Clustering Methods[1]

[1] Based on materials in ISLR 10.3, PRML 9.2, 9.3 and DMA 15.1

Unsupervised Learning

- We only observe the variables or features $X_1, X_2, \ldots, X_p$
- The is no response Y or class labels
- This generally means analysis goals are not clearly defined and we cannot directly validate any findings
- However, there are still many important questions that we can consider. Is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations? Is there any hidden pattern of the data?

Unsupervised Learning

- We only observe the variables or features $X_1, X_2, \ldots, X_p$
- The is no response Y or class labels
- This generally means analysis goals are not clearly defined and we cannot directly validate any findings
- However, there are still many important questions that we can consider. Is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations? Is there any hidden pattern of the data?

Examples:

- groups of shoppers characterized by their browsing and purchase histories
- reduce the dimensionality

Unsupervised Learning

- We only observe the variables or features $X_1, X_2, \ldots, X_p$
- The is no response Y or class labels
- This generally means analysis goals are not clearly defined and we cannot directly validate any findings
- However, there are still many important questions that we can consider. Is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations? Is there any hidden pattern of the data?

Examples:

- groups of shoppers characterized by their browsing and purchase histories
- reduce the dimensionality

We are going to discuss

- Clustering
- PCA
- Community detection in random networks

Clustering Methods

- **Clustering** refers to a very broad set of techniques for finding **subgroups**, or **clusters**, in a data set.
- We seek a partition of the data into distinct groups so that the observations within each group are quite similar to each other.
- To make this concrete, we must define what it means for two or more observations to be **similar** or **different**.
- Indeed, this is often a domain-specific consideration that must be made based on knowledge of the data being studied.

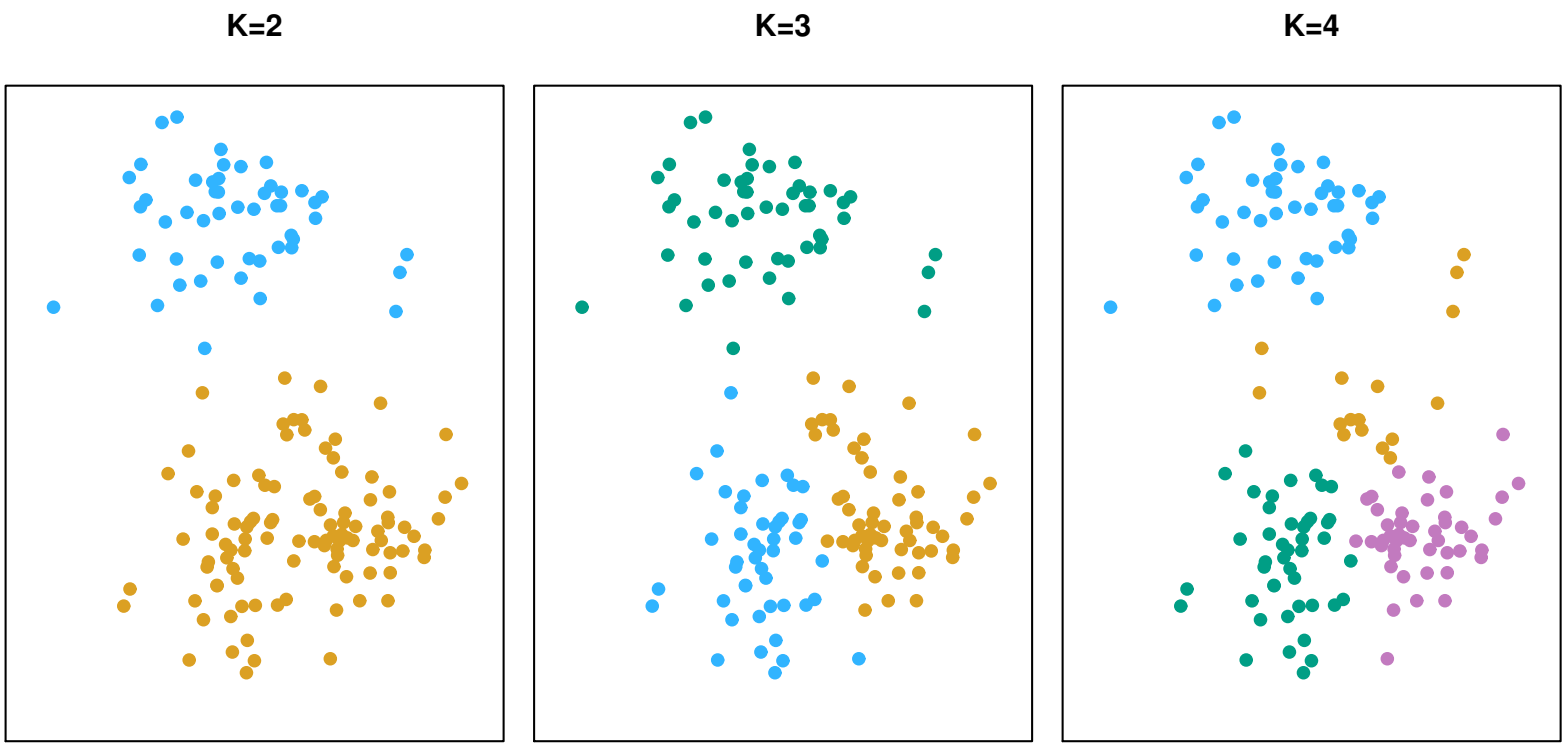
Example: Market Segmentation

- Suppose we have access to a large number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large number of people.
- Our 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.

Clustering Methods

- **K-means clustering:** we seek to partition the observations into a pre-specified number of clusters.
- **Gaussian mixture models:** we look for soft cluster assignment in a probabilistic way so that the level of uncertainty over the most appropriate assignment can be quantified.
- **Density-based clustering:** we do not constrain the shape of the clusters to be ellipsoid or convex; instead, we use local density of points to determine the clusters whose shape can be arbitrary.
- **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.

K-means clustering



A simulated data set with 150 observations in 2-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.

Details of K-means clustering

Let $C_1, \cdots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \cdots, n\}$. In other words, each observation belongs to at least one of the K clusters.
2. $C_k \cap C_{k'} = \varnothing$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

For instance, if the ith observation is in the kth cluster, then $i \in C_k$.

Details of K-means clustering: continued

- The idea behind K-means clustering is that a **good** clustering is one for which the **within-cluster variation** is as small as possible.
- The within-cluster variation for cluster C_k is a measure $\text{WCV}(C_k)$ of the amount by which the observations within a cluster differ from each other.
- Hence we want to solve the problem

$$\text{minimize}_{C_1,\cdots,C_K} \left\{ \sum_{k=1}^{K} \text{WCV}(C_k) \right\}. \qquad (2)$$

- In words, this formula says that we want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible.

How to define within-cluster variation?

- Typically we use Euclidean distance

$$\text{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2, \qquad (3)$$

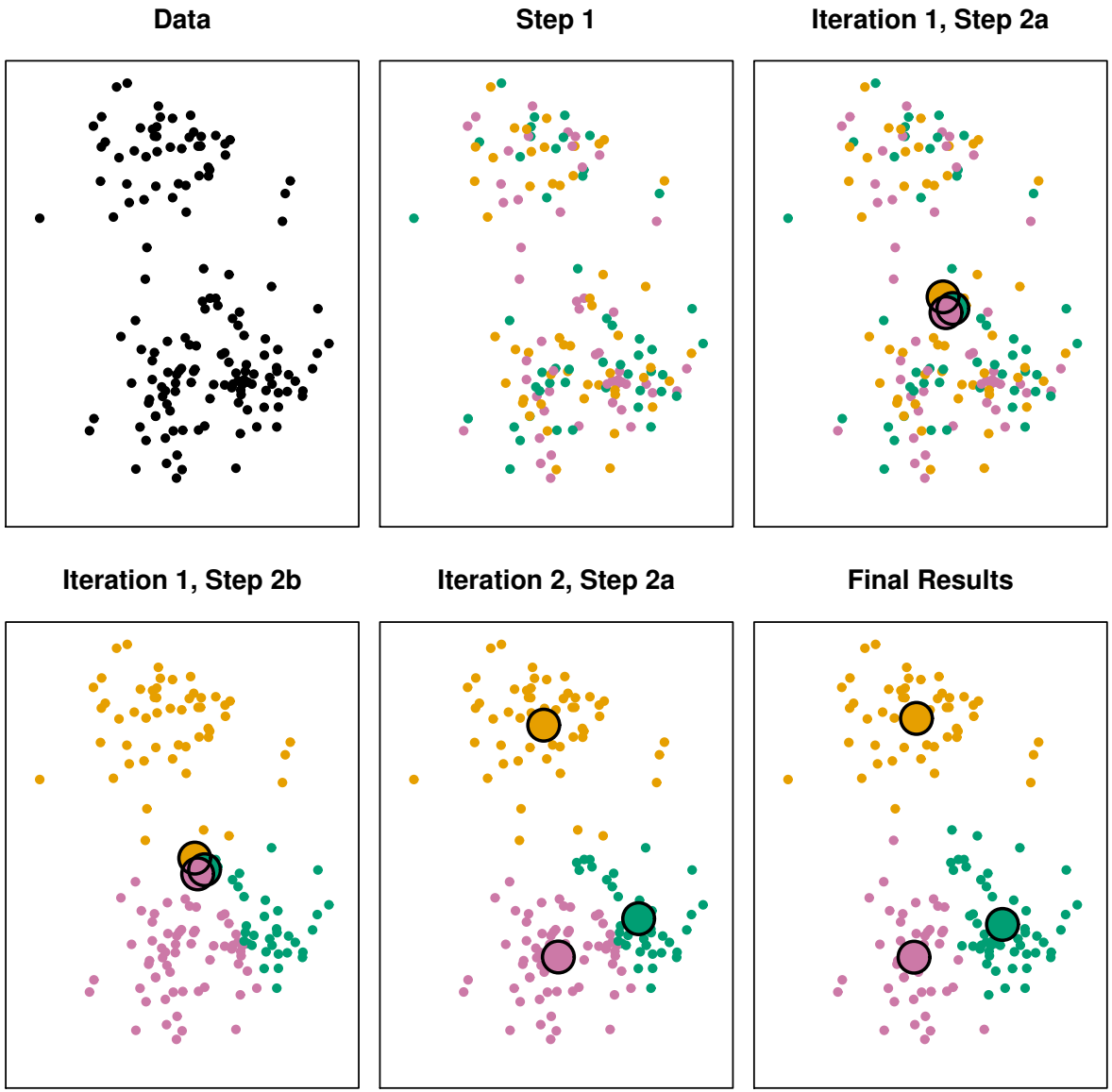
  where $|C_k|$ denotes the number of observations in the kth cluster.
- Combining (2) and (3) gives the optimization problem that defines K-means clustering,

$$\text{minimize}_{C_1,\dots,C_k} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}. \qquad (4)$$

K-Means Clustering 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:
   - (a) For each of the K clusters, compute the cluster **centroid**. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   - (b) Assign each observation to the cluster whose centroid is closest (where **closest** is defined using Euclidean distance).

