## Section 14 Unsupervised Learning- Clustering

- Unsupervised learning
  - Clustering
    - **K means.**
    - **Hierarchical clustering**
    - Density based
    - Spectral clustering
    - Distribution methods

## ➢ Supervised Learning

**Data:** $(\vec{x}, y)$. $\vec{x}$ is input, $y$ is output/response (label)

**Goal:** Learn a function to map $h$: $\vec{x} \rightarrow y$

**Examples:**

- Classification,
- Regression,
- Object detection,
- Semantic segmentation.

Cat

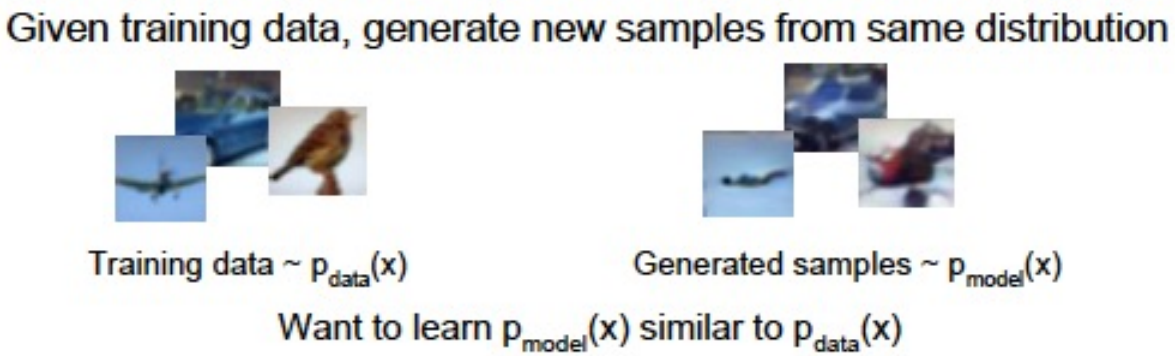
## ➢ Unsupervised Learning

**Data:** $\vec{x}$
Just input data, no output labels

**Goal:** Learn some **underlying hidden structure** of the data. Often used as part of exploratory data analysis.

**Examples:**

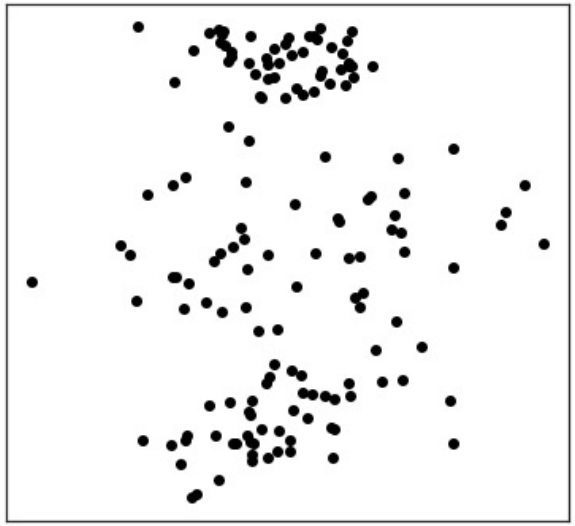
- **Clustering**, e.g., Identify the particular subgroups of stocks with close Relations. Accurate advertising based on the customers age, profession, reading, shopping habits.
- **Dimensionality reduction** (PCA, manifold learning),
- **Feature learning,**
- Generating samples, etc.

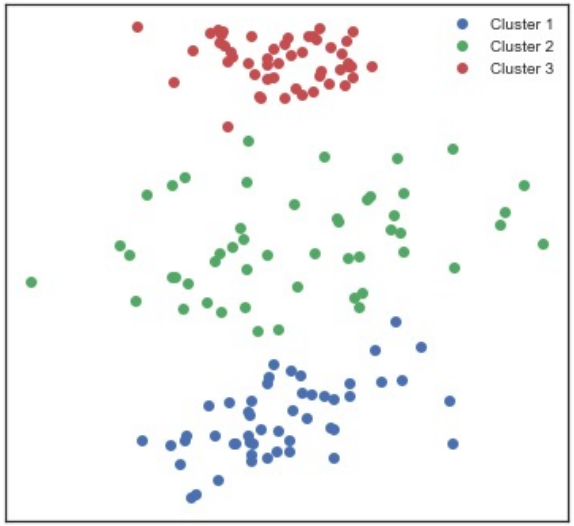


Both clustering and PCA seek to simplify the data via a small number of summaries, but their mechanisms are different:

1. **PCA** looks to find a low-dimensional representation of the observations that explain a good fraction of the variance;

2. **Clustering** looks to find homogeneous subgroups among the observations.






A clustering algorithm that emphasizes similarly labeling close by points will not be as aware of the total distance in feature space, while a method that emphasizes not having far away points will ignore local information.

## Example: Market segmentation

You are the owner of a shop. It doesn't matter if you own an e-commerce or a supermarket. It doesn't matter if it is a small shop or a huge company such as Amazon or Netflix, it's better to know your customers.

| Customer ID | Gender | Age | Annual Income | Spending Score |
|---|---|---|---|---|
| 1 | Male | 19 | 15 | 39 |
| 2 | Male | 21 | 15 | 81 |
| 3 | Female | 20 | 16 | 6 |
| 4 | Female | 23 | 16 | 77 |
| 5 | Female | 31 | 17 | 40 |

You were able to collect basic data about your customers holding a membership card such as Customer ID, age, gender, annual income, and spending score. This last one is a score based on customer behavior and purchasing data. There are some new products on the market that you are interested in selling. But you want to target a specific type of clients for each one of the products.

The **goal** is to perform market segmentation by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.

The task of performing market segmentation amounts to **clustering** the people in the data set.

https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python

## ➢ Cluster Analysis

Techniques for finding **subgroups** or data points, or **clusters**, in a data set, so that the observations within each group are quite similar to each other.

Clustering is one of the most widely used techniques in exploratory data analysis. Clustering is the task of organizing data into groups so that “similar” objects end up in the same cluster and “different" objects end up in different clusters.

The basic problem for clustering is a problem for all unsupervised learning: the problem of ground truth (to discover structure).

Find distinct clusters on the basis of a data set. Since there are no labels from which to learn what possible clusters might look like, we have to make assumptions about how the data are grouped.

## ➢ Type of clustering:

Two best-known clustering approaches: K-means clustering and hierarchical clustering.

1. **K-means clustering:** seek to partition the observations into a pre-specified number of clusters. K-means is an example of centroid based clustering is governed by overall distance to closest centroid.

2. **Hierarchical clustering:** a tree-like visual representation of the observations, called a **dendrogram**, that allows us to view at once the clustering obtained for each possible number of clusters, from 1 to n.

**Cluster observations** on the basis of the features in order to identify subgroups among the observations;

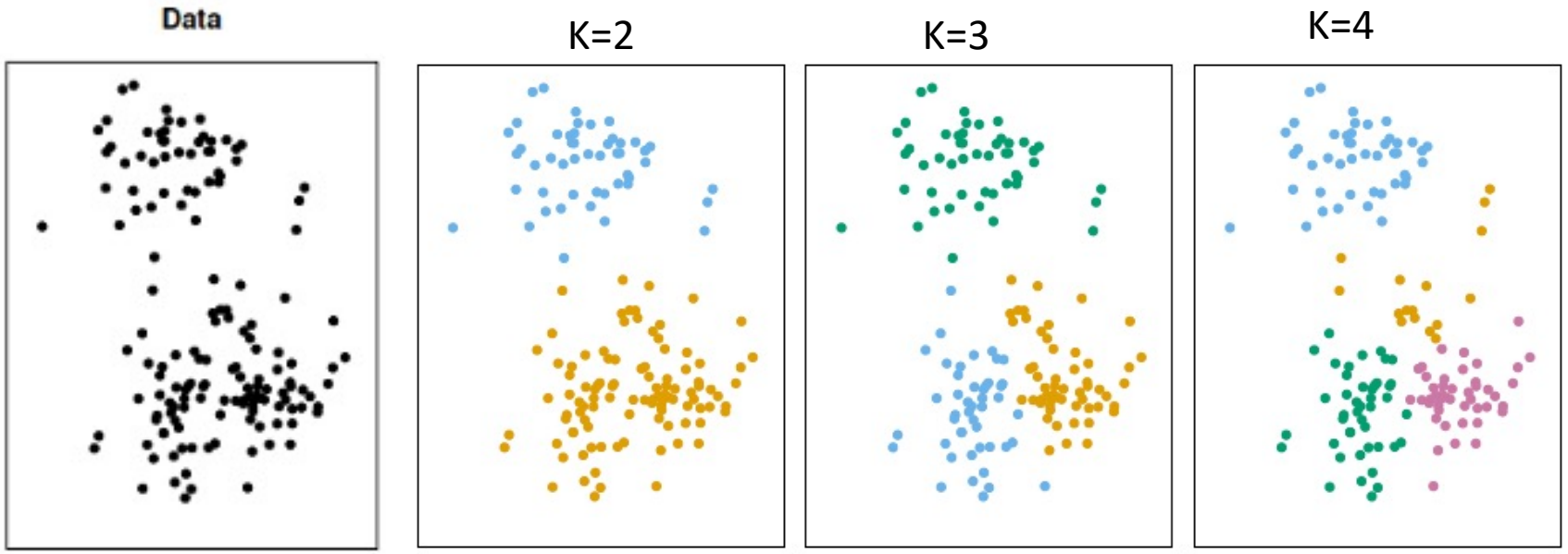
Or **cluster features** on the basis of the observations in order to discover subgroups

Several other clustering are also used in practice, e.g., density based method, spectral clustering, distribution method, .

