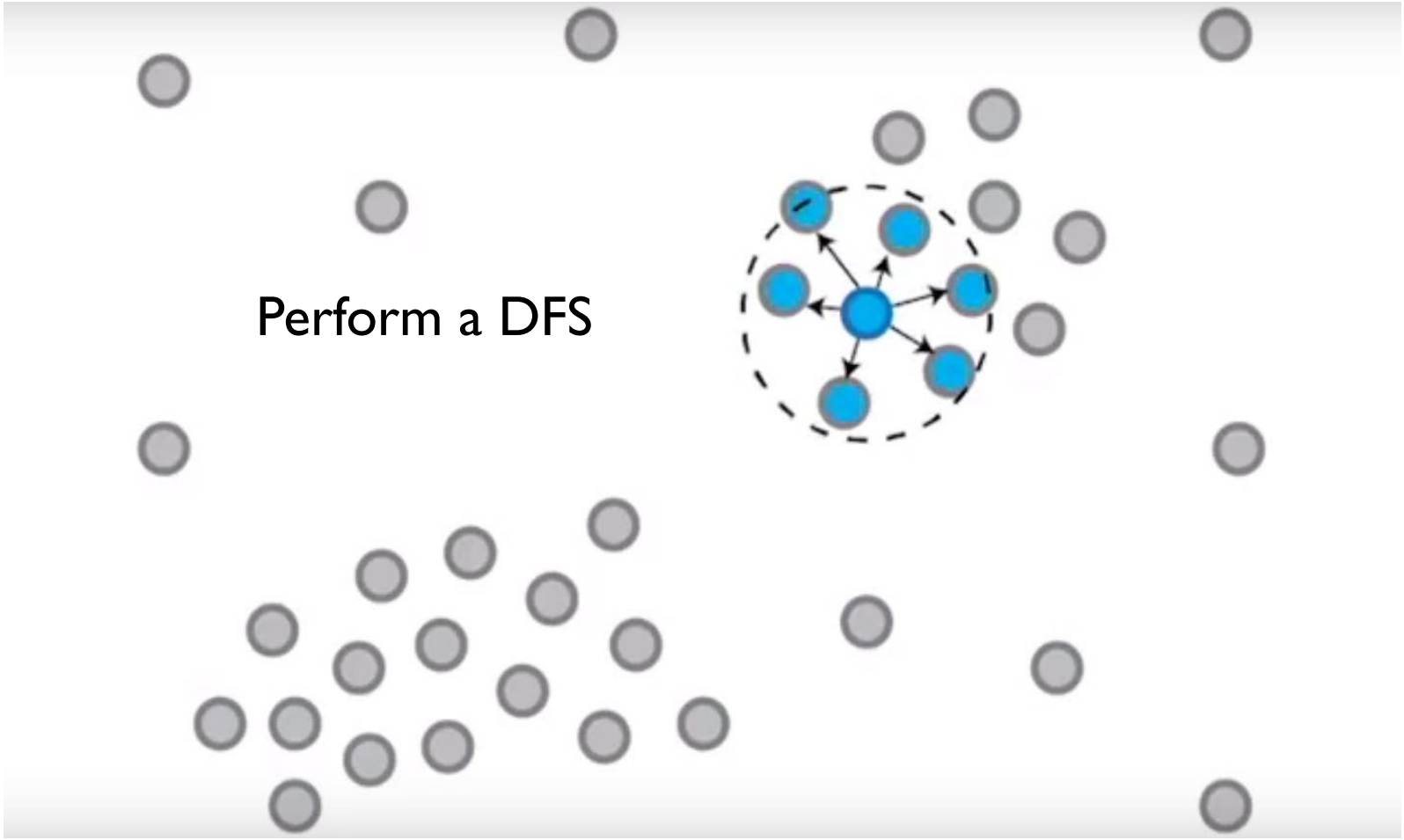
DBSCAN



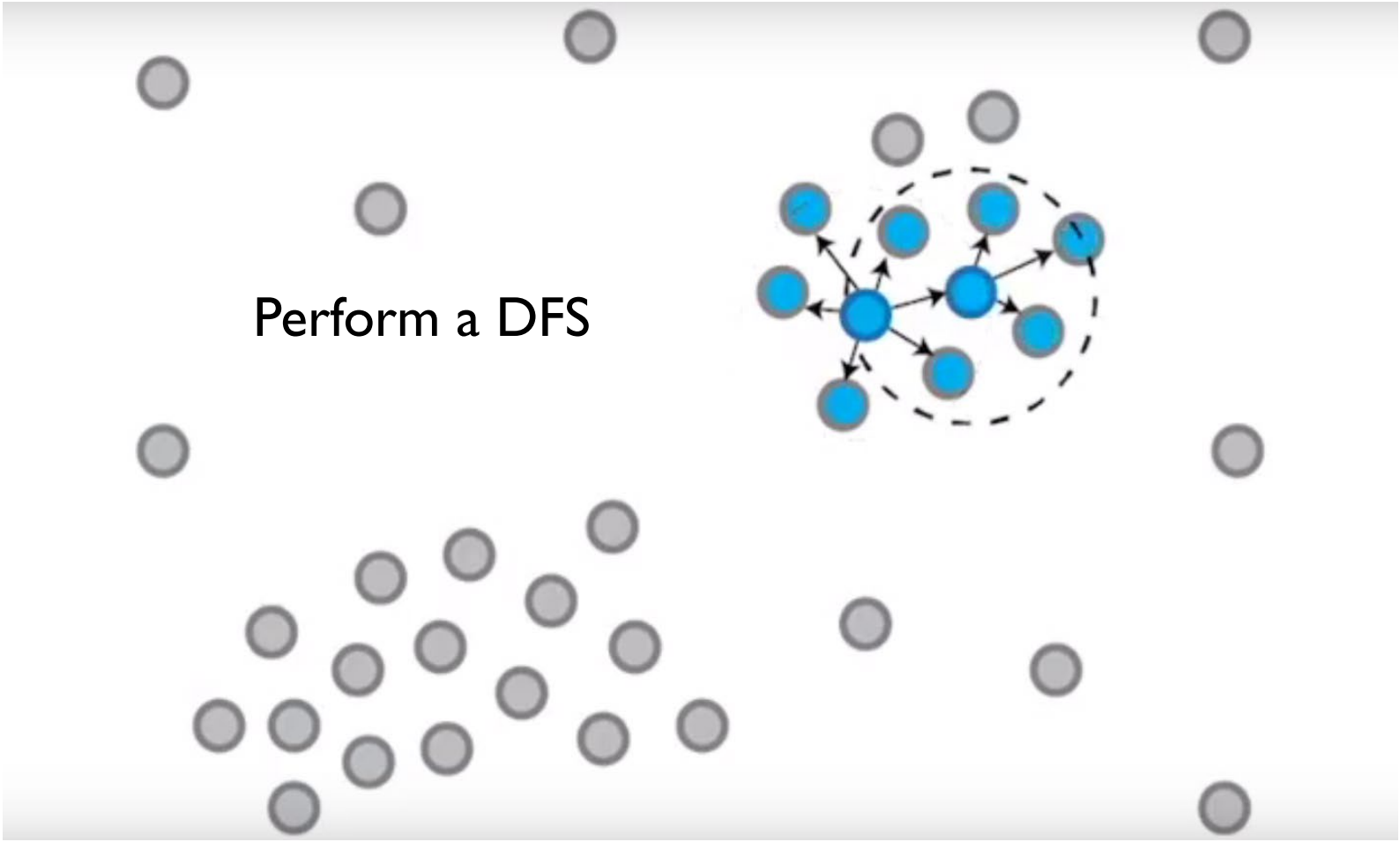
DBSCAN



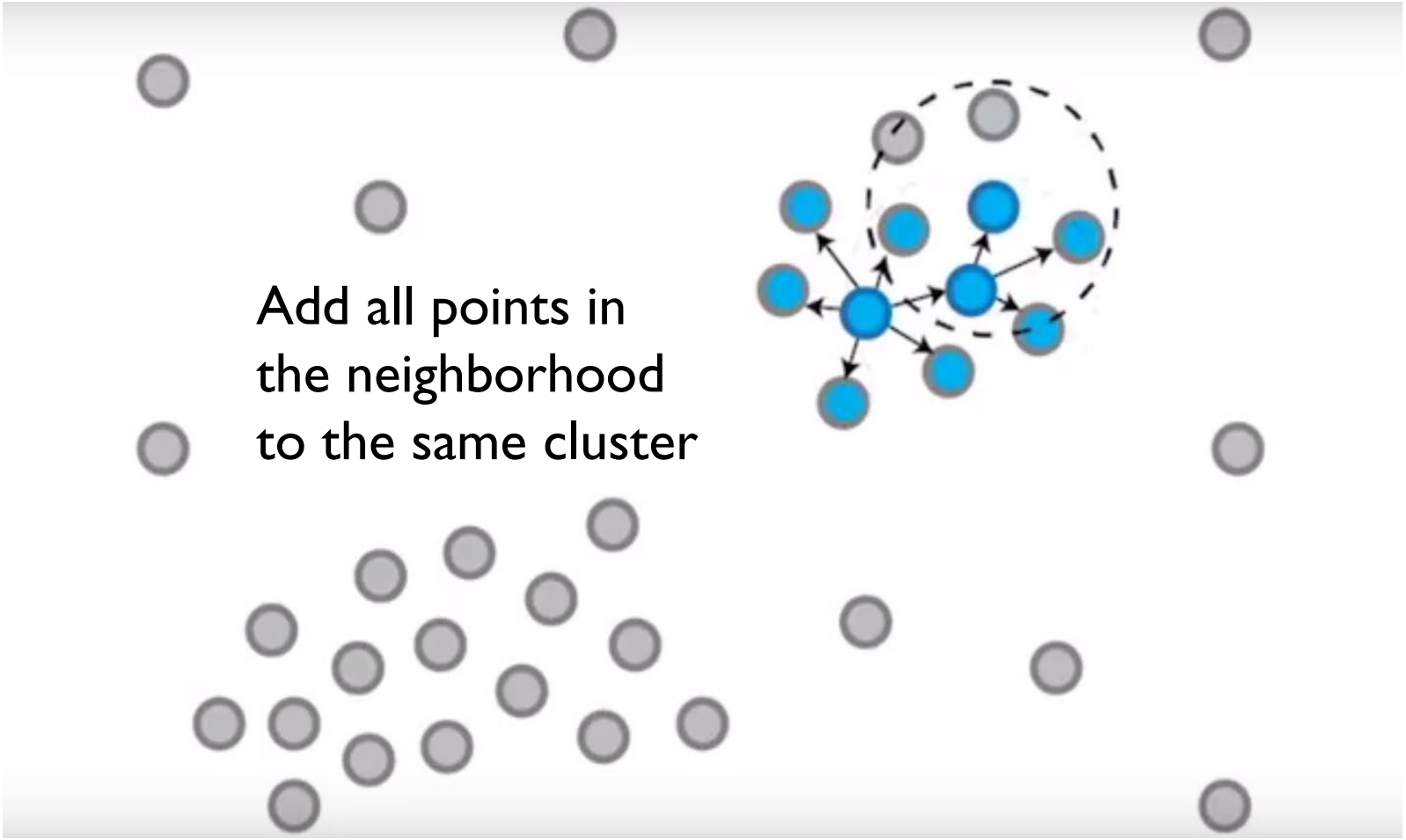
DBSCAN



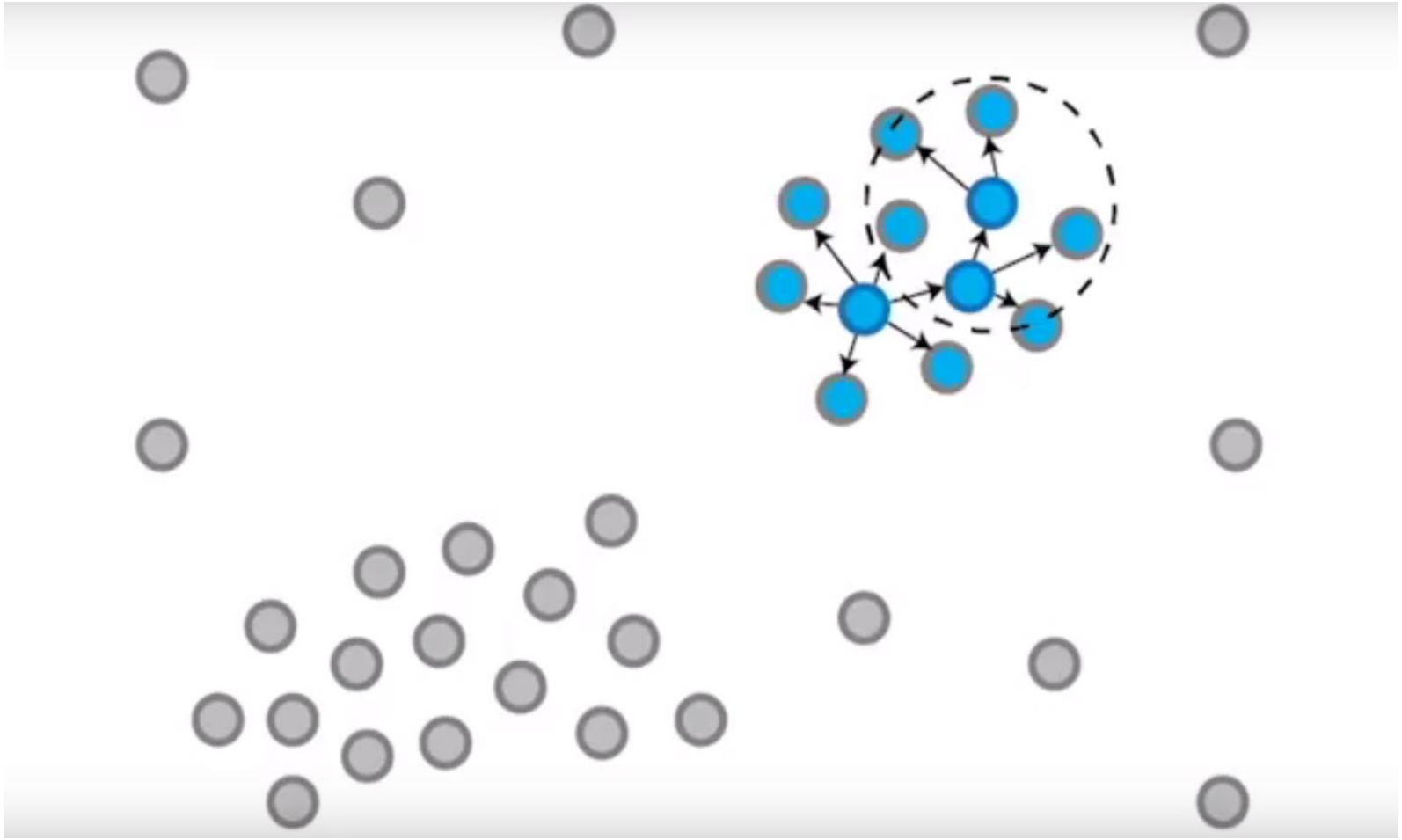
DBSCAN