Example


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective (4) at each step. **Why**? Note that

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2,$$

  where $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i\in C_k} x_{ij}$ is the mean for feature j in cluster C_k.
- However, it is not guaranteed to give the global minimum since the objective is not convex. **We may want to run K-means multiple times with different initializations.**

Example: different starting values

Details of Previous Figure

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

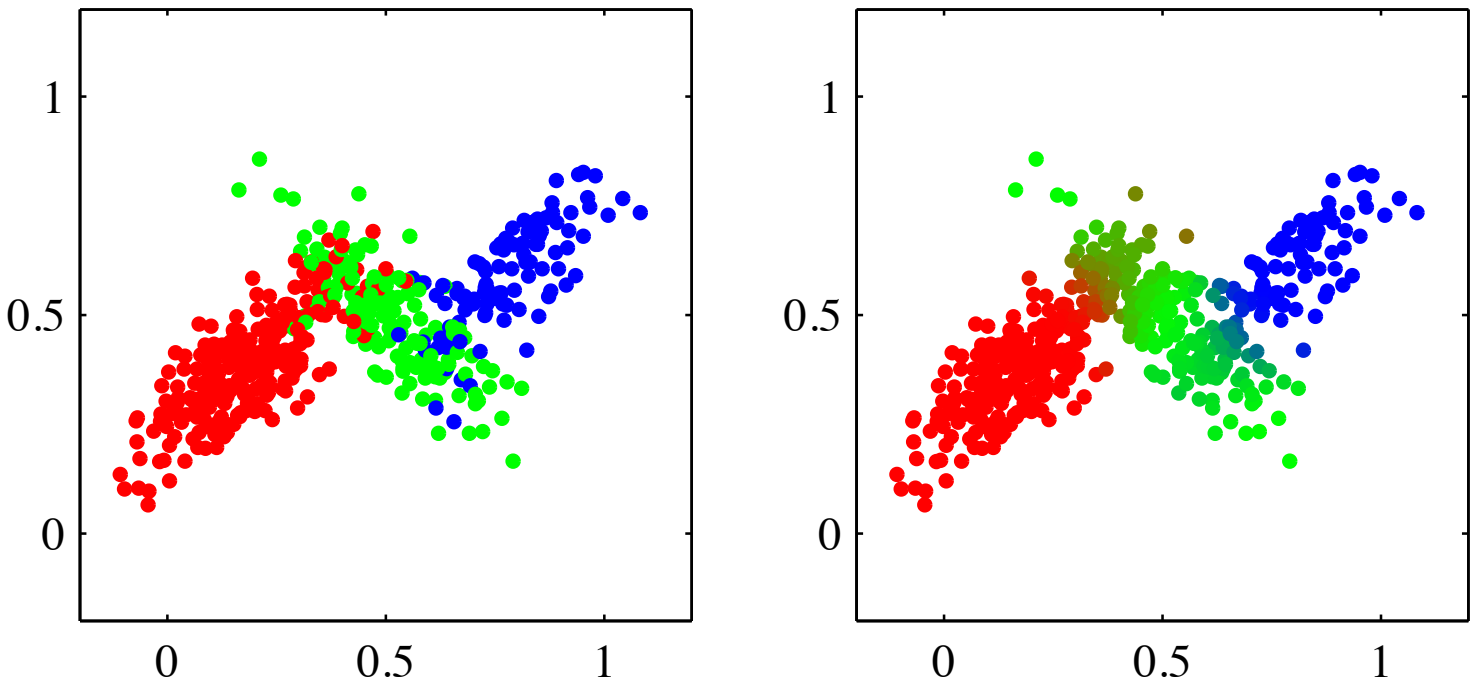
Above each plot is the value of the objective (4). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8

Mixture models

- K-means assigns each data point uniquely to one and only one cluster. Some data points may lie roughly midway between cluster centroids. It is not clear that the **hard** assignment to the nearest cluster is the most appropriate.
- **Mixture models** adopt a probabilistic approach and obtain **soft** assignments of data points to clusters in a way that reflects the level of **uncertainty** of cluster assignment.
- In this section, we introduce **Gaussian** mixture models, one of the most commonly used mixture models.

Gaussian mixture models: soft assignment

A simulated data set with 500 observations drawn from the mixture of three 2-dimensional Gaussians.



- *Left*: true labels indicated by red, green and blue.
- *Right*: soft assignment with proportions of red, blue, and green colors.

Gaussian mixture models: definition

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ and let $N(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a multivariate Gaussian density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. A Gaussian mixture model with K components is a weighted average of K Gaussian densities

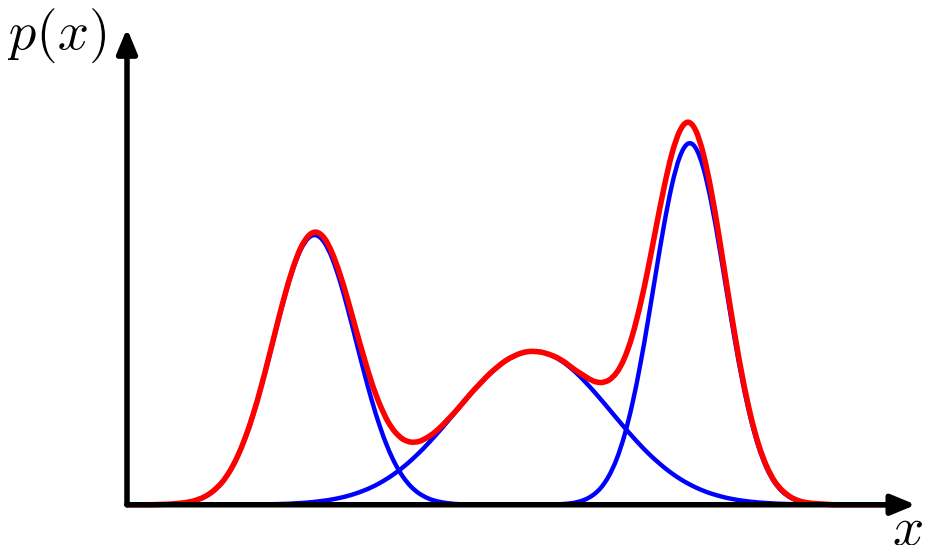
$$p(\boldsymbol{x}_i) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

with

$$\sum_{k=1}^{K} \pi_k = 1$$

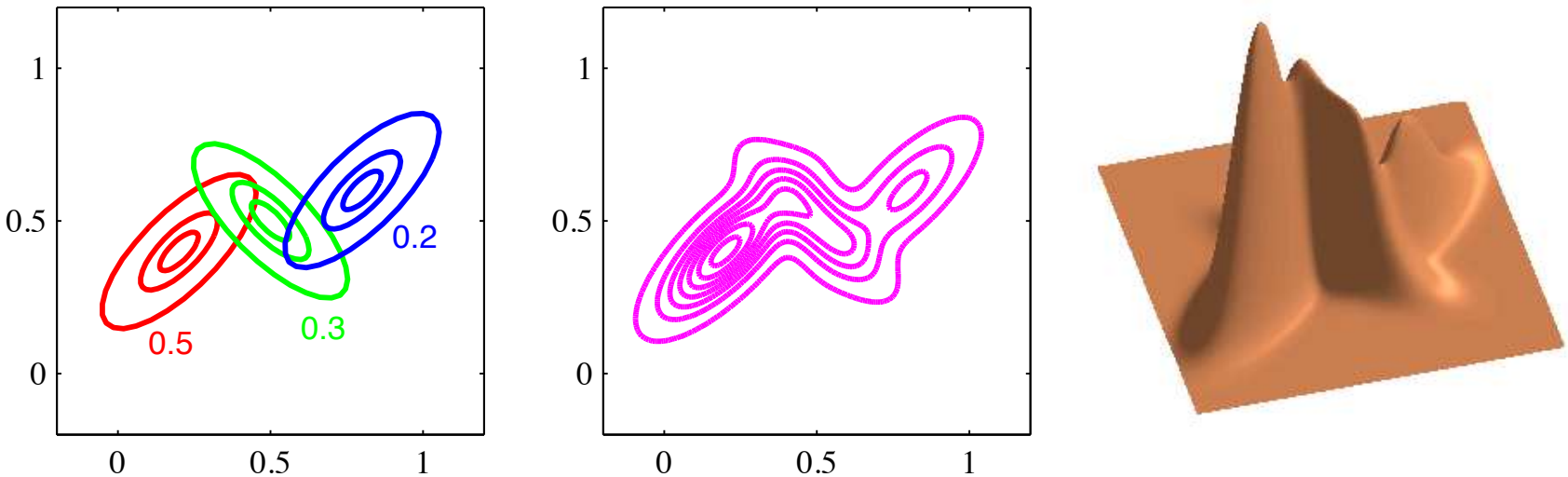
- $N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is a **component** of the mixture.
- $0 \leq \pi_k \leq 1$ is the **mixture weight** or **mixing coefficients**.

One-dimensional example



Three Gaussians (each scaled by a mixture weight) in blue and their sum in red.

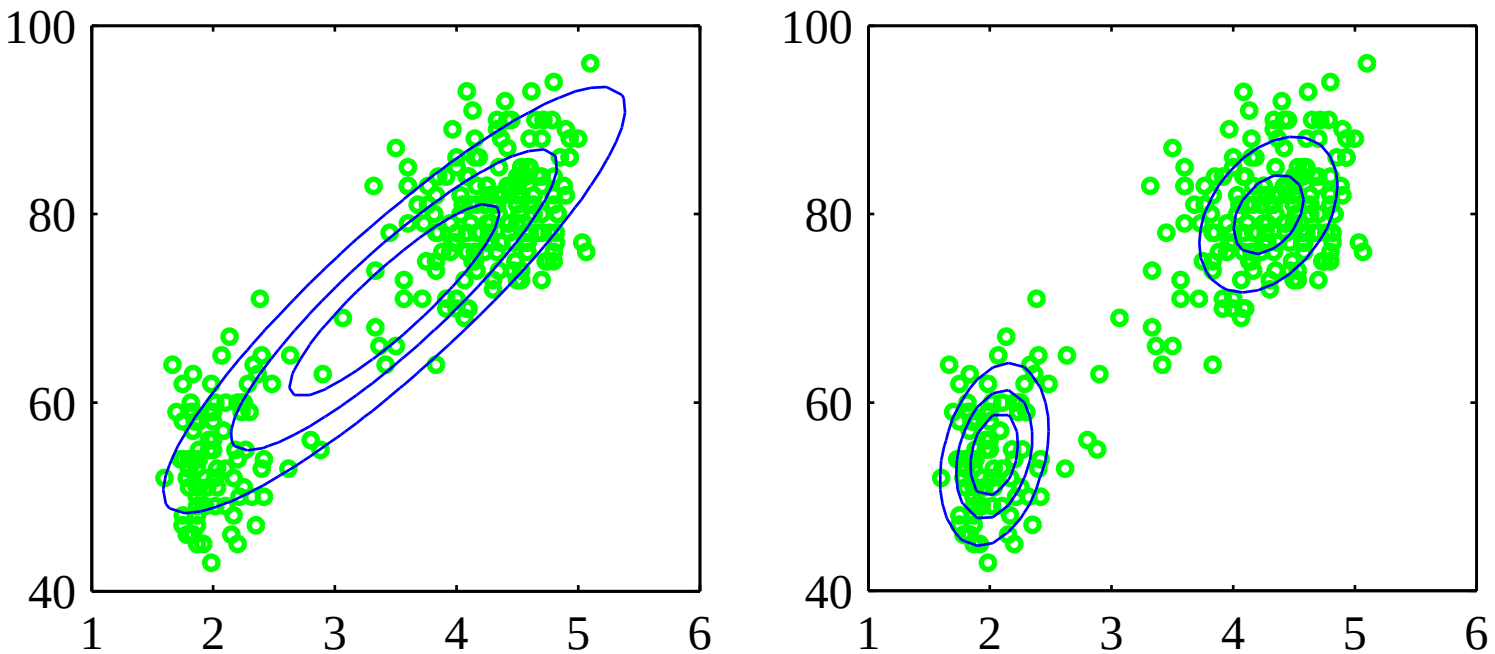
Two-dimensional example



- *Left*: contours of constant density for each of the 3 Gaussians (red, blue and green), and the mixture weights (numbers below each component).
- *Center*: contour of the weighted average of 3 Gaussians.
- *Right*: surface plot of the weighted average of 3 Gaussians.