## ❖ K-Means clustering

Partition the data set of $n$ observations into $K$ distinct, non-overlapping subsets,

denoted as $C_k$, for $k = 1, \ldots, K$, is called a **cluster**.




A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.

**Good clustering**: the within-cluster variation is as small as possible

Let $W(C_k)$ be a measure of the **within-cluster variation** for cluster $C_k$.

There are several different ways to define $W(C_k)$.

For example, using **squared Euclidean distance**, we define

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

We wish to **minimize the total within-cluster variations**

$$\text{minmize}_{C_1,\ldots,C_K} \left\{ \sum_{i=1}^{K} W(C_K) \right\}$$

The problem is computationally difficult ("non-deterministic polynomial acceptable problems" NP-hard).

## ➢ K-Means cluster algorithm

1. Randomly assign a number, from 1 to $K$, to each of the observations. These serve as **initial cluster assignments** for the observations.

2. Iterate until the cluster assignments stop changing:

   1) For each of the $\boldsymbol{K}$ **clusters**, compute the **cluster centroid** $\boldsymbol{\mu_k}$. The $k$-th cluster centroid $\mu_k$ is the vector of the $\boldsymbol{p}$ **feature means** for the observations in the $k$-th cluster.

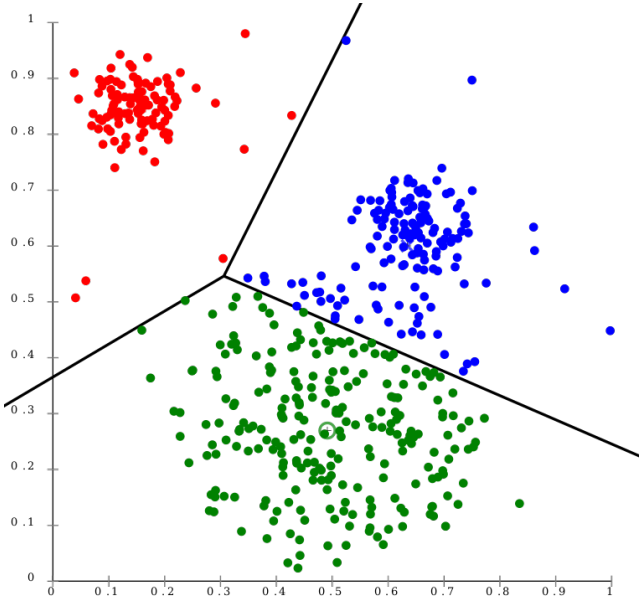
   2) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

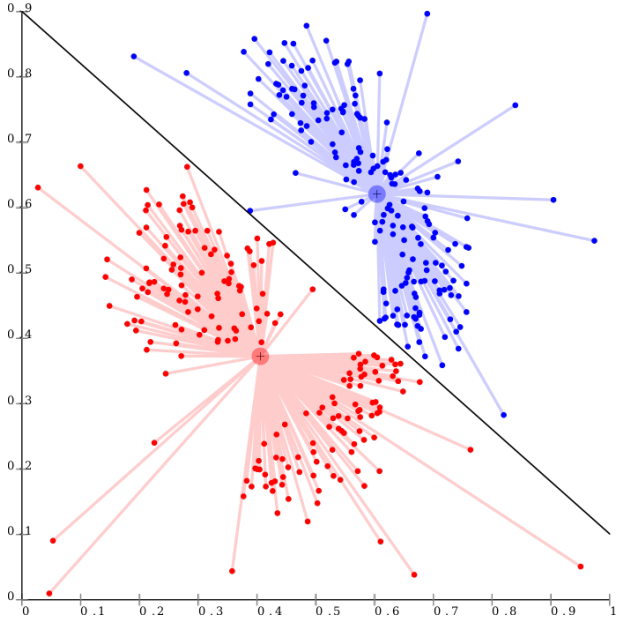
- For every $j$, compute **new cluster centers**:

$$\mu_j := \frac{\sum_{i=1}^{m} \mathbb{I}\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} \mathbb{I}\{c^{(i)} = j\}}$$

- For every $i$, **assign** each point to cluster,

$$c^{(i)} := \arg\min_{j} ||x^{(i)} - \mu_j||$$





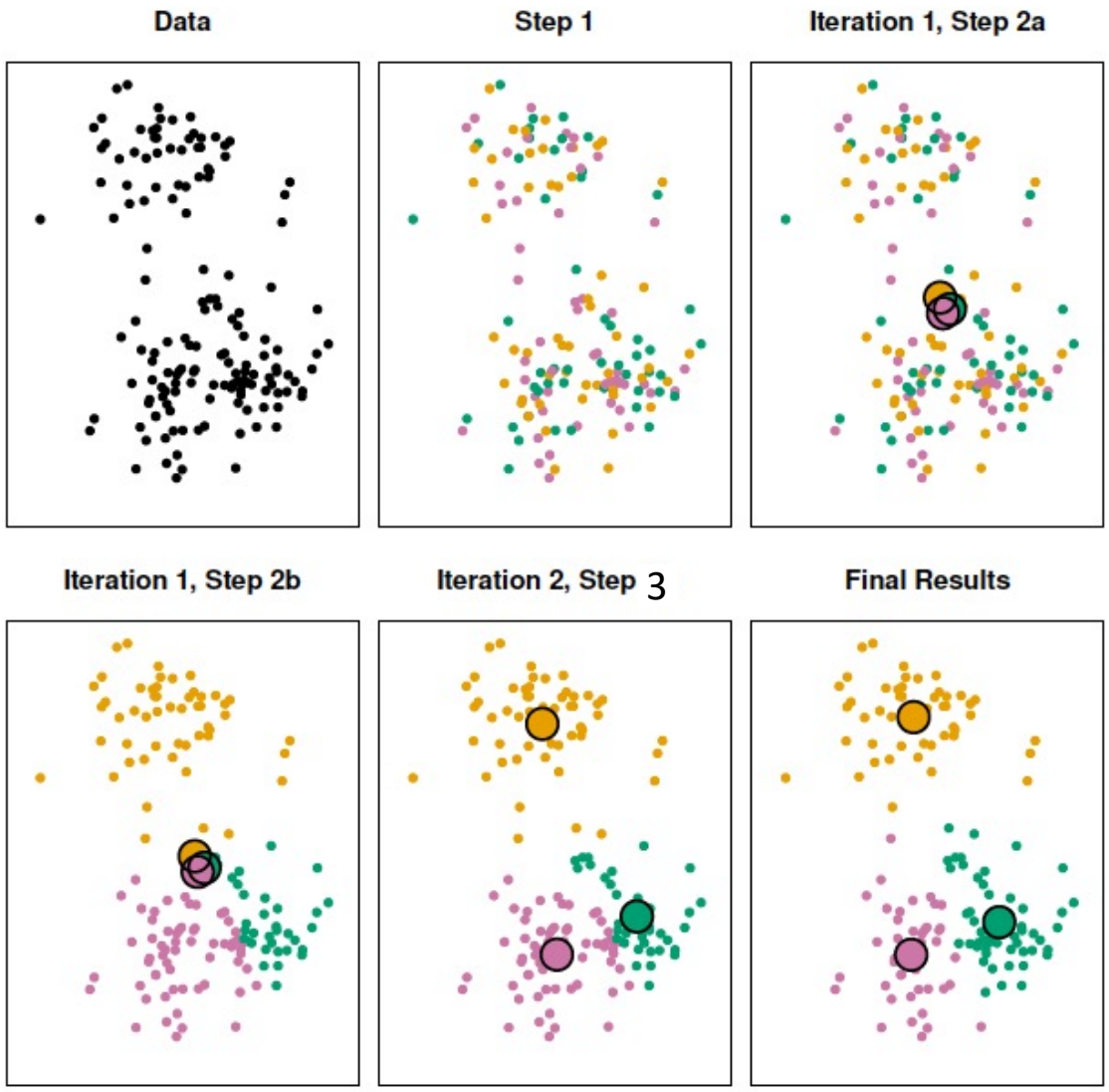
- The **objective function** always **decreases** at each step.

$$\frac{1}{|C_k|}\sum_{i,j\in C_k}||\vec{x}^{(i)}-\vec{x}^{(j)}||^2 = 2\sum_{i\in C_k}||\vec{x}^{(i)}-\bar{x}||^2$$

where $\bar{x}$ is the sample mean $x^{(i)}$ for $i \in C_k$

- K-means algorithm finds a **local minimum**.
- The result depends on the **initial** (random) cluster assignment.
- Try several different initials, and select the best result (the smallest objective function).