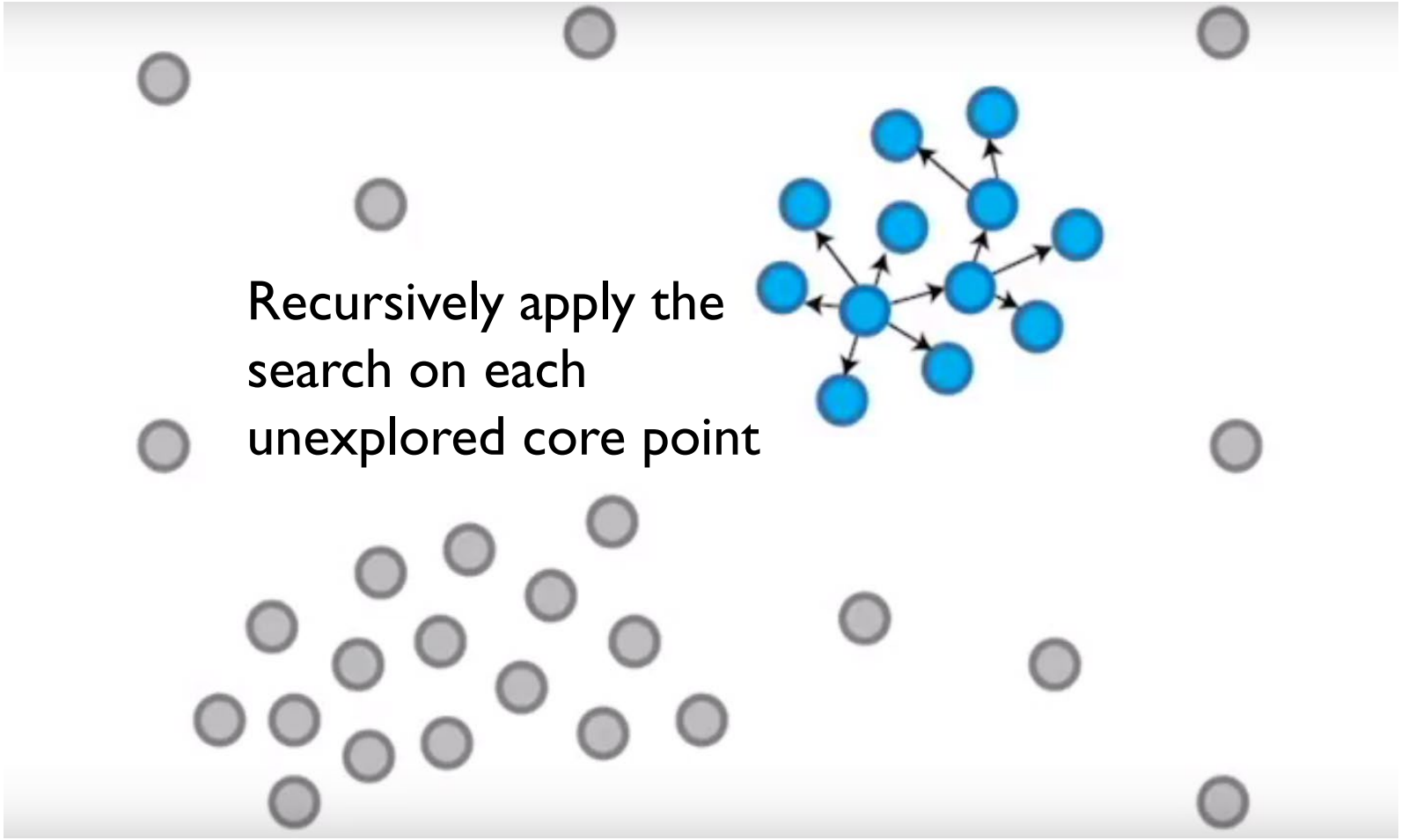
DBSCAN



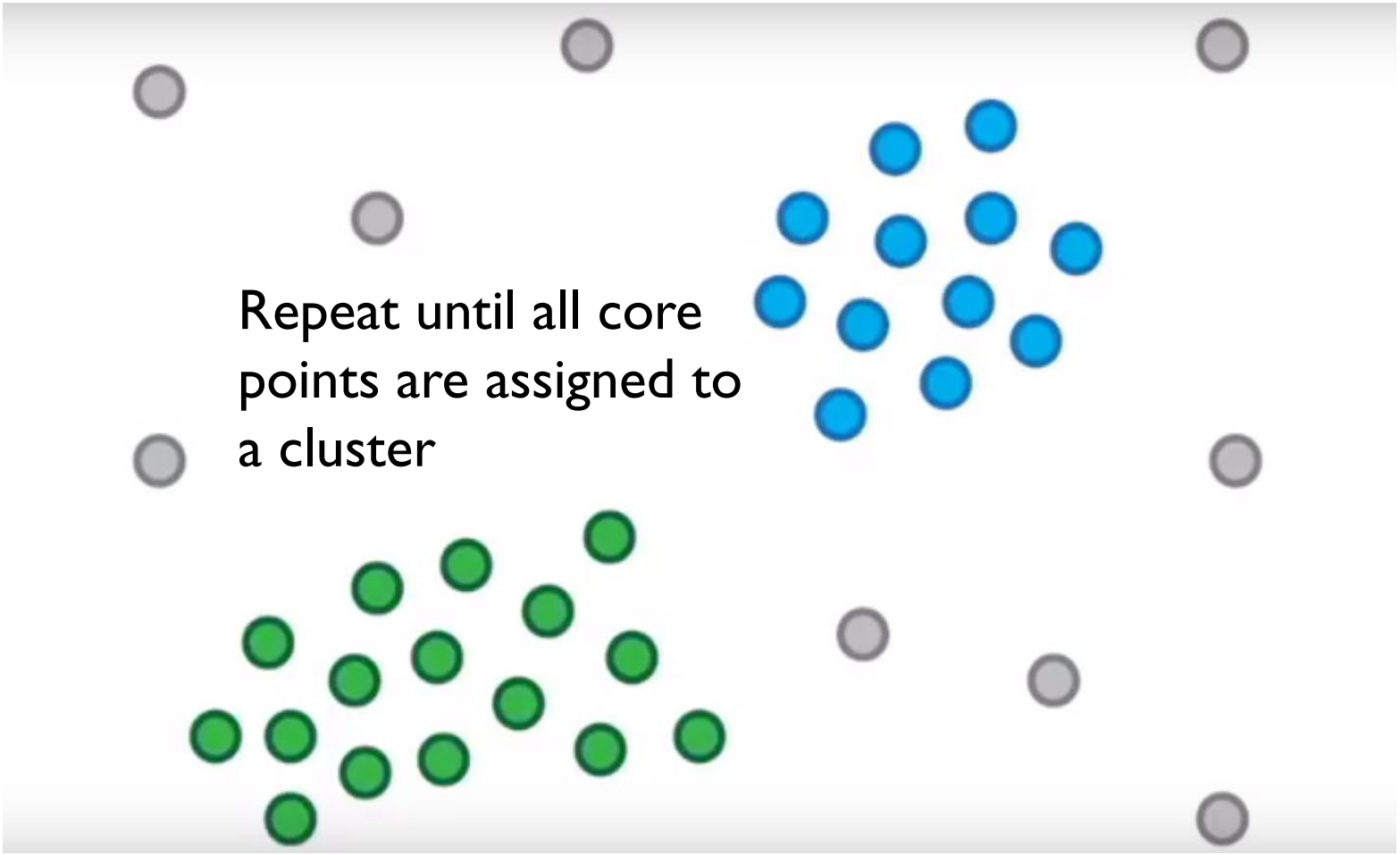
DBSCAN



DBSCAN


Recursively apply the
search on each
unexplored core point

DBSCAN



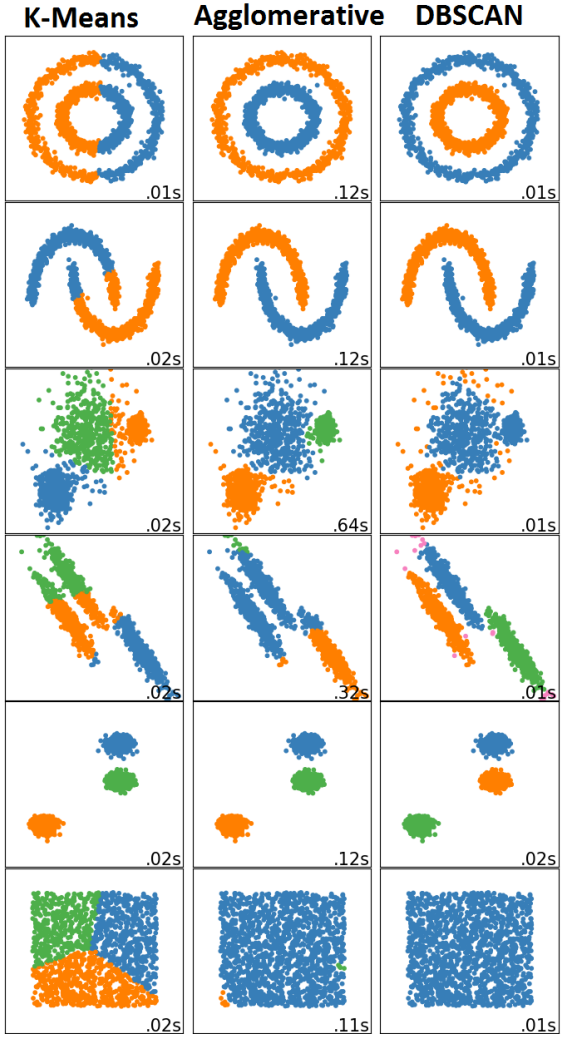
DBSCAN