Old faithful data

Old faithful geyser at Yellowstone. Eruption duration in minutes (x-axis) vs lapse time in minutes (y-axis)



- *Left*: a single Gaussian distribution. This distribution fails to capture the two clumps in the data and indeed places much of its probability mass in the central region between the clumps where the data are relatively sparse.
- *Right*: two Gaussians which seem to fit better. **Each Gaussian forms a cluster!**

Latent variable representation

- To make the connection between Gaussian mixtures and clustering concrete, we introduce a latent variable $s_i \in \{1, \ldots, K\}$ for each observation i and assume

$$p(s_i = k) = \pi_k \tag{1}$$
$$p(\boldsymbol{x}_i | s_i = k) = N(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2}$$

- The marginal distribution of (1) and (2)

$$p(\boldsymbol{x}_i) = \sum_{k=1}^{K} p(\boldsymbol{x}_i | s_i = k) p(s_i = k)$$

  is equivalent to the weighted average representation.

- **Interpretation**: $s_i = k$ indicates observation i belongs to cluster k.

Soft assignment through Bayes theorem

- Since we have a proper probability model, we can calculate conditional probability $p(s_i = k|\boldsymbol{x}_i)$ using Bayes theorem

$$p(s_i = k|\boldsymbol{x}_i) = \frac{\pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l N(\boldsymbol{x}_i|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

- **Soft assignment**: $p(s_i = k|\boldsymbol{x}_i)$ is the posterior probability that observation i belongs to cluster k.

Estimation through EM algorithm

Expectation-maximization algorithm:

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixture weights π_k
2. E step. Evaluate the posterior probabilities using the current parameter values

$$p(s_i = k|\boldsymbol{x}_i) = \frac{\pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l N(\boldsymbol{x}_i|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

3. M step. Re-estimate the parameters using the posterior probabilities

$$\boldsymbol{\mu}_k^{new} = \frac{1}{n_k} \sum_{i=1}^{n} p(s_i = k|\boldsymbol{x}_i)\boldsymbol{x}_i$$

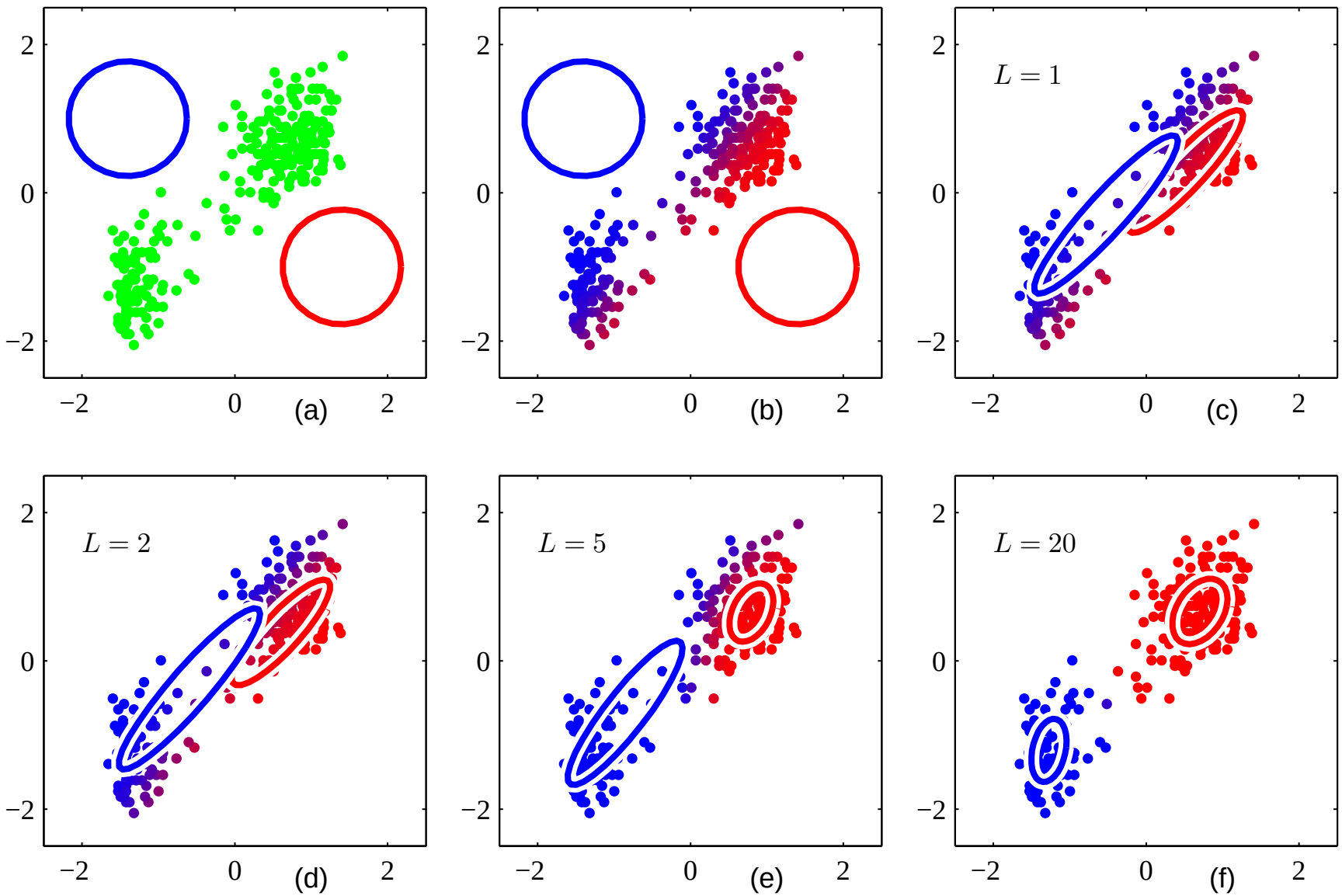
$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{n_k} \sum_{i=1}^{n} p(s_i = k|\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})^T$$

$$\pi_k^{new} = \frac{n_k}{n}$$

   where $n_k = \sum_{i=1}^{n} p(s_i = k|\boldsymbol{x}_i)$ is the effective number of observations assigned to cluster k.

4. Check convergence (e.g. through parameters or likelihood). If not convergent, return to step 2.

Illustration of EM algorithm



Comparison with K-means

- K-means outputs hard cluster assignment whereas Gaussian mixture model outputs soft cluster assignment.
- If $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}_p$ (variance times an identity matrix) for all $k = 1, \ldots, K$, when ϵ is small, the soft assignment and hard assignment are similar

$$p(s_i = k|\boldsymbol{x}_i) \approx 1 \text{ for } k = \arg\min_k \sum_{j=1}^{p} (x_{ij} - \mu_{kj})^2$$

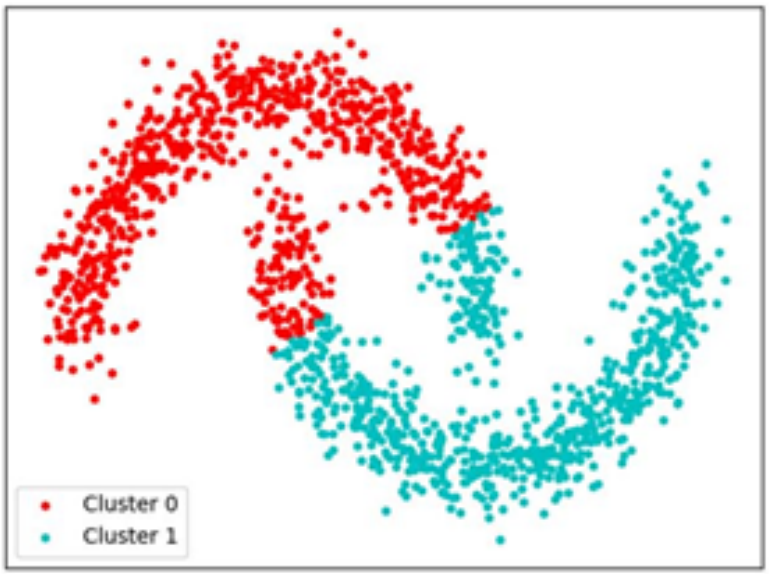
$$p(s_i = l|\boldsymbol{x}_i) \approx 0 \text{ for } l \neq k$$

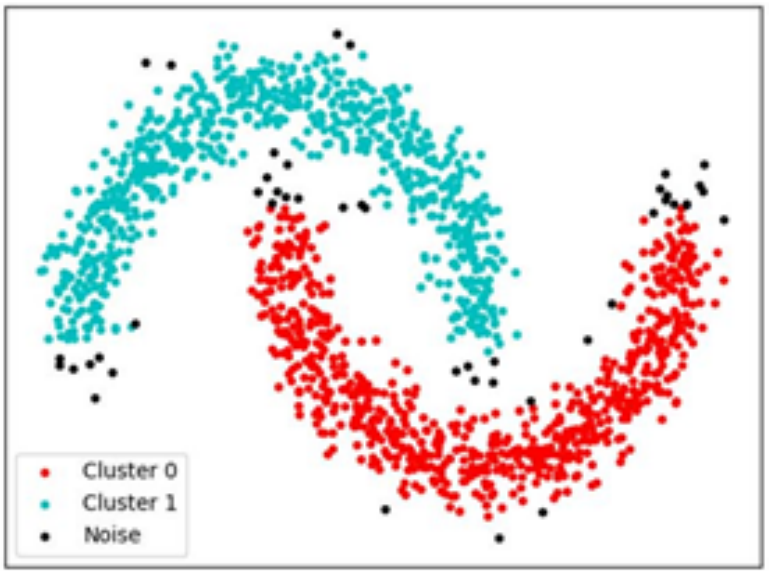
Essentially, each observation is assigned to the cluster having the closest mean.