An example of the K-means algorithm with K = 3.



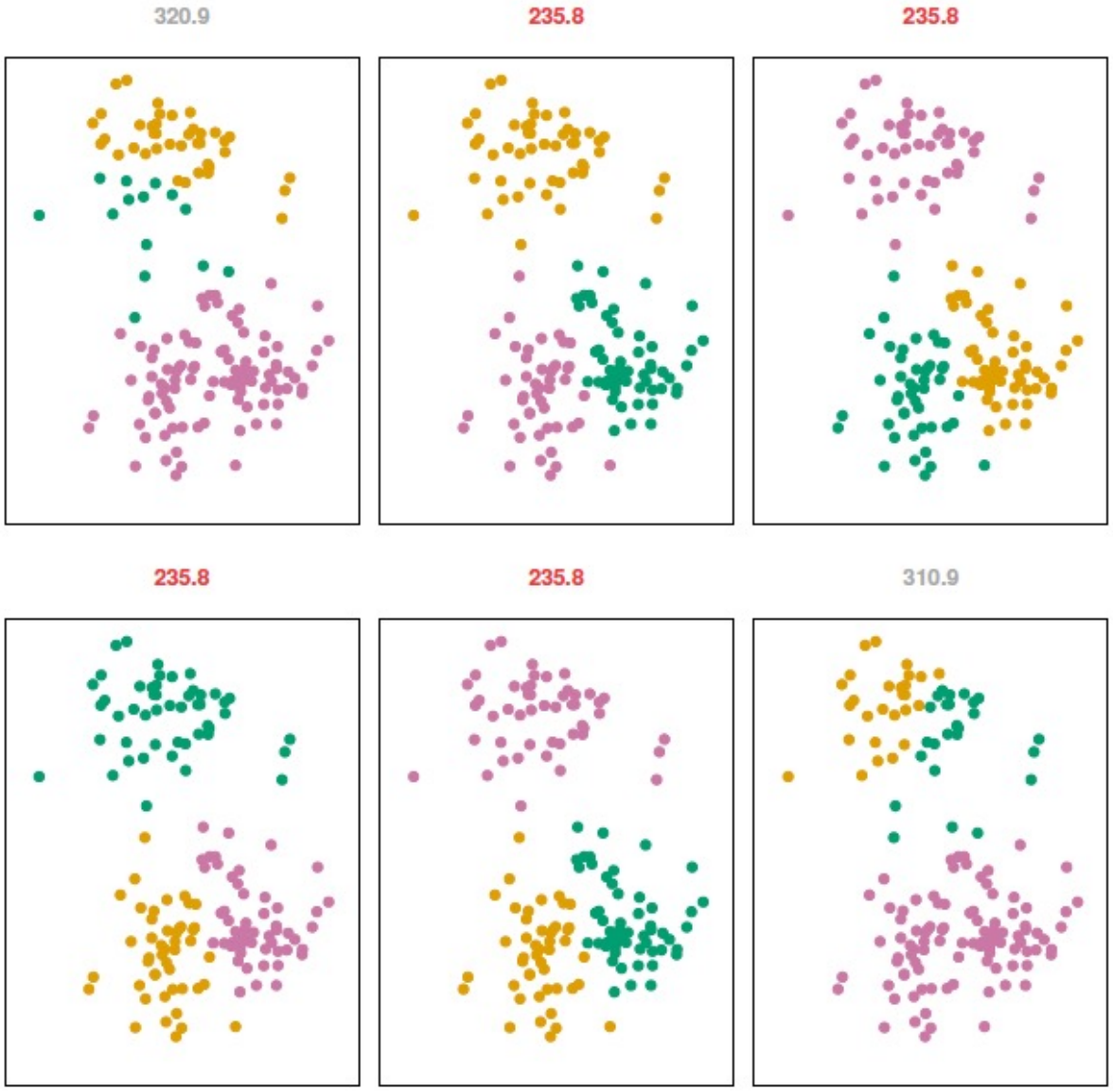

Step 1, each observation is randomly assigned to a cluster.

Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.

Step 2(b), each observation is assigned to the nearest centroid.

Step 3 is once again performed, leading to new cluster centroids.

The final results obtained after ten iterations.



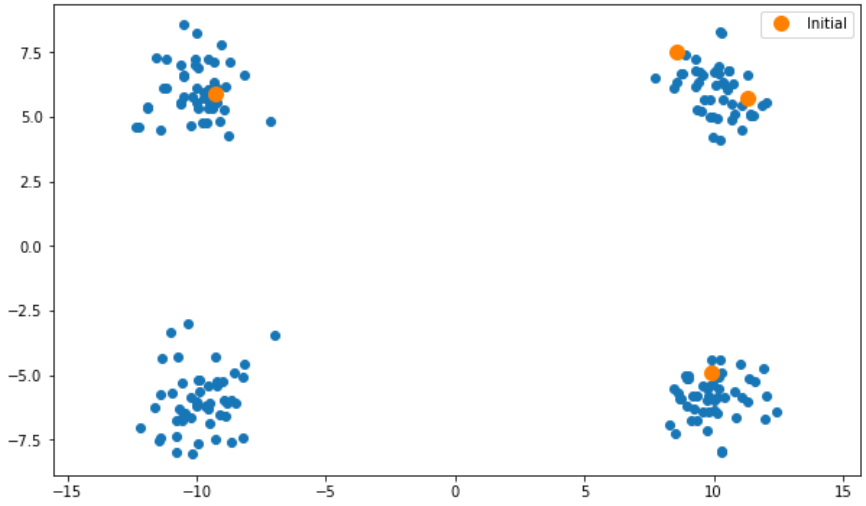

**K-means clustering** performed **six times** on the previous data with K = 3, each time with a **different** random assignment of the observations in Step 1 of the K-means algorithm.

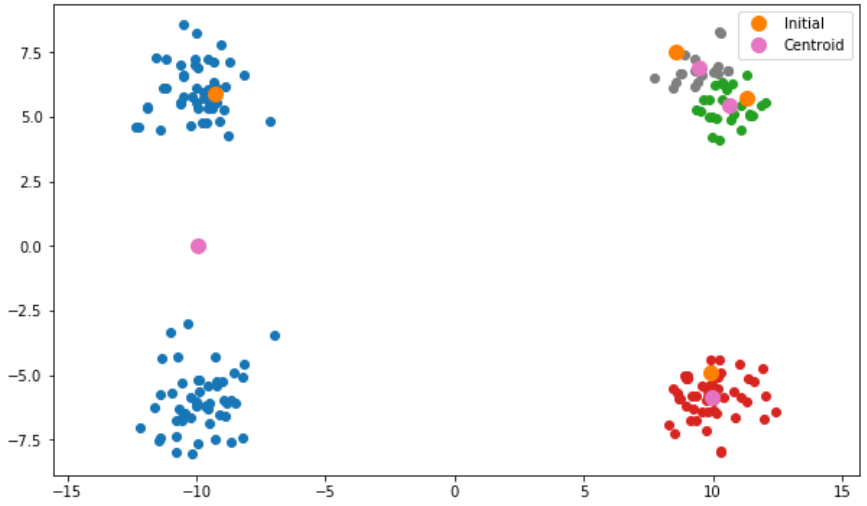
Above each plot is the value of the objective. **Three different** local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters.

Those labeled in red all achieved the same best solution, with an objective value of 235.8.

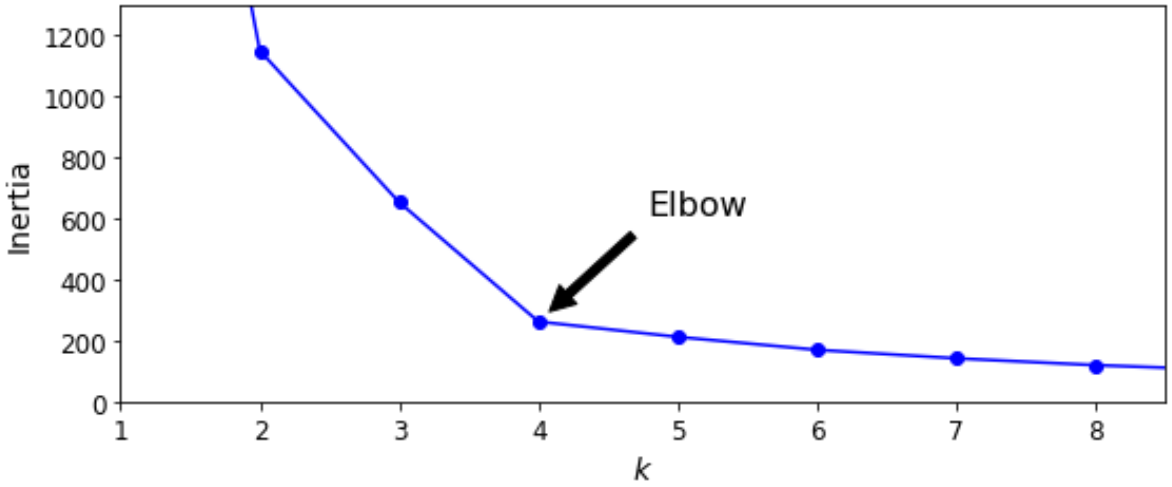
## Practical Issues: Initialization

Initialization is probably the greatest difficulty in most greedy algorithms.



## Choosing K




The choice of K is the other main practical issue in K-means clustering. A common criteria is to compute the clustering for many K, plot them, and set K to be just larger than the steepest “elbow”.

Assume we initialize by setting $\mu_j$ to be randomly selected data points $x_j$ .

If the data has K clusters, each of size N/K, then the probability of randomly selecting a point from each cluster is

$$\frac{K-1}{K}\frac{K-2}{K}\cdots\frac{1}{K} = \frac{K!}{K^K} \leq \left(\frac{1}{2}\right)^{\frac{K}{2}}$$

So we will almost always have bad initial centroids for K large.

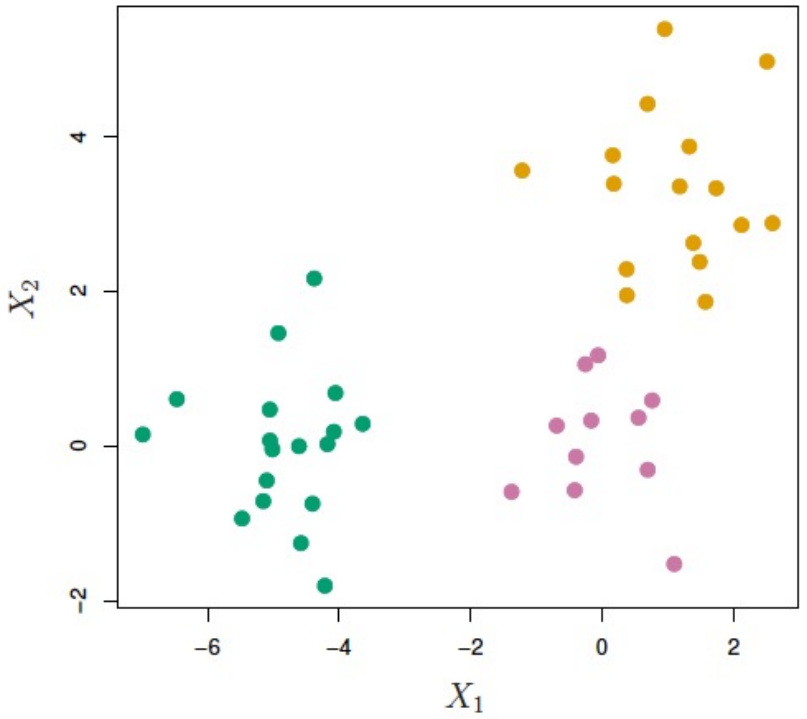
To get better results:

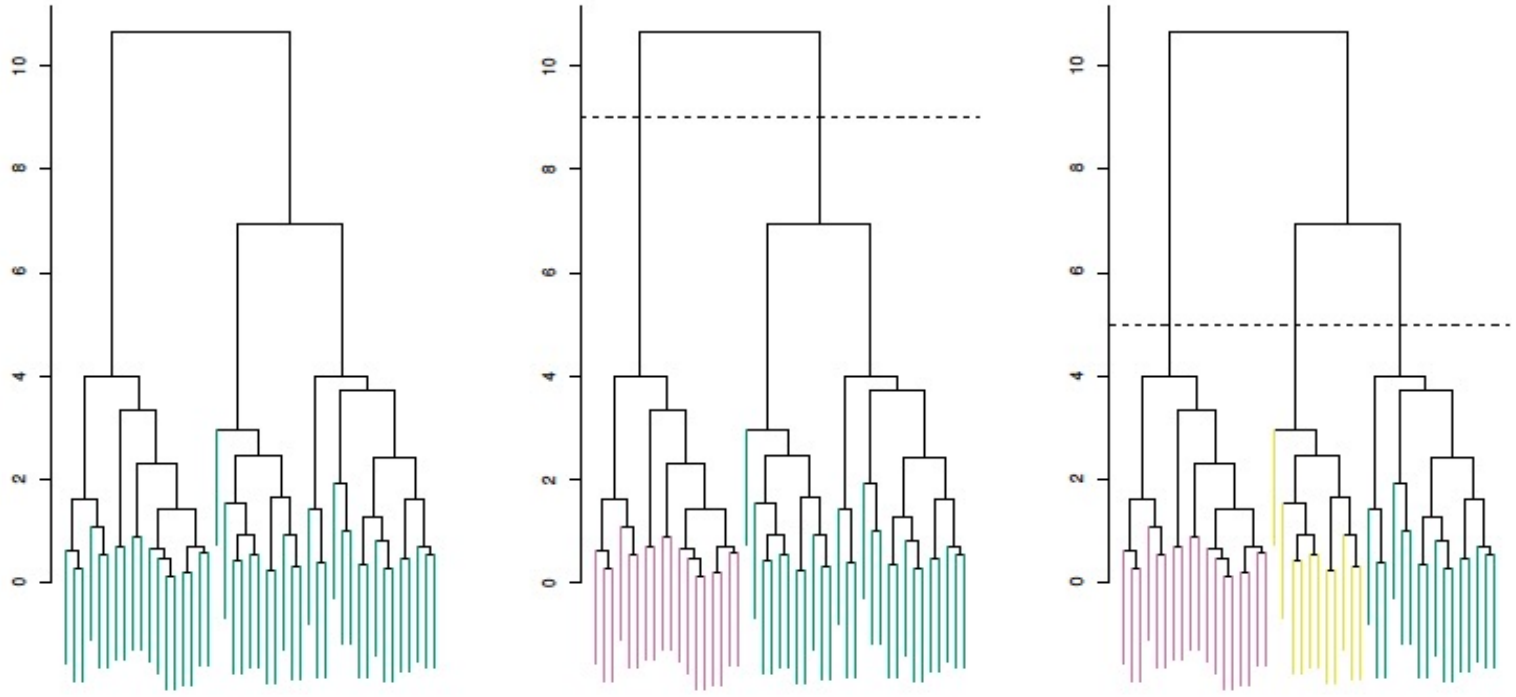
- Always take multiple randomized starts and consider ensembling them.
- Initialize with bottom up hierarchical methods.
- Alternatively, select more than $k$ points and only keep the most widely separated. Finally, we can use a different initialization scheme.

## ❖ Hierarchical clustering

- K-means clustering requires pre-specified number of clusters, a disadvantage.
- Hierarchical clustering does not require that.
- Hierarchical clustering results in a tree-based representation of the observations, called a **dendrogram**.
- Hierarchical clustering is bottom-up or agglomerative clustering.

Forty-five observations generated in two-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.






**Left**: dendrogram obtained from hierarchically clustering the data with **complete linkage and Euclidean distance.**

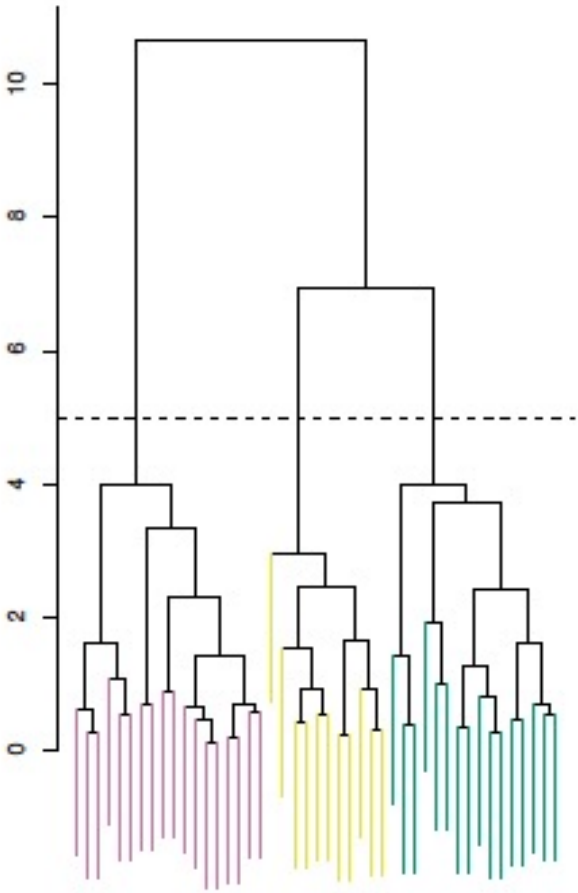
**Center**: the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.

**Right**: the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors.

Note: the colors were not used in clustering, but are simply used for display purposes in the figure.

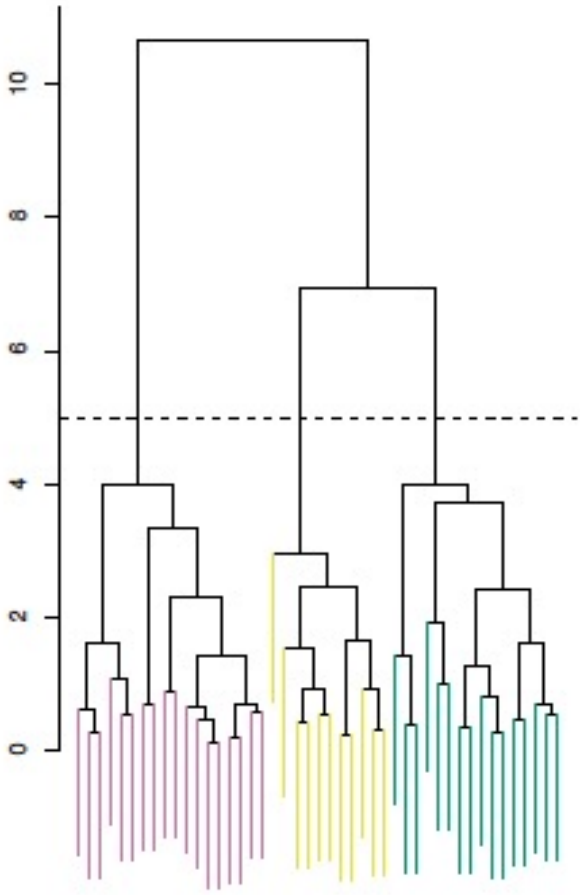
## Interpreting a dendrogram

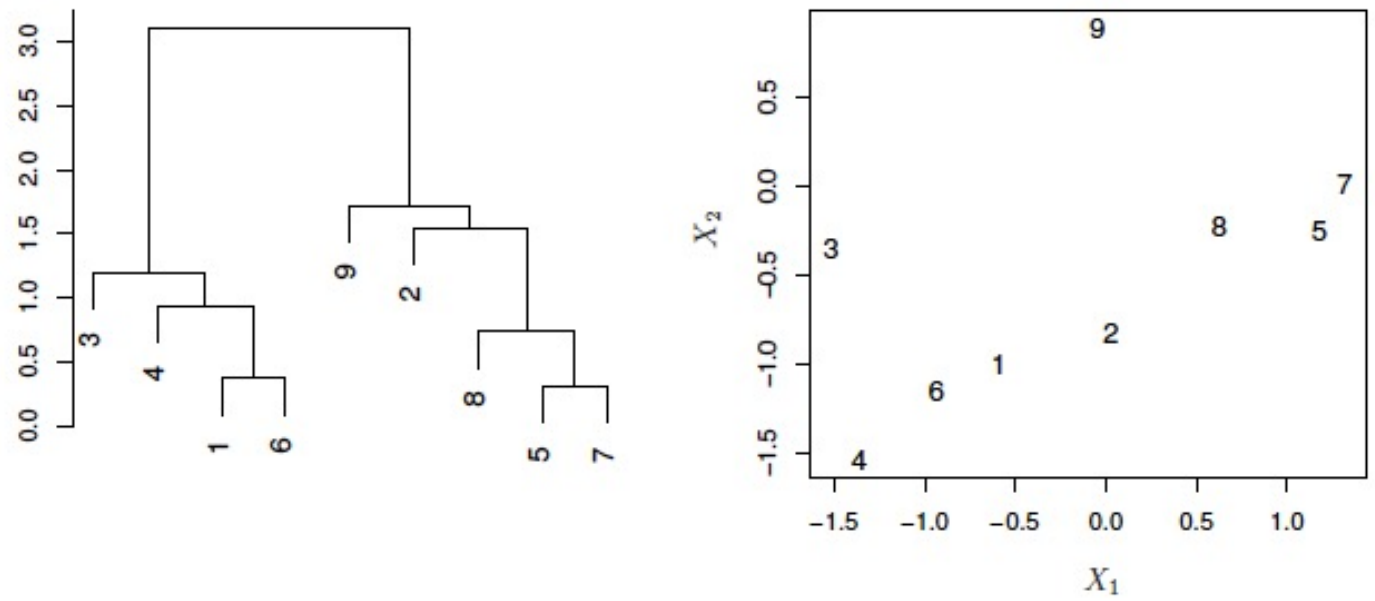
- Each **leaf** of the dendrogram represents one of the 45 observations.
- However, as we **move up** the tree, some leaves begin to **fuse into branches**. These correspond to observations that are **similar** to each other.
- As we move higher up the tree, branches themselves fuse, either with leaves or other branches.
- The earlier (lower in the tree) fusions occur, the more similar the groups of observations are to each other.
- Observations that fuse later (near the top of the tree) can be quite different.



## A rough closeness measure

- For any two observations, we can look for the point in the tree where **branches** containing those two observations are first fused. The **height** of this fusion, as measured on the **vertical axis**, indicates how different the two observations are.
- Observations that fuse at the very **bottom** of the tree are quite **similar** to each other, whereas observations that fuse close to the **top** of the tree will tend to be quite **different**.





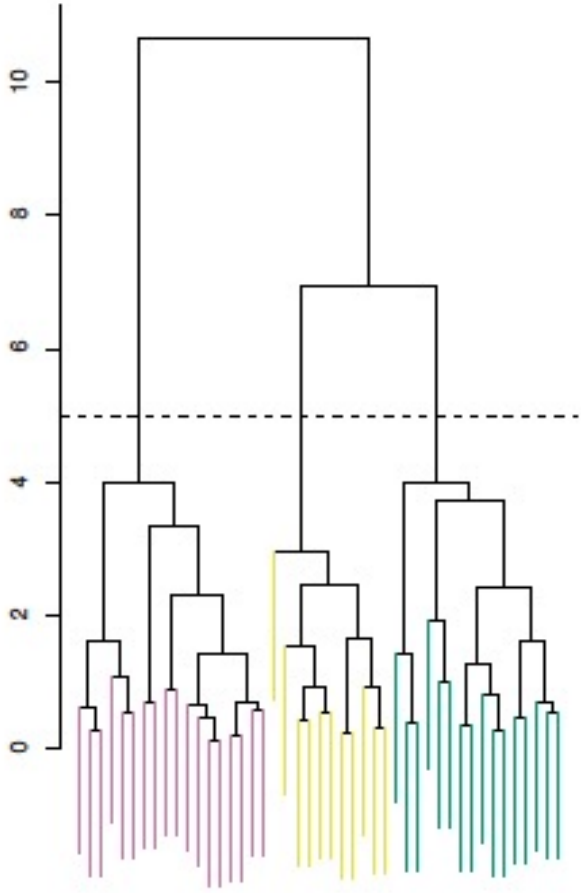

An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.

**Left**: a dendrogram generated using Euclidean distance and complete linkage. Observations 5 and 7 are quite similar to each other, as are observations 1 and 6. However, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7, even though observations 9 and 2 are close together in terms of horizontal distance. This is because observations 2, 8, 5, and 7 all fuse with observation 9 at the same height, approximately 1.8.

**Right**: the raw data used to generate the dendrogram can be used to confirm that indeed, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7.

## Identifying clusters

- Make a **horizontal cut** across the dendrogram.
- The distinct sets of observations beneath the cut can be interpreted as **clusters**.
- The **lower cuts** create **more clusters**.
  The higher cuts create less clusters.
- One single dendrogram can be used to obtain any number of clusters.
- Choice of cuts can even be done by **visual** judgment of the dendrogram.
- When hierarchical structure does **not exist** in data, the hierarchical clustering could be worse than K-means clustering.



## ➢ Hierarchical clustering algorithm

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \ldots, 2$

    1) Examine all pairwise **inter-cluster dissimilarities** among the $i$ clusters and identify the pair of clusters that are **least dissimilar** (that is, **most similar**). **Fuse** these two clusters. The dissimilarity between these two clusters indicates the **height** in the dendrogram at which the fusion should be placed.

    2) Compute the new pairwise **inter-cluster dissimilarities** among the $i-1$ remaining clusters.

## Linkage: the dissimilarity measure between two clusters

A summary of the four most commonly-used types of **linkage**

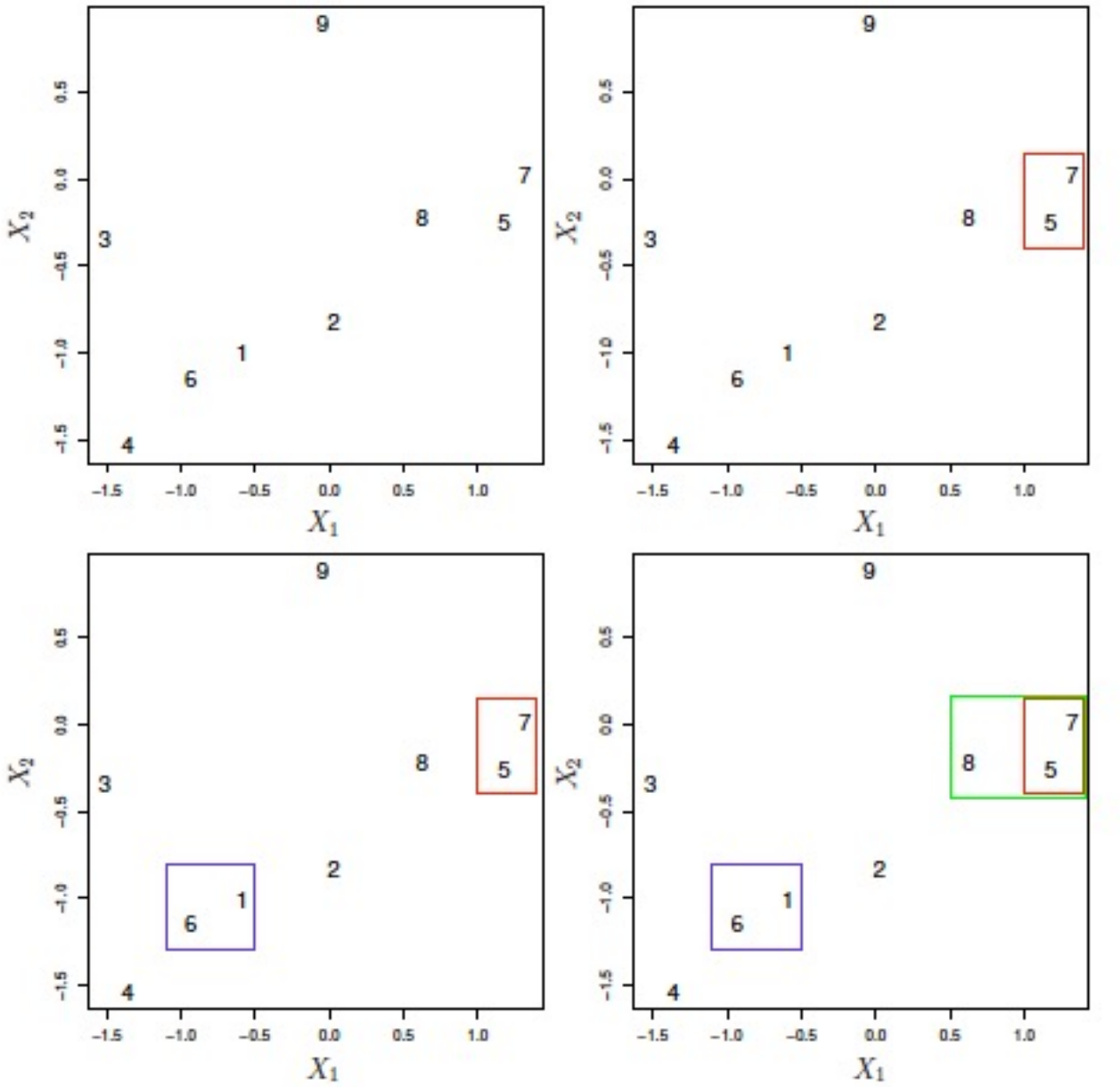
**1. Complete: Maximal inter-cluster dissimilarity**. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the **largest** of these **dissimilarities**.

**2. 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. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.

**3. 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.

**4. 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.

An illustration of the first few steps of the hierarchical clustering algorithm, using the data, with **complete linkage and Euclidean distance.**




**Top Left:** initially, there are nine distinct clusters, {1},{2},…,{9}.

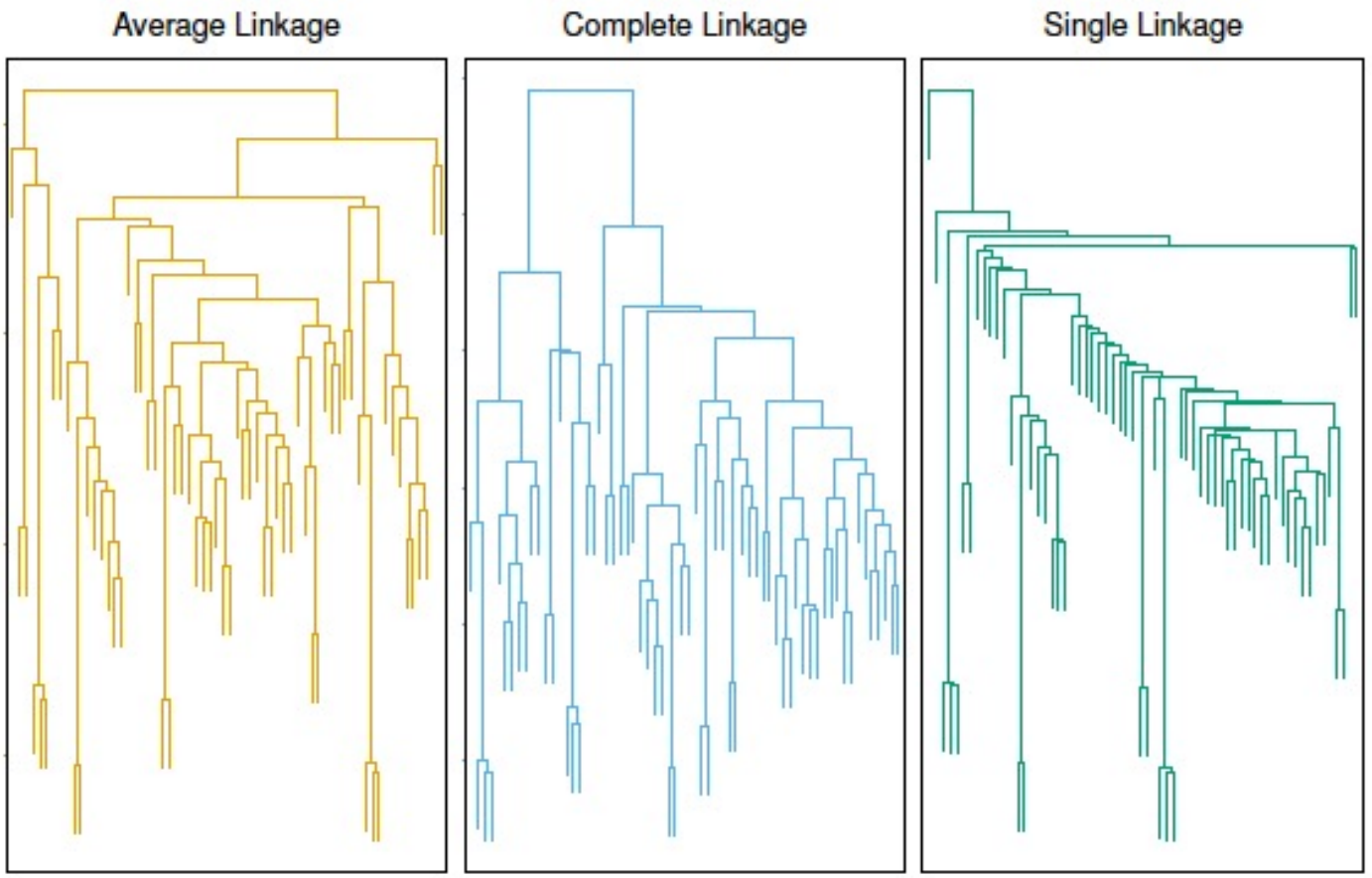
**Top Right:** the two clusters that are closest together, {5} and {7}, are fused into a single cluster.

**Bottom Left:** the two clusters that are closest together, {6} and {1}, are fused into a single cluster.

**Bottom Right:** the two clusters that are closest together using complete linkage, {8} and the cluster {5,7}, are fused into a single cluster.

**Average**, **complete**, and **single linkage** applied to an example data set.

Average and complete linkage tend to yield more balanced clusters.



## Dissimilarity measure

- Very important and can greatly affect the final result.
- Euclidean distance is a commonly used measure.
- Correlation based distance: if two observations have high correlation, the distance is closer. (Caution: this is not correlation between two variables, but between two observations.)
- Different problem may need different dissimilarity measure.

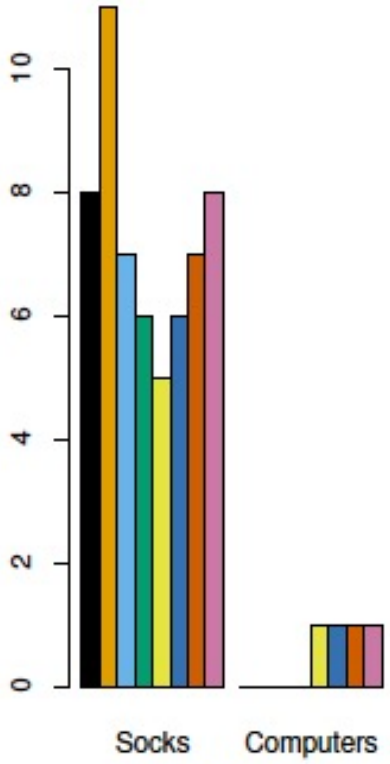
## The online shopping example

- Using Euclidean distance may not be appropriate. Those with same shopping habit but **different shopping volume** should be but may **not** be clustered together.
- Using **correlation-based distance** is more appropriate.
- **Variable scaling**, as in PCA, whether the variables should be standardized is problem specific.
- Example next: High-frequency purchases like socks therefore tend to have a much larger effect on the inter-shopper dissimilarities, and hence on the clustering ultimately obtained, than rare purchases like computers. This may not be desirable.
- Variables measured in different units should be **standardized**.

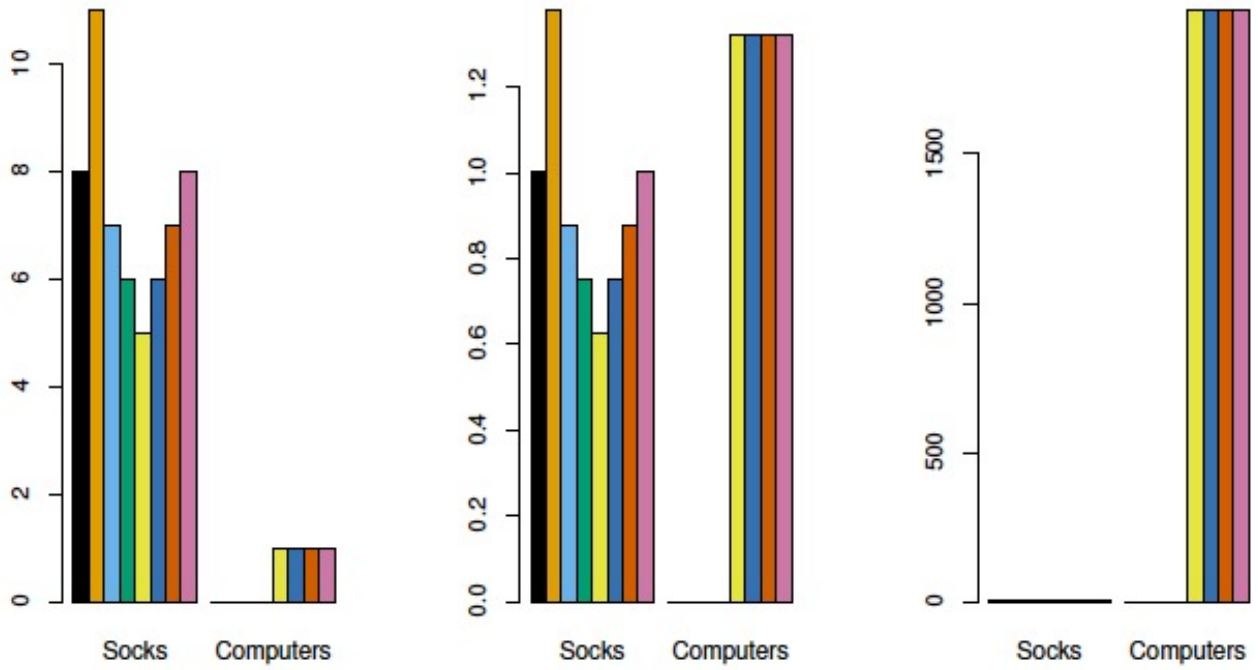
**A Basic Example:**
An online retailer sells two items: **socks and computers.**

The number of pairs of socks, and computers, purchased by eight online shoppers is displayed. Each shopper is shown in a different color.

If inter-observation dissimilarities are computed using Euclidean distance on the raw variables, then the number of socks purchased by an individual will drive the dissimilarities obtained, and the number of computers purchased will have little effect.




This might be undesirable, since (1) computers are more expensive than socks and so the online retailer may be more interested in encouraging shoppers to buy computers than socks, and (2) a large difference in the number of socks purchased by two shoppers may be less informative about the shoppers overall shopping preferences than a small difference in the number of computers purchased.




**Center:** the same data is shown, after **scaling each variable by its standard deviation**. Now the number of computers purchased will have a much greater effect on the inter-observation dissimilarities obtained.

**Right:** the same data are displayed, but now the y-axis represents the number of **dollars** spent by each online shopper on socks and on computers. Since computers are much more expensive than socks, now computer purchase history will drive the inter-observation dissimilarities obtained.

Some questions to consider using clustering:

- Should the observations or features first be **standardized** in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of **hierarchical clustering**,
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
  - Where should we cut the dendrogram in order to obtain clusters?
- In the case of **K-means clustering**, how many clusters should we look for in the data?

Further Remarks:

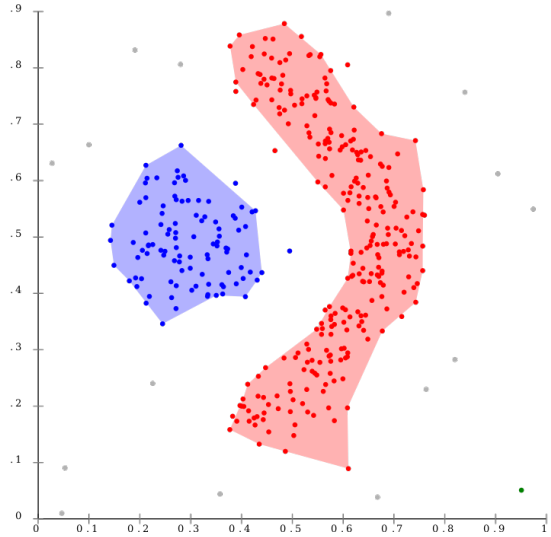
1. Forcing every observation, including outliers, into clusters, can distort the final outcome. (A **soft version** of K-means clustering by mixture model may help)

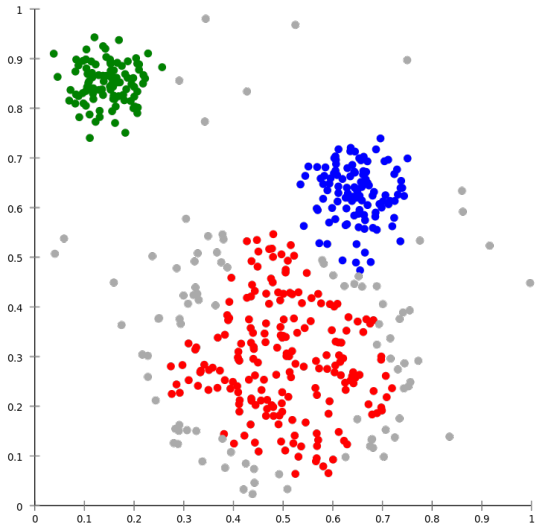
2. Clustering methods generally are **not very robust** to perturbations to the data.

3. Performing clustering with different choices of these parameters (linkage, standardization or not, etc), and looking at the full set of results.

4. Clustering subsets of the data in order to get a sense of the robustness.

❖ **Density based methods** are determined by contiguous regions with density above some threshold.



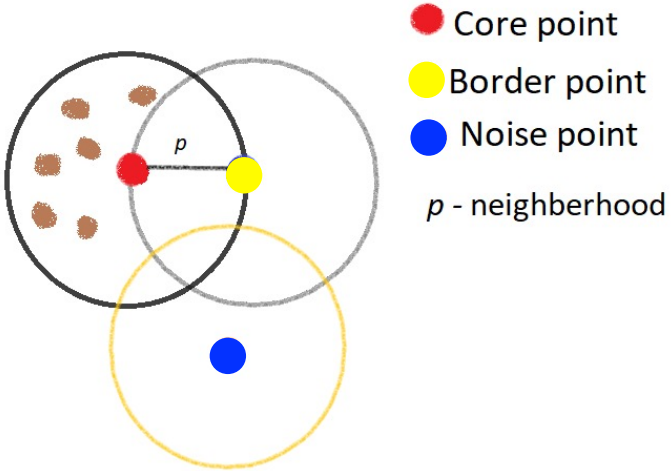


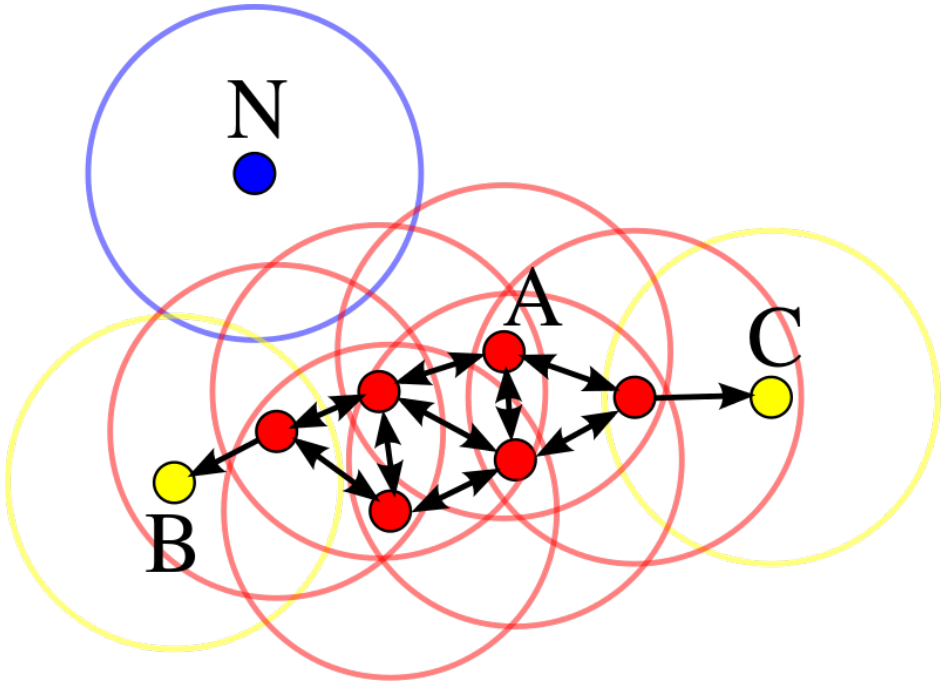
Density based clustering algorithms determine regions by applying a local density threshold (with a dynamically determined or fixed density) and then extracting the connected regions. Algorithms like DBSCAN assume clusters are of similar density and have trouble separating nearby clusters. Algorithms like OPTICS improve upon this.

**DBSCAN** (Density-Based Spatial Clustering of Applications with Noise)

- For a fixed radius $p$
- Fix $n$ to be the number of points required to be in a cluster.

The DBSCAN algorithm classifies points a **core points** like A, **boundary points** like B and C, and **noise points** like N.



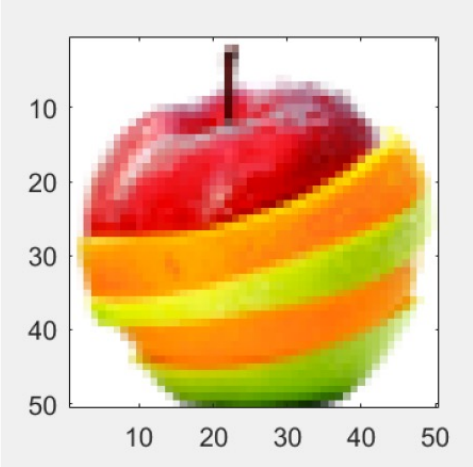
**DBSCAN** has quite a few **advantages**:

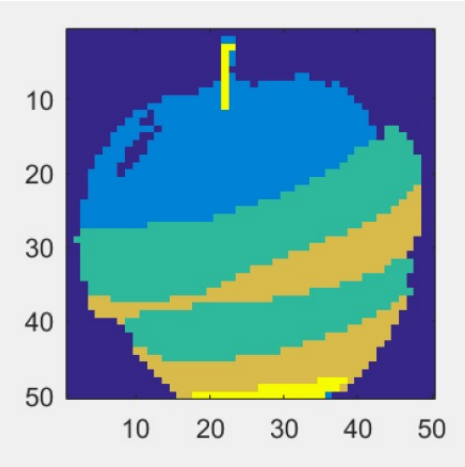
- It does not require the number of clusters K to be chosen beforehand.
- It can find non-circular data, and therefore **nonlinear** decision boundaries.
- It includes a criteria for outliers.
- It only requires setting two meaningful parameters, **p** and n.

**DBSCAN** also has a few **problems**:

- The **robustness** of the search depends heavily on the metric. In particular, if it is Euclidean it falls victim to the curse of dimensionality and may be almost useless.
- If there are large distances in cluster densities, the parameters in DBSCAN may only be tunable to a subset.
- If the data are poorly understood, picking p and n  may be hard.

❖ **Spectral clustering** decomposes the similarity graph $d(x_i, x_j)$ as in factor analysis and PCA.

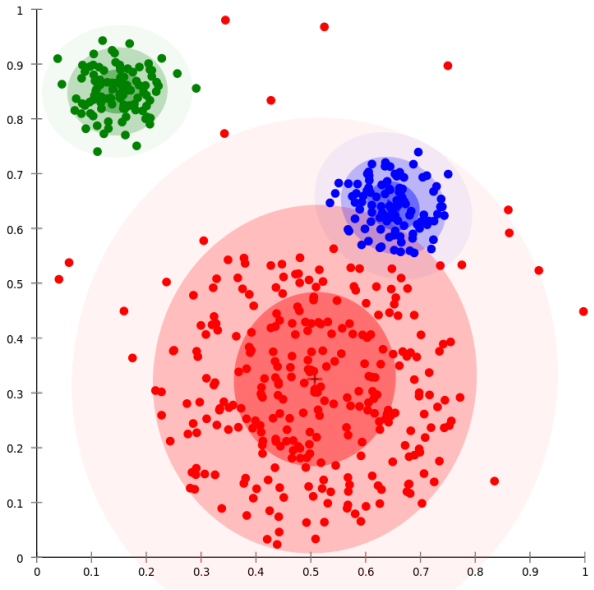


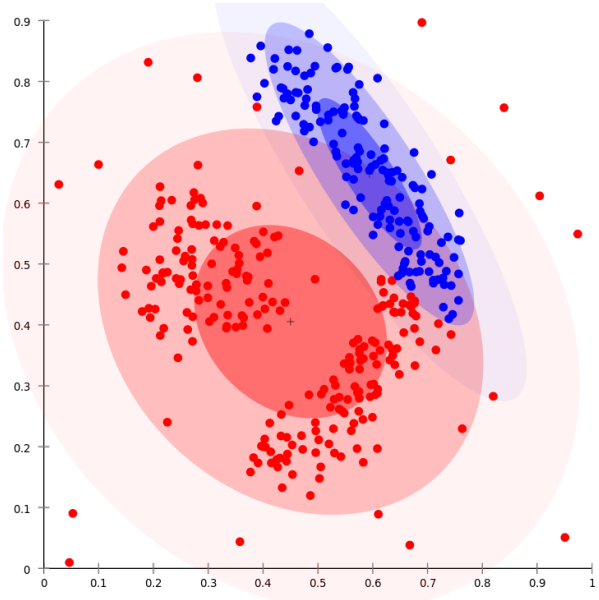




Spectral clustering forms a **similarity matrix** $S_{ij}$ where each entry encode the similarity between datapoints i and j . We then project onto the first $k$ **eigenvectors** of $S_{ij}$ and perform the clustering there. Equivalently, we can construct a similarity graph and thing of pruning away low relevance connections until the graph disconnects.

Sci-kit learn implements this as SpectralClustering.

❖ **Distribution models**/Gaussian mixing models assume clusters provide an underlying distribution on the domain.




Distribution based, or mixing models, try to fit an underlying distribution to the data, often using maximum likelihood estimation.

This is implemented in sci-kit learn as GaussianMixture.

Scikit Learn:

https://scikit-learn.org/stable/modules/clustering.html