- Strengths
  - Can discover any number of clusters
  - Can find clusters of varying size and shape
  - Can detect and ignore outliers in the data

change

- Weaknesses
  - Very sensitive the choice of parameters (e.g. neighborhood parameter)

Comparison



- Choosing the "best" algorithm is a challenge.
  - Every algorithm has limitations and works well with certain types of data.
  - One also needs to choose a suitable distance function and to select other parameter values.
  - Clustering is to certain extent subjective (personal preferences).

- We usually try different algorithms and analyze / explore the results …

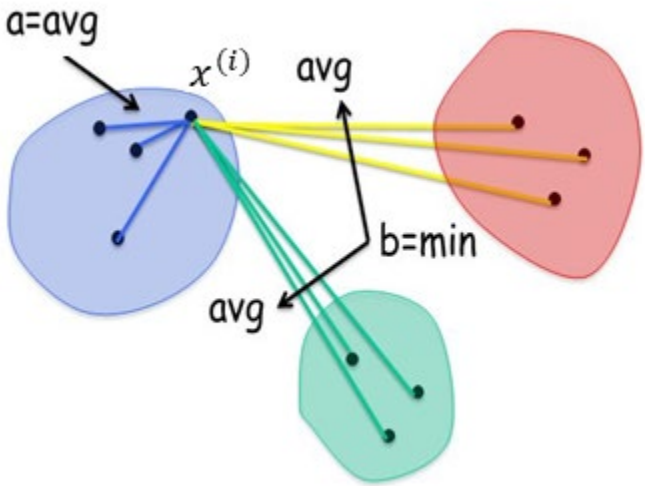
Evaluating the clustering results

Evaluation of the clustering quality

- No labels ➔ we do not know the "correct" clusters.
- Usually we use a measure that takes into account:
  - **Intra-cluster** cohesion (compactness):
    - Cohesion measures how close are the points within the same cluster.
  - **Inter-cluster** separation (isolation):
    - Separation means that different clusters should be far away from one another.