Density-based clustering

- K-means and Gaussian mixtures assume ellipsoidal/convex clusters.
- In practice, we can have irregular-shaped/**non-convex** clusters.
- Two observations from different clusters may be closer than two observations in the same cluster.
- **Density-based clustering** is able to discover non-convex clusters.
- Next, we introduce **density-based spatial clustering of applications with noise (DBSCAN)** algorithm.

Non-convex shaped clusters



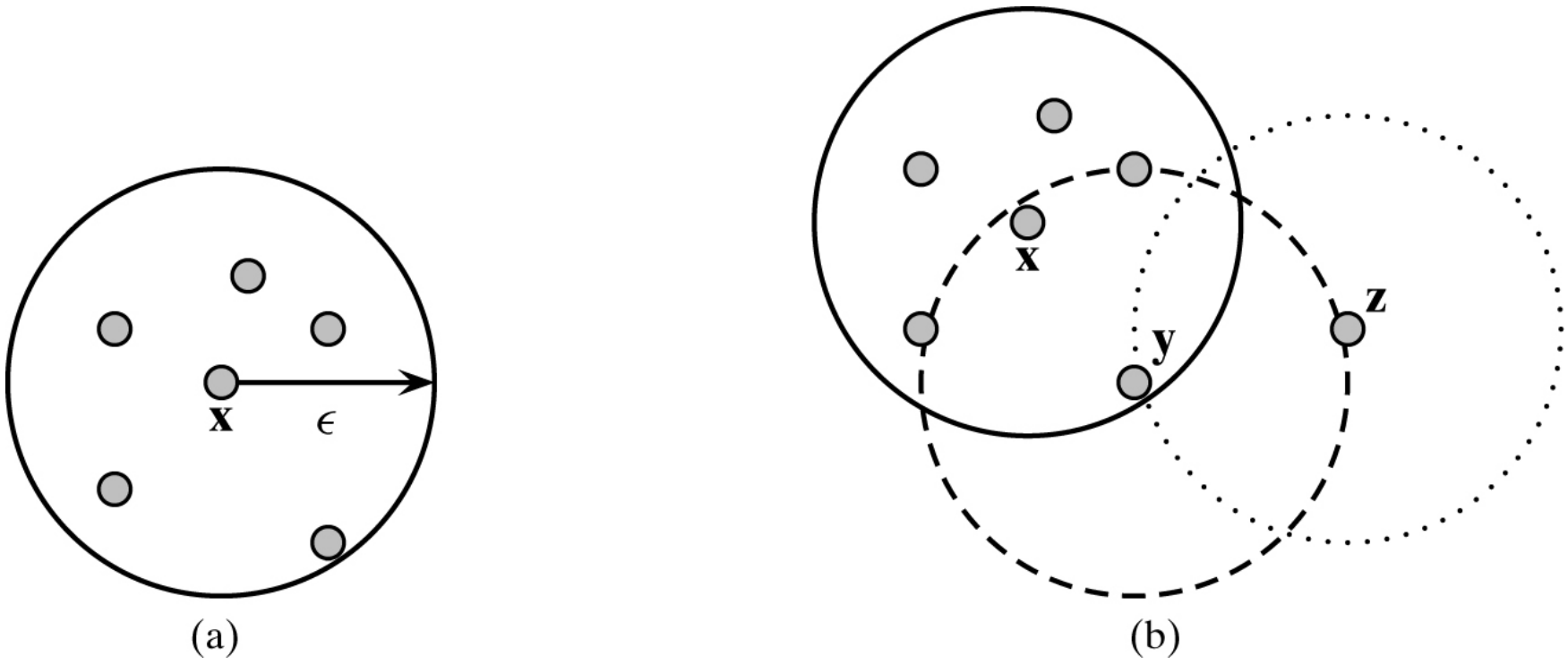


Core, border and noise points

ϵ-**neighborhood**: a ball of radius ϵ around a point $\boldsymbol{x} = (x_1, \ldots, x_p)$

$$N_\epsilon(\boldsymbol{x}) = \left\{ \boldsymbol{y} \middle| \sum_{j=1}^{p} (x_j - y_j)^2 \leq \epsilon \right\}$$

Core points: $\boldsymbol{x}$ is a core point if $|N_\epsilon(\boldsymbol{x})| \geq m$, i.e. there are at least m points in its ϵ-neighborhood.
Border points: $\boldsymbol{x}$ is a border point if it is not a core point and it belongs to the ϵ-neighborhood of some core point $\boldsymbol{z}$, that is, $\boldsymbol{x} \in N_\epsilon(\boldsymbol{z})$.
Noise points: neither a core nor a border point.

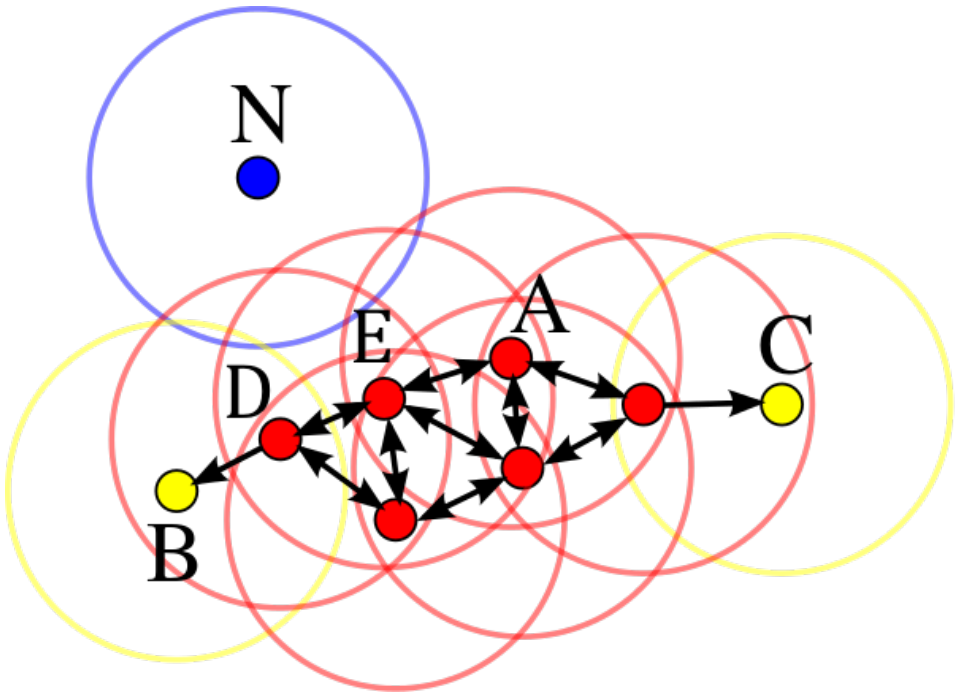


(a) ϵ-neighborhood of x. (b) Core point x, border point y, and noise point z with $m = 6$.

Reachability

Directly density reachable: $\boldsymbol{x}$ is directly density reachable from another point $\boldsymbol{y}$ if $\boldsymbol{x} \in N_\epsilon(\boldsymbol{y})$ and $\boldsymbol{y}$ is a core point.
Density reachable: $\boldsymbol{x}$ is density reachable from $\boldsymbol{y}$ if there is set of core points leading from $\boldsymbol{y}$ to $\boldsymbol{x}$, i.e. there exists a sequence of points $\boldsymbol{y} = \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_l = \boldsymbol{x}$ such that $\boldsymbol{x}_i$ is directly density reachable from $\boldsymbol{x}_{i-1}$ for all $i = 1, \ldots, l$.
Density connected: $\boldsymbol{x}$ and $\boldsymbol{y}$ are density connected if there exists a core point $\boldsymbol{z}$ such that both $\boldsymbol{x}$ and $\boldsymbol{y}$ are density reachable from $\boldsymbol{z}$.

Illustration



$m = 4$. Point A and the other red points are core points, because the ϵ-neighborhood of these points contains at least 4 points (including the point itself). B (border point) is directly density reachable from D since B is in the ϵ-neighborhood of a core point D. B is density reachable from A because B is directly density reachable from D, D is directly density reachable from E, and E is directly density reachable from A. B and C are density connected since they are both density reachable from A. N is a noise point.

DBSCAN algorithm

A density-based cluster is defined as a maximal set of density connected points.

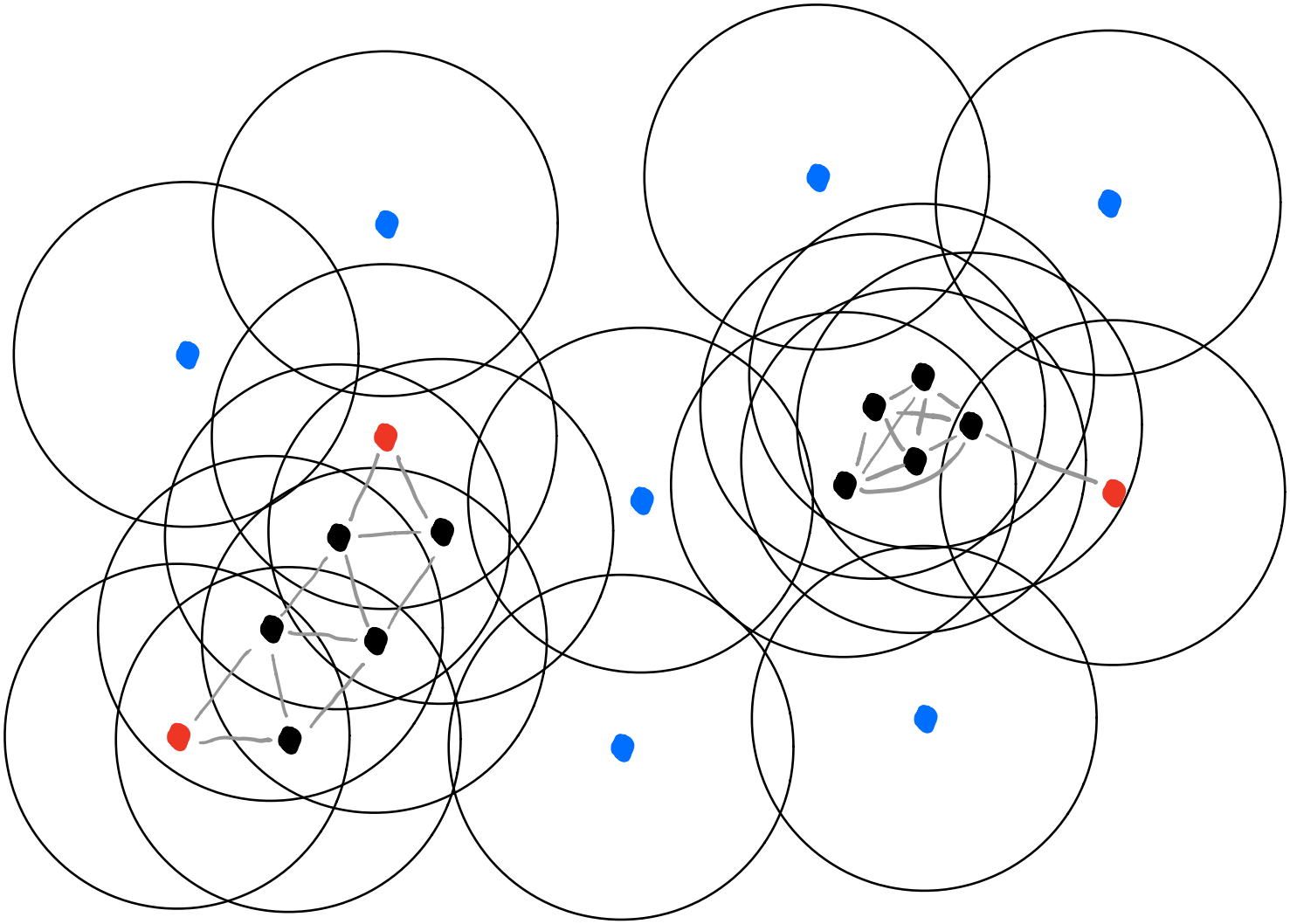
1. Compute the ϵ-neighborhood $N_\epsilon(\boldsymbol{x}_i)$ for each observation $\boldsymbol{x}_i$ and checks if it is a core point.
2. For each core point, if it's not already assigned to a cluster, create a new cluster. Find recursively all its density connected points and assign them to the same cluster as the core point.
3. Observations that do not belong to any cluster are noise points.

In plain English..

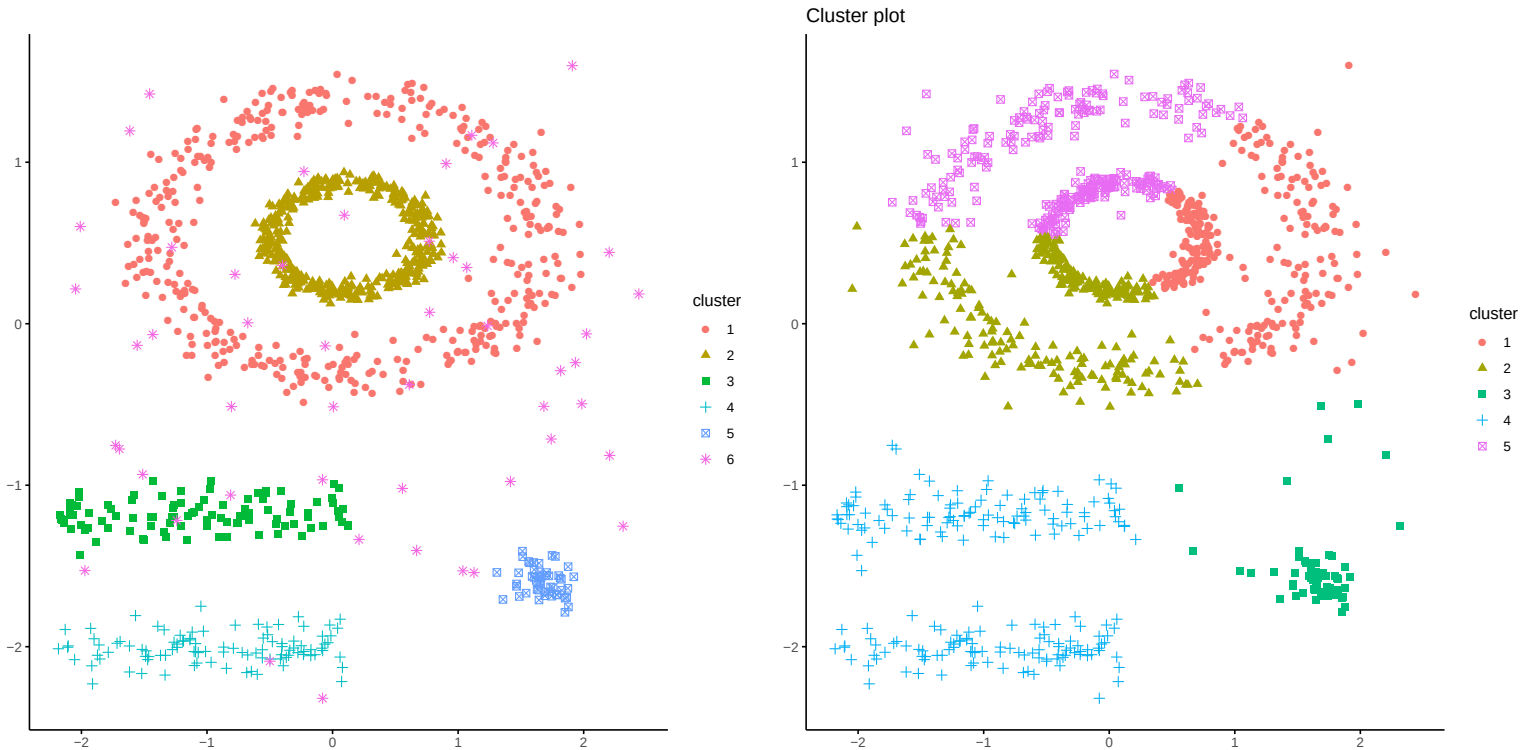
We begin by picking an arbitrary point in our dataset. If there are more than m points within a distance of ϵ from that point, (including the original point itself), we consider all of them to be part of a "cluster". We then expand that cluster by checking all of the new points and seeing if they too have more than m points within a distance of ϵ, growing the cluster recursively if so.

Eventually, we run out of points to add to the cluster. We then pick a new arbitrary point and repeat the process. Now, it's entirely possible that a point we pick has fewer than m points in its ϵ ball, and is also not a part of any other cluster. If that is the case, it's considered a noise point not belonging to any cluster.

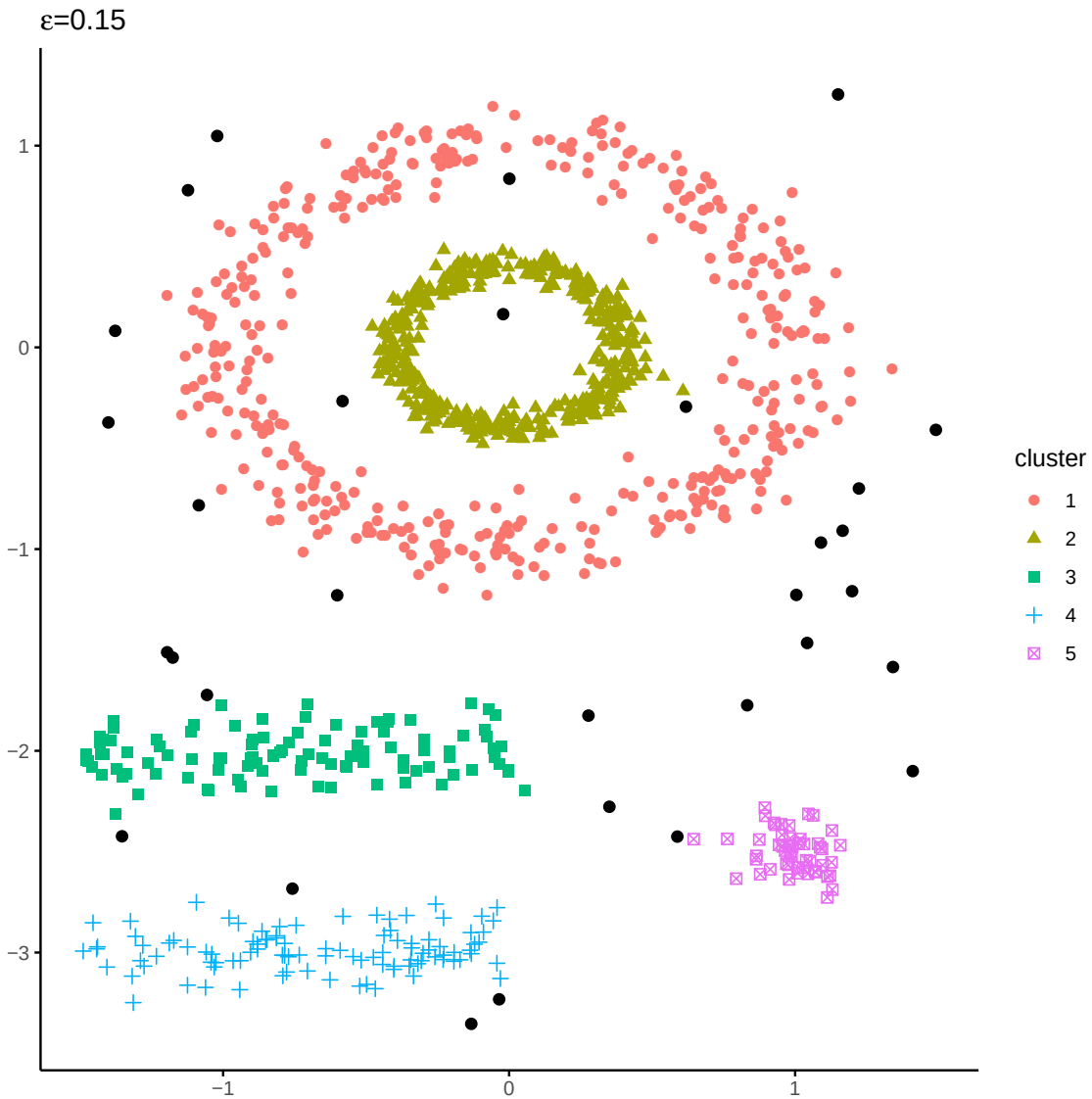
Illustration



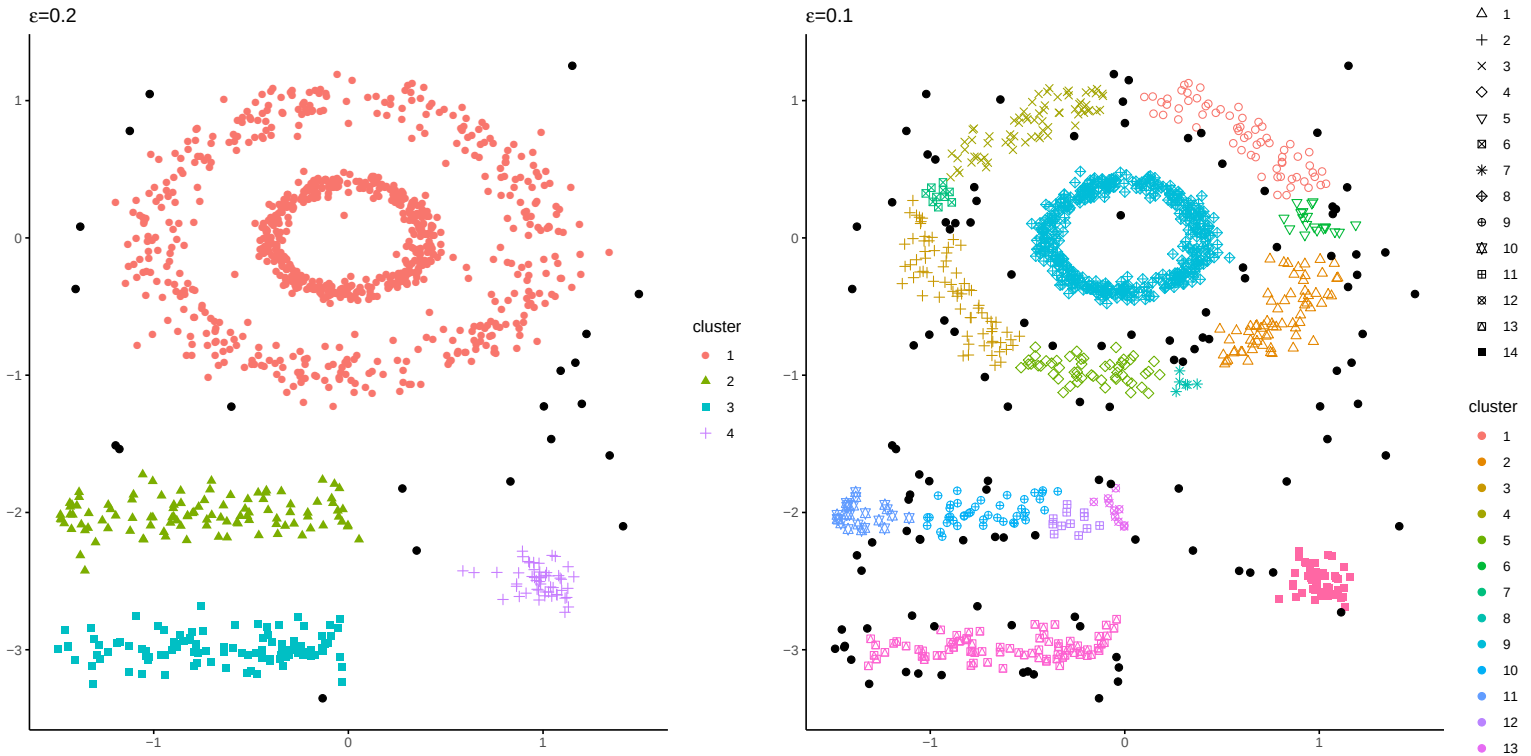
Simulated example



DBSCAN



But it's quite sensitive to the choice of ϵ



More examples of DBSCAN

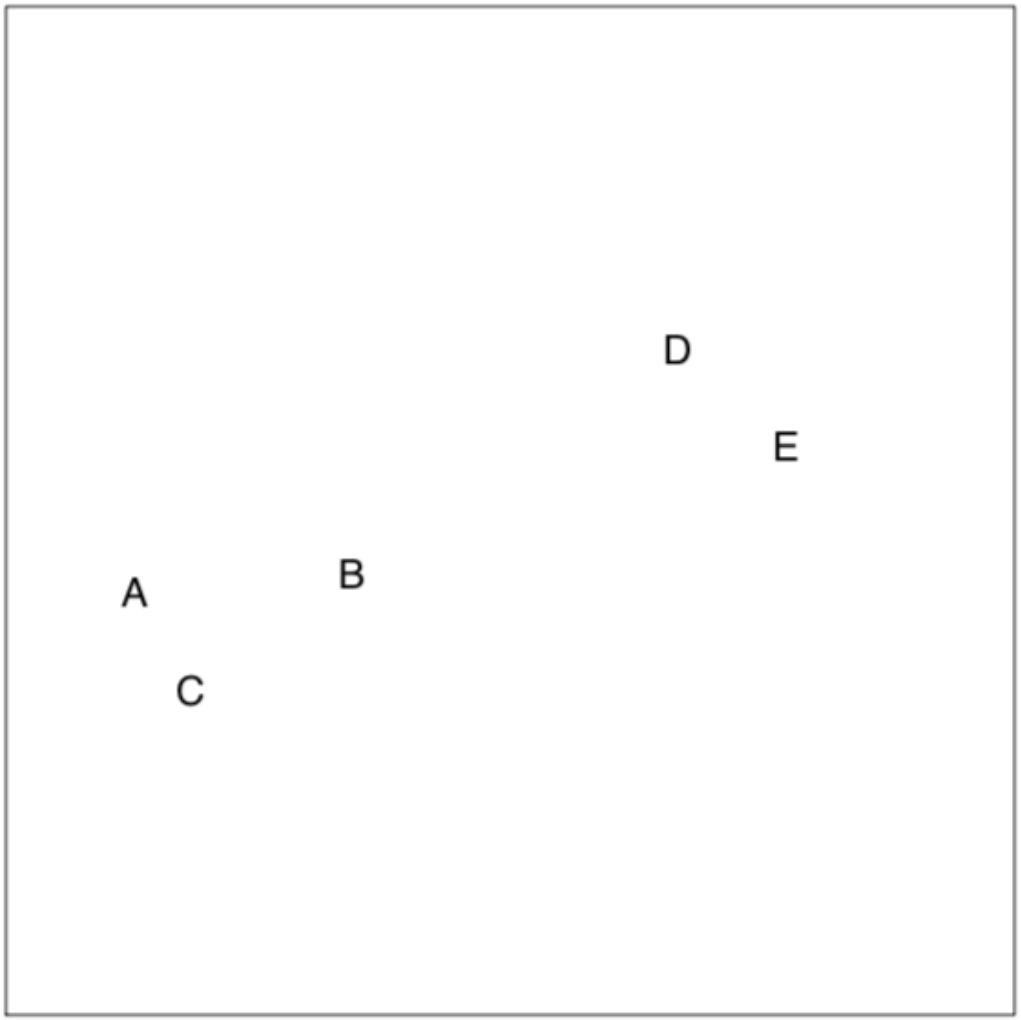
https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Hierarchical Clustering

- K-means clustering, mixture models and density-based clustering require us to **pre-specify** the number of clusters K. This can be a disadvantage.
- **Hierarchical clustering** is an alternative approach which does not require that we commit to a particular choice of K.
- In this section, we describe **bottom-up** or **agglomerative** clustering. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk.

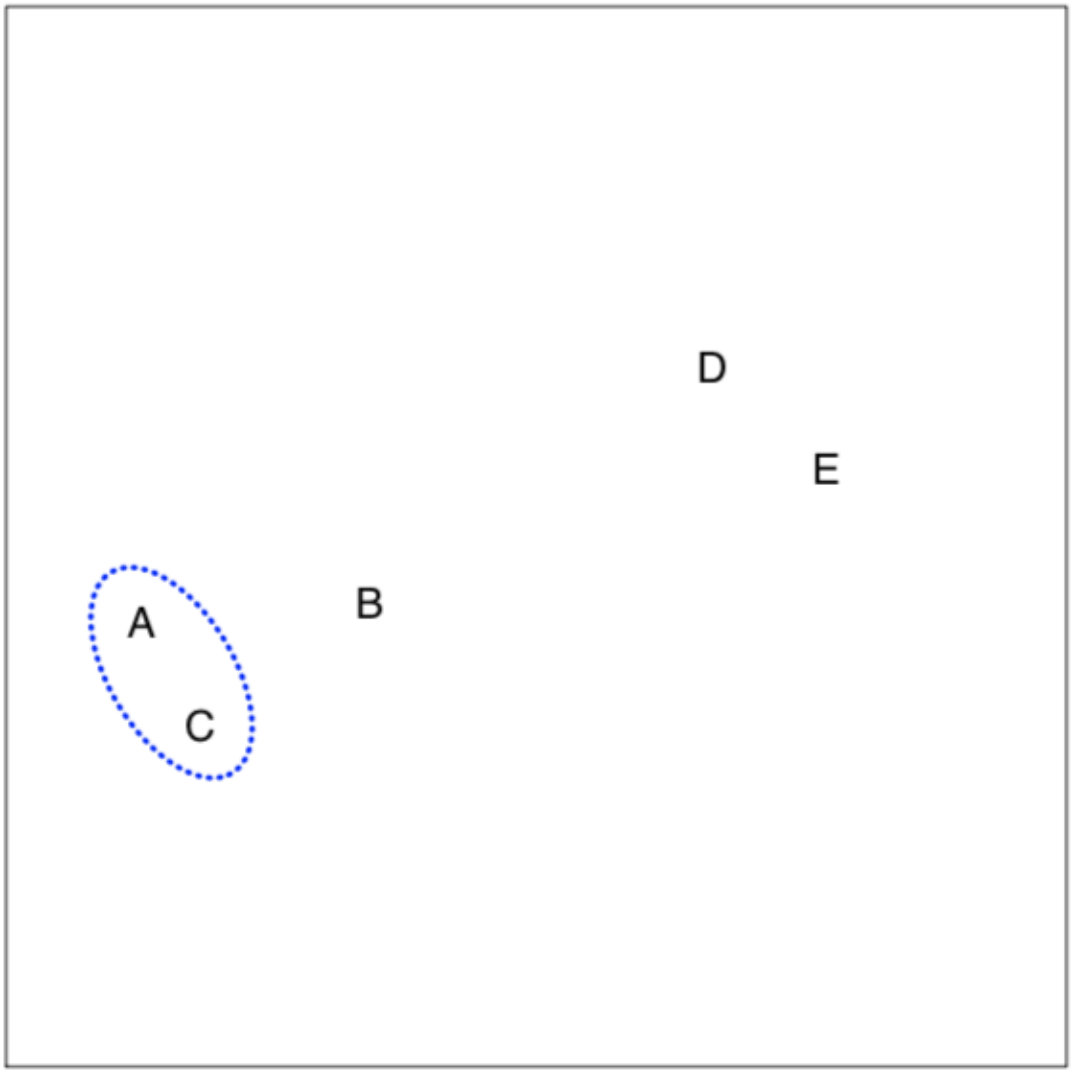
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



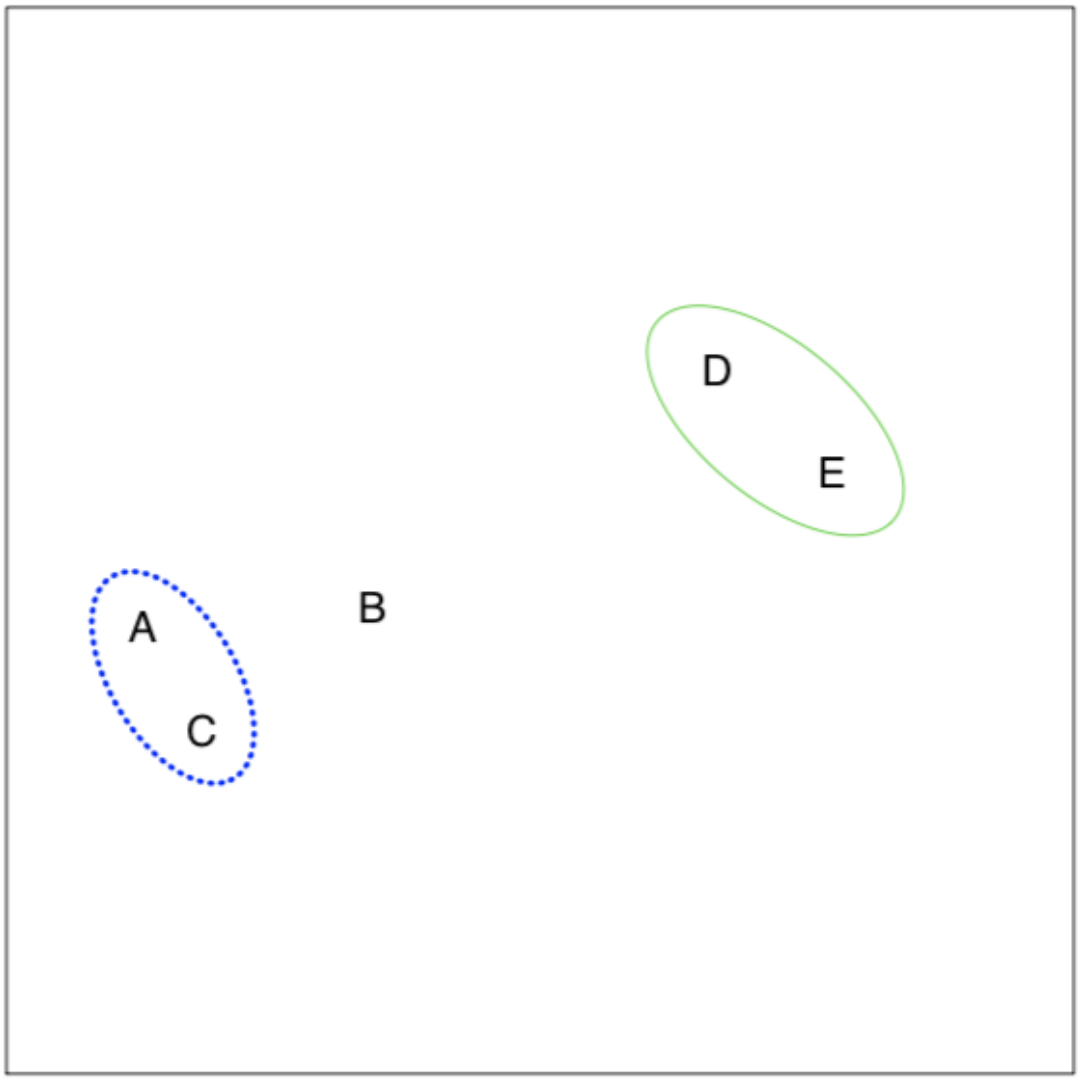
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



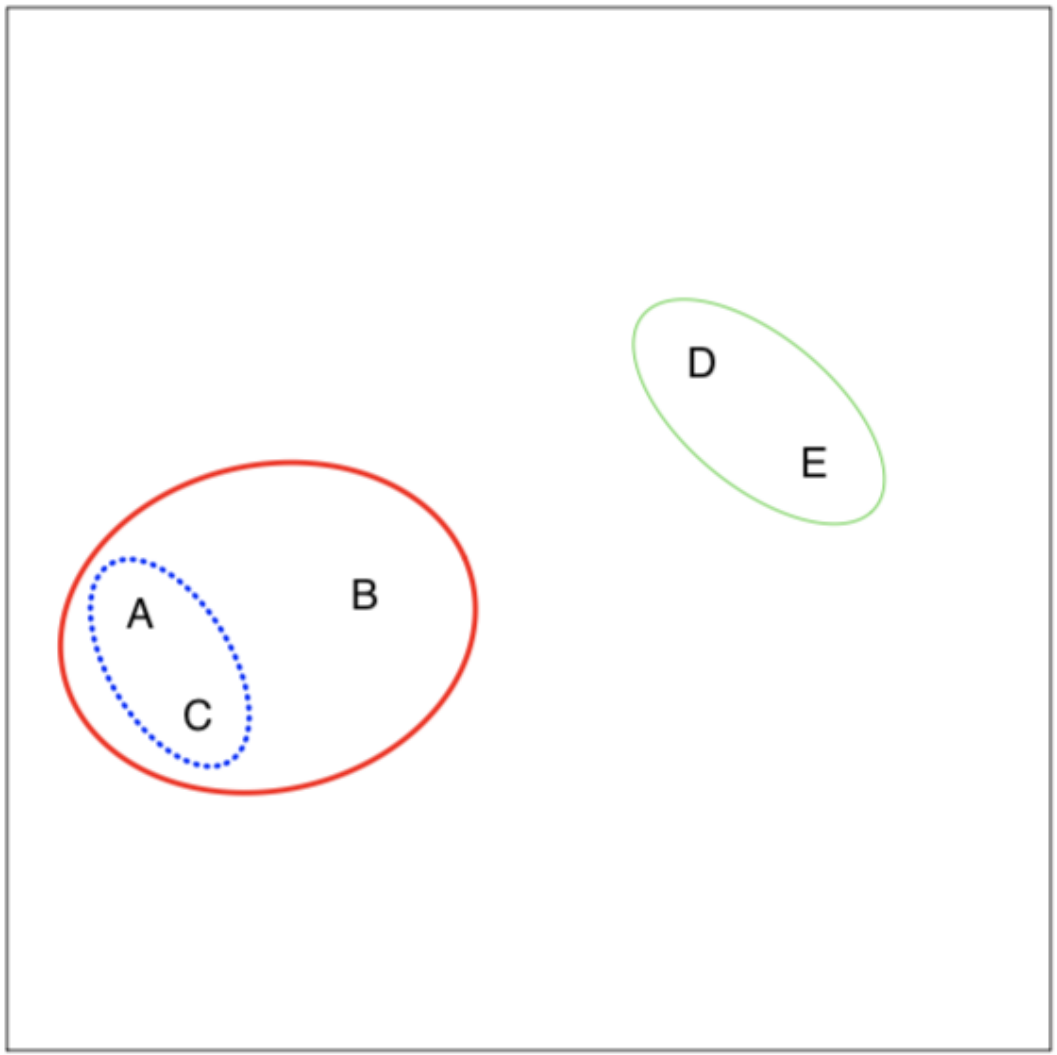
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



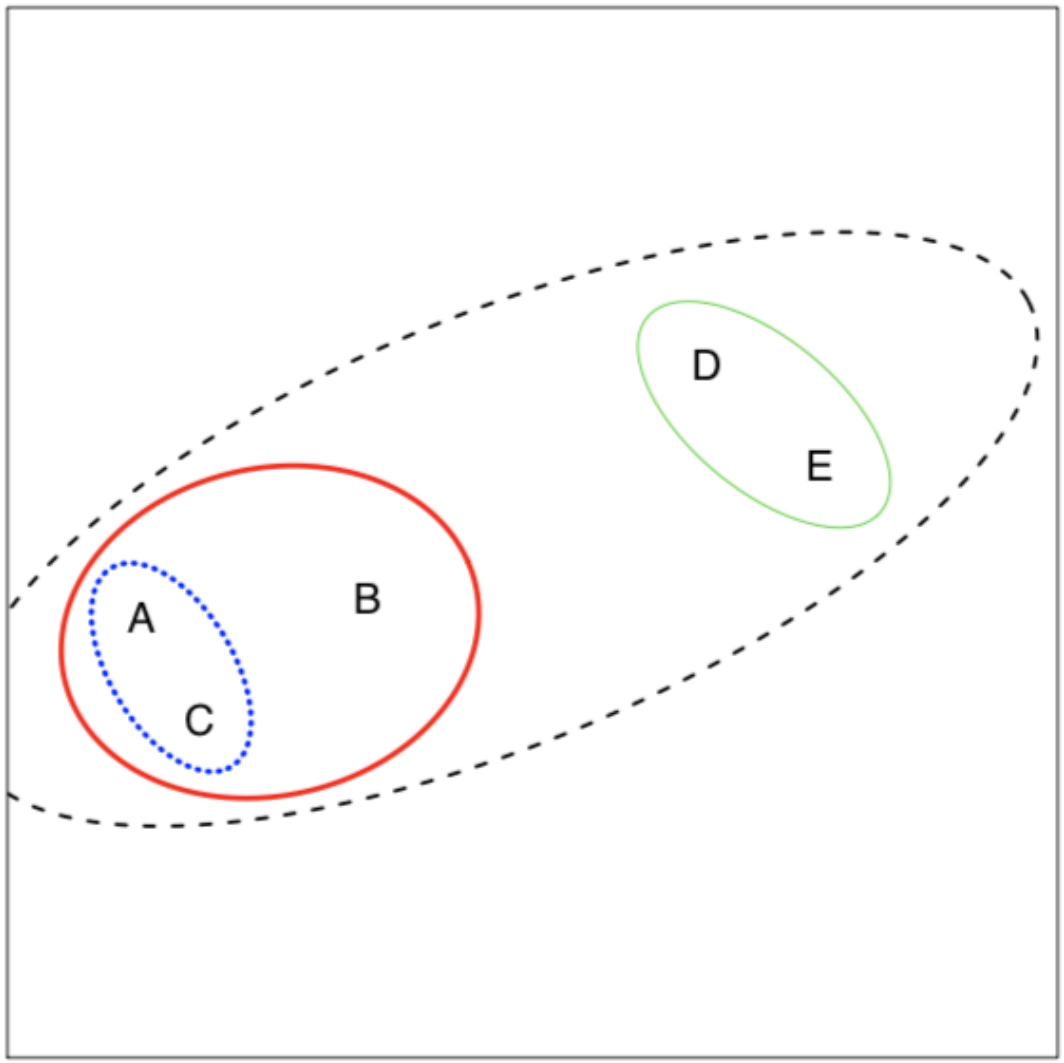
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



Hierarchical Clustering: the idea

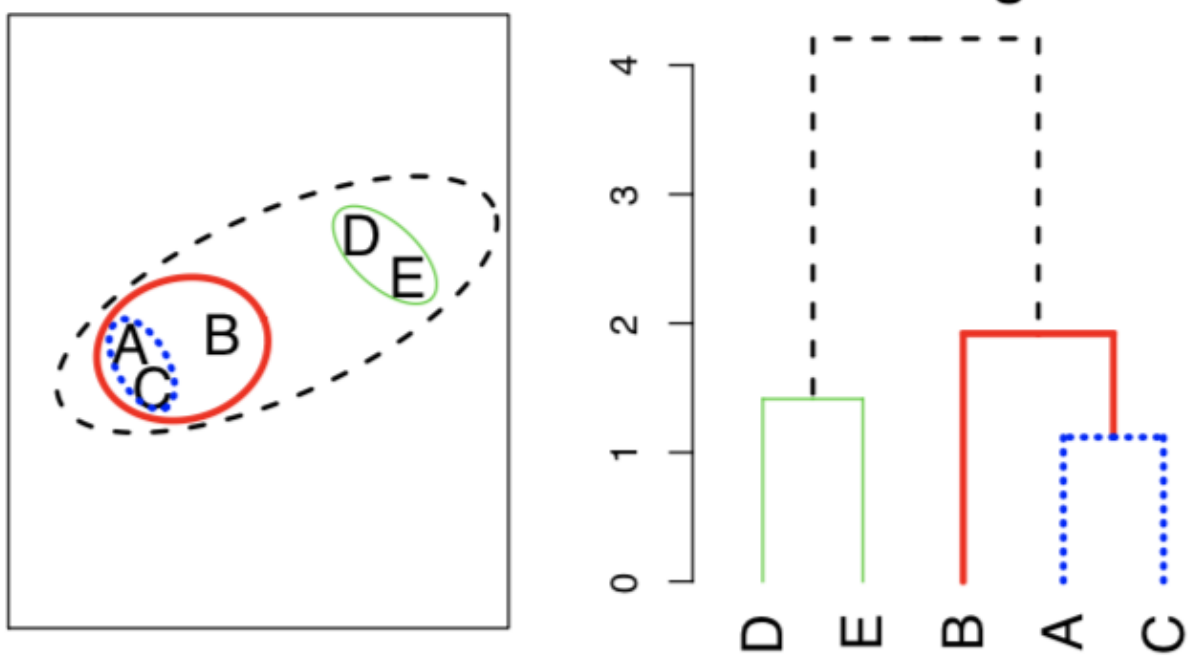
Builds a hierarchy in a "bottom-up" fashion...



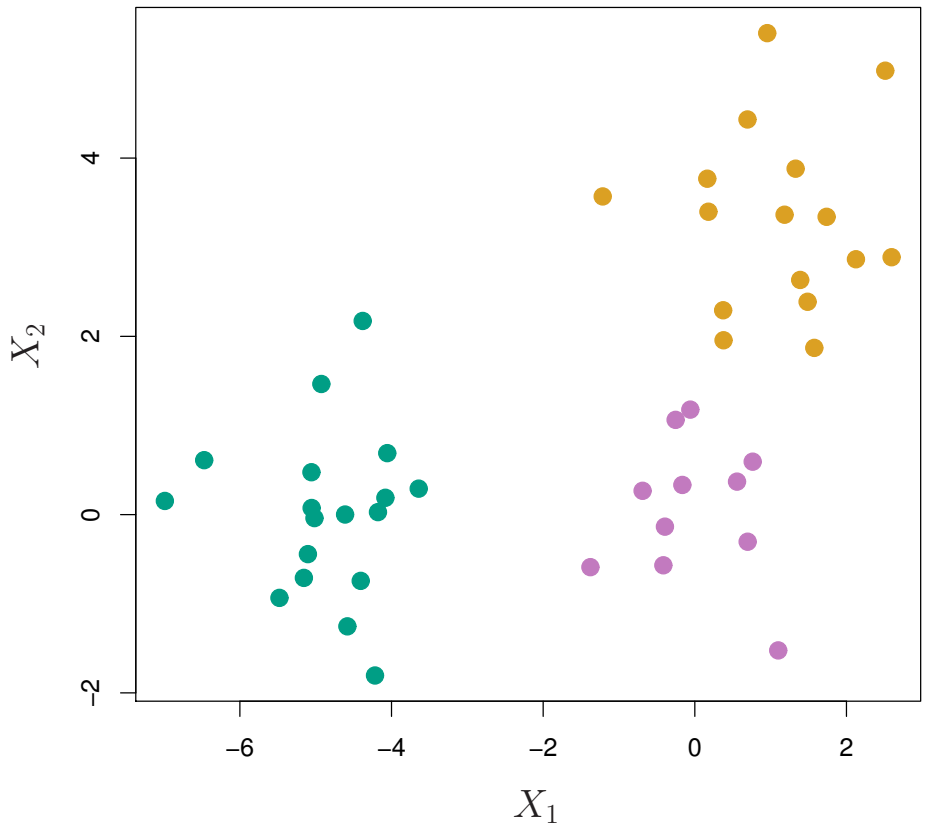
Hierarchical Clustering Algorithm

The approach in words:

1. Start with each point in its own cluster.
2. Identify the "closest" two clusters and merge them.
3. Repeat.
4. Ends when all points are in a single cluster.

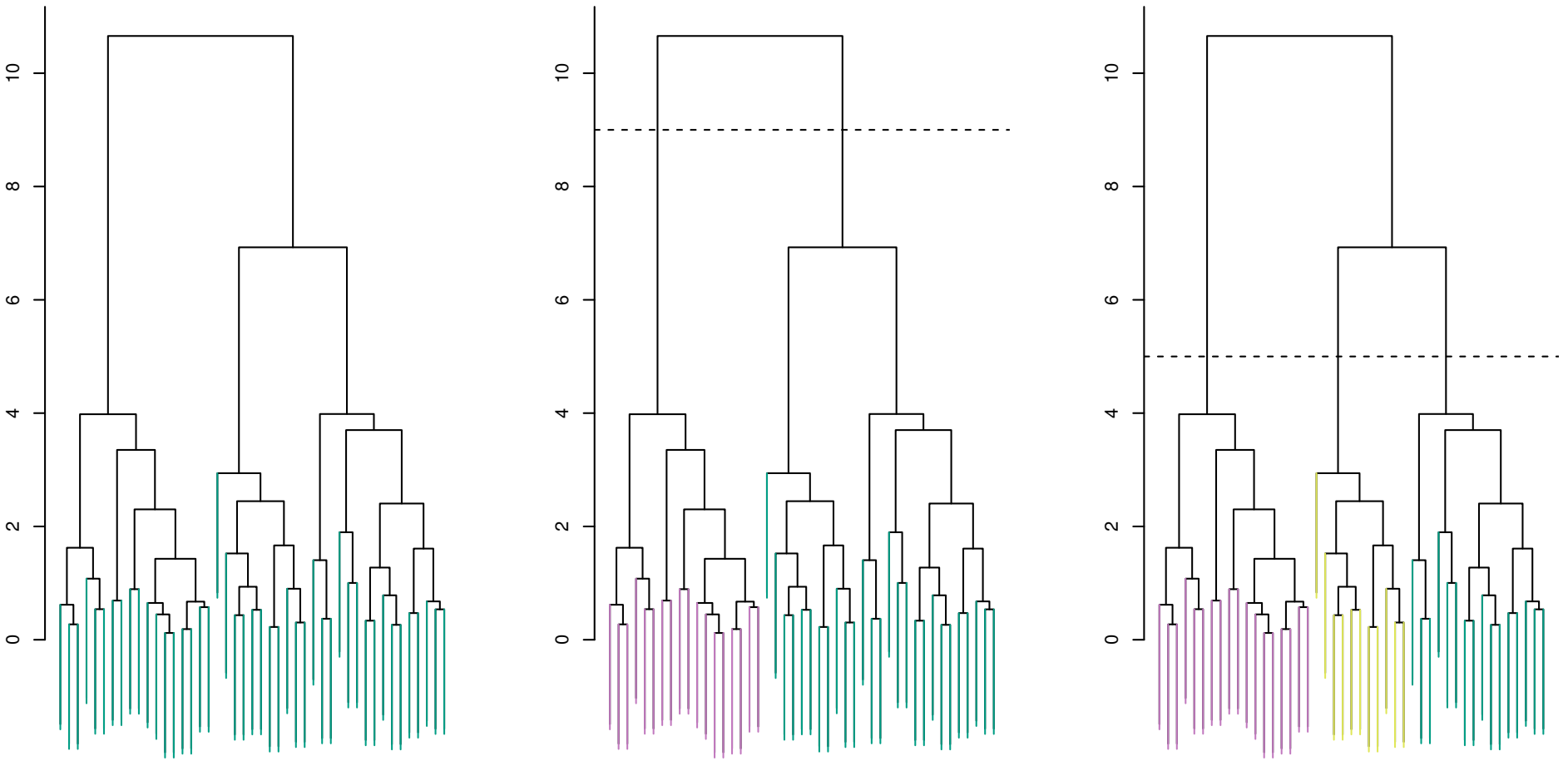


An Example



- 45 observations generated in 2-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.

Application of hierarchical clustering



Details of previous figure

- *Left*: Dendrogram obtained from hierarchically clustering the data from previous slide, with "complete linkage" and Euclidean distance.
- *Center*: The dendrogram from the left-hand panel, cut at a height of 9 (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 5. This cut results in three distinct clusters, shown in different colors.

Common Linkages

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.

Choice of Dissimilarity Measure

- So far we have used Euclidean distance.
- An alternative is correlation-based distance which considers two observations to be similar if their features are highly correlated.
- This is an unusual use of correlation, which is normally computed between variables; here it is computed between the observation profiles for each pair of observations.

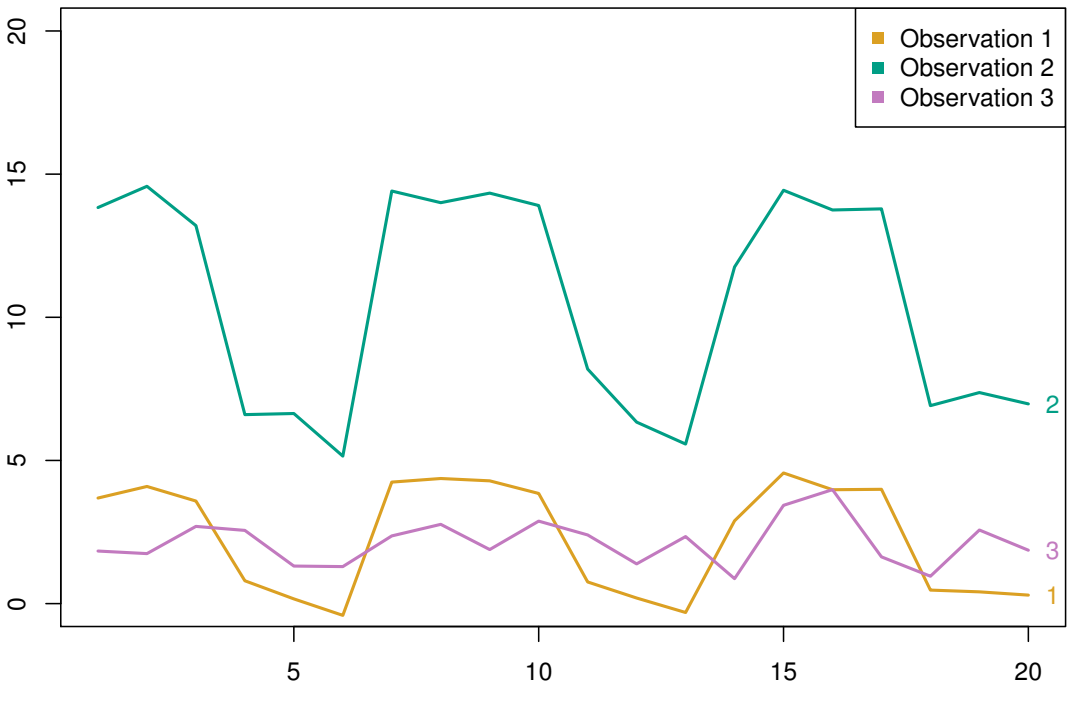


Figure: Observation profiles

Practical issues in 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?
- How many clusters to choose? Difficult problem. No agreed-upon method. See Elements of Statistical Learning, section 14.3.11 for more details.