Silhouette coefficient:

- Let a_i be the average distance from point $x^{(i)}$ to the points in the same cluster as $x^{(i)}$ (i.e. blue cluster in the figure).
- Let b_i be the average distance from point $x^{(i)}$ to the points in the closest cluster to $x^{(i)}$ (e.g. say the green cluster is closer to $x^{(i)}$ than the red one).



- The silhouette coefficient s_i of $x^{(i)}$ is then

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

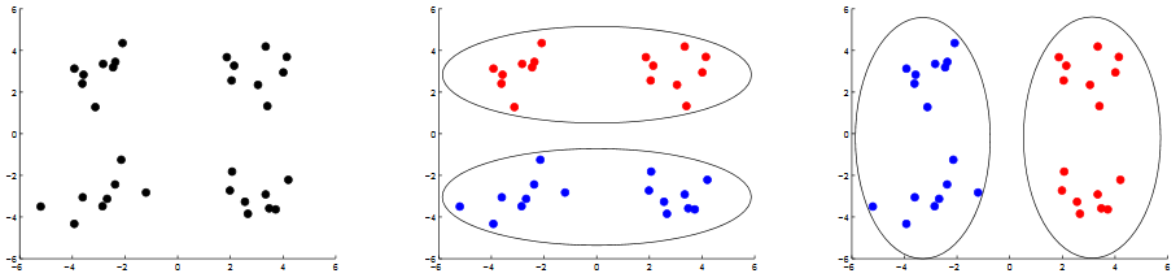
Final silhouette coefficient :

$$\mathrm{s} = \frac{1}{n}\sum_{i=1}^{n} s_i$$

Some problems with clustering

Problems with clustering (subjective)

- Multiple (different) clustering solutions, for the same data

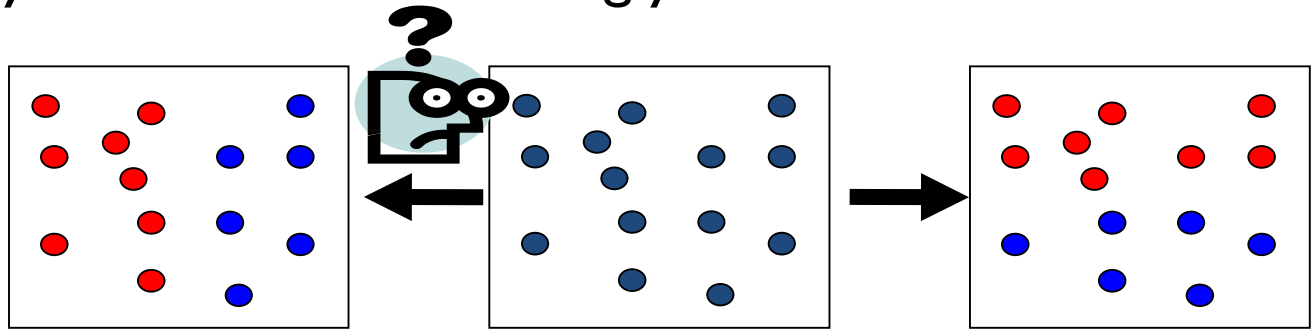


UserID	Country	Age	Gender	Blood	Heartbeat	Weight	Height	Sports	Income	Profession
1	China	young	male	healthy					poor	
2										
3										
4									rich	
5		old	female	unhealthy						
6										
7										
8	US	young	male							
9										
10										
11										
12		old	female	healthy					poor	
13										
14										
15										

People can be clustered by different criteria

Problems with clustering (towards semi-supervised)

- How do you tell which clustering you want?



- Constrained clustering techniques: