Unsupervised Learning
K-means clustering

Unsupervised learning introduction

Unsupervised vs **Supervised** Learning:

- Most of this course focuses on **supervised learning** methods such as **regression** and **classification**.
- In that setting we observe both a **set of features** $x_1, x_2, ..., x_p$ for each object, as well as a response or outcome variable Y. The goal is then to **predict Y** using $x_1, x_2, ..., x_p$.
- Here we instead focus on unsupervised learning, we where observe **only the features** $x_1, x_2, \ldots, x_p$. We are not interested in prediction, because we **do not have** an associated response variable **Y**.

The Challenge of Unsupervised Learning

Unsupervised learning is **more subjective** than supervised learning, as there is no simple goal for the analysis, such as prediction of a response. But techniques for unsupervised learning are of growing importance in a number of fields:

- Subgroups 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.

Another advantage

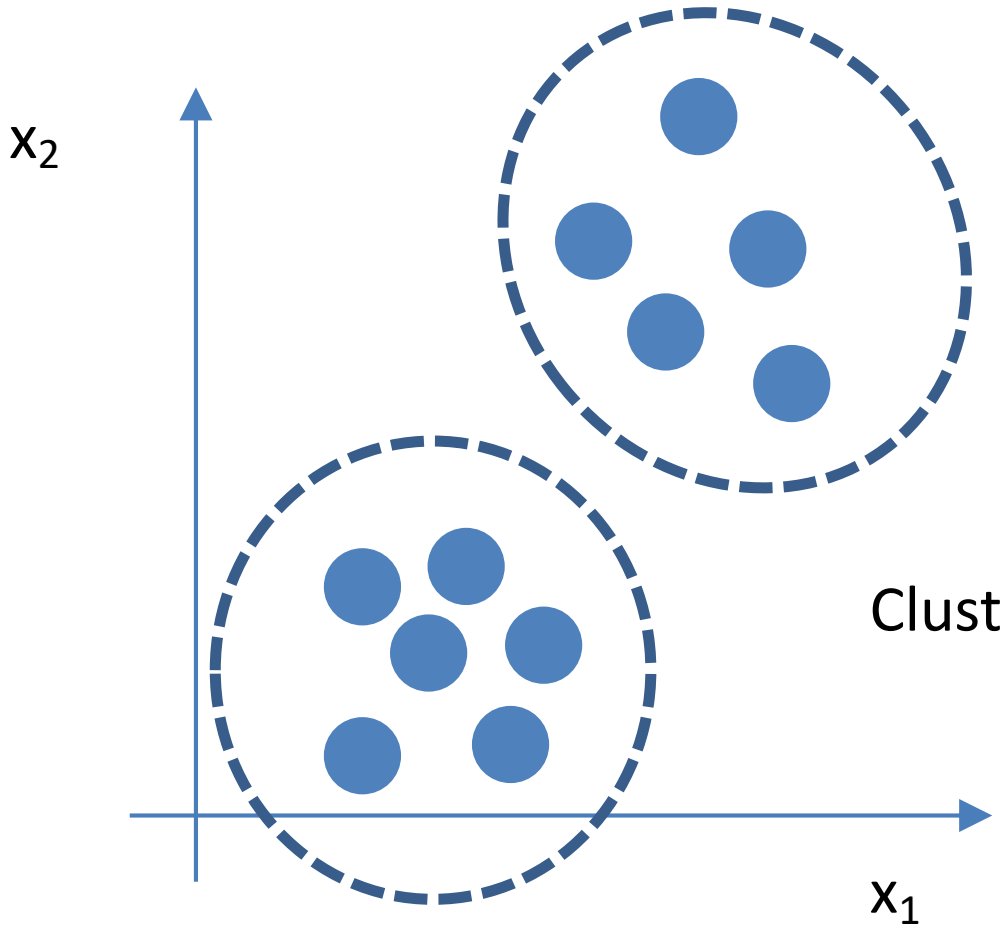
It is often **easier to obtain unlabeled data** from a lab instrument or a computer than labeled data, which can require human intervention.

For example it is difficult to automatically assess the overall sentiment of a movie review: is it favorable or not?

Clustering

- **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.
- It 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

Unsupervised learning - Clustering



Clustering algorithm

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

Clustering for 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.

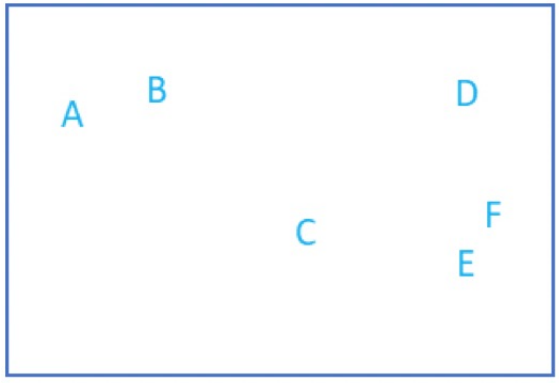
- Our goal is to perform **market segmentation** by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.

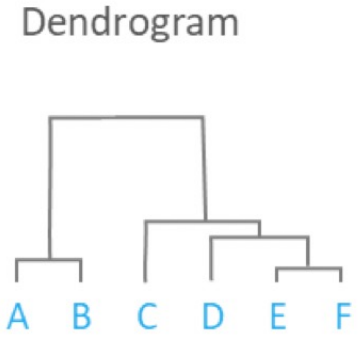
- The task of performing market segmentation amounts to clustering the people in the data set.

Two clustering methods

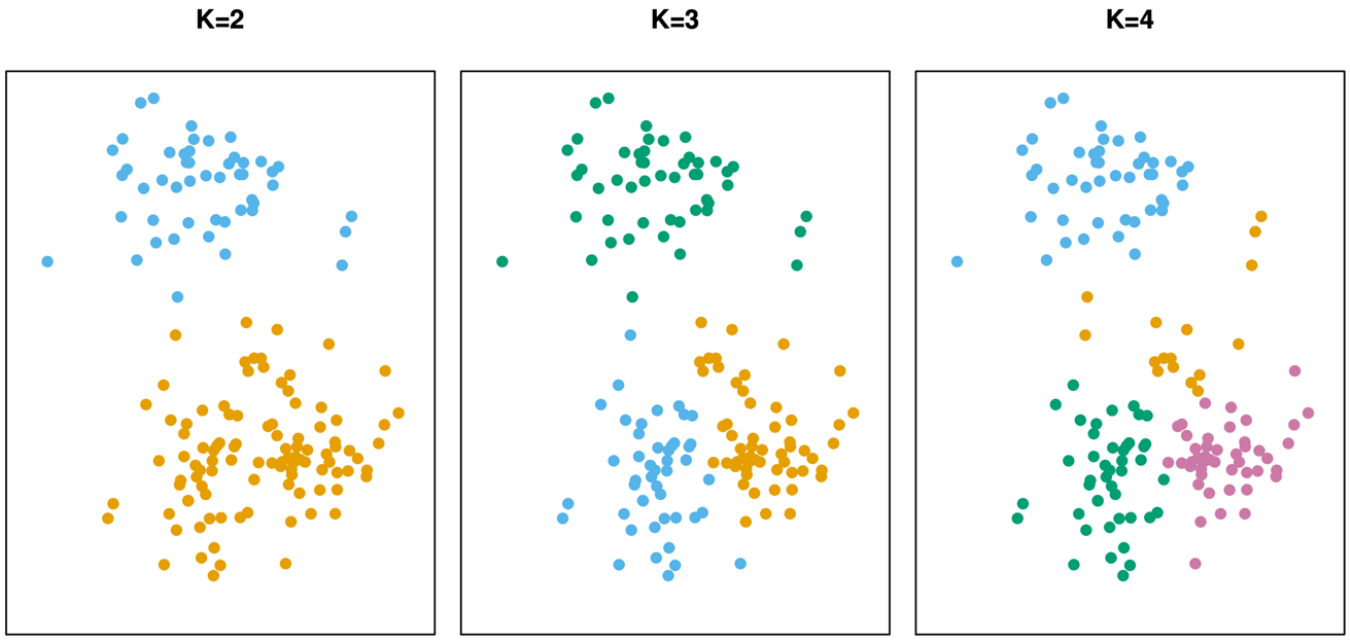
- In **K-means clustering**, we seek to partition the observations into a **pre-specified number of clusters**.

- In **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 clusters 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. 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**.

K-means Clustering

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

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.
2. $C_k \cap C_{k'} = \emptyset$ 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 $i \in C_k$.

K-means Clustering

- 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 $\mathrm{WCV}(C_k)$ of the amount by which the observations within a cluster differ from each other.
- Hence we want to solve the problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \mathrm{WCV}(C_k) \right\}. \tag{2}$$

- 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

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

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,

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

How to define within-cluster variation?

each observation ***i*** has ***p*** features $x_{i1},\ldots,x_{ip}$

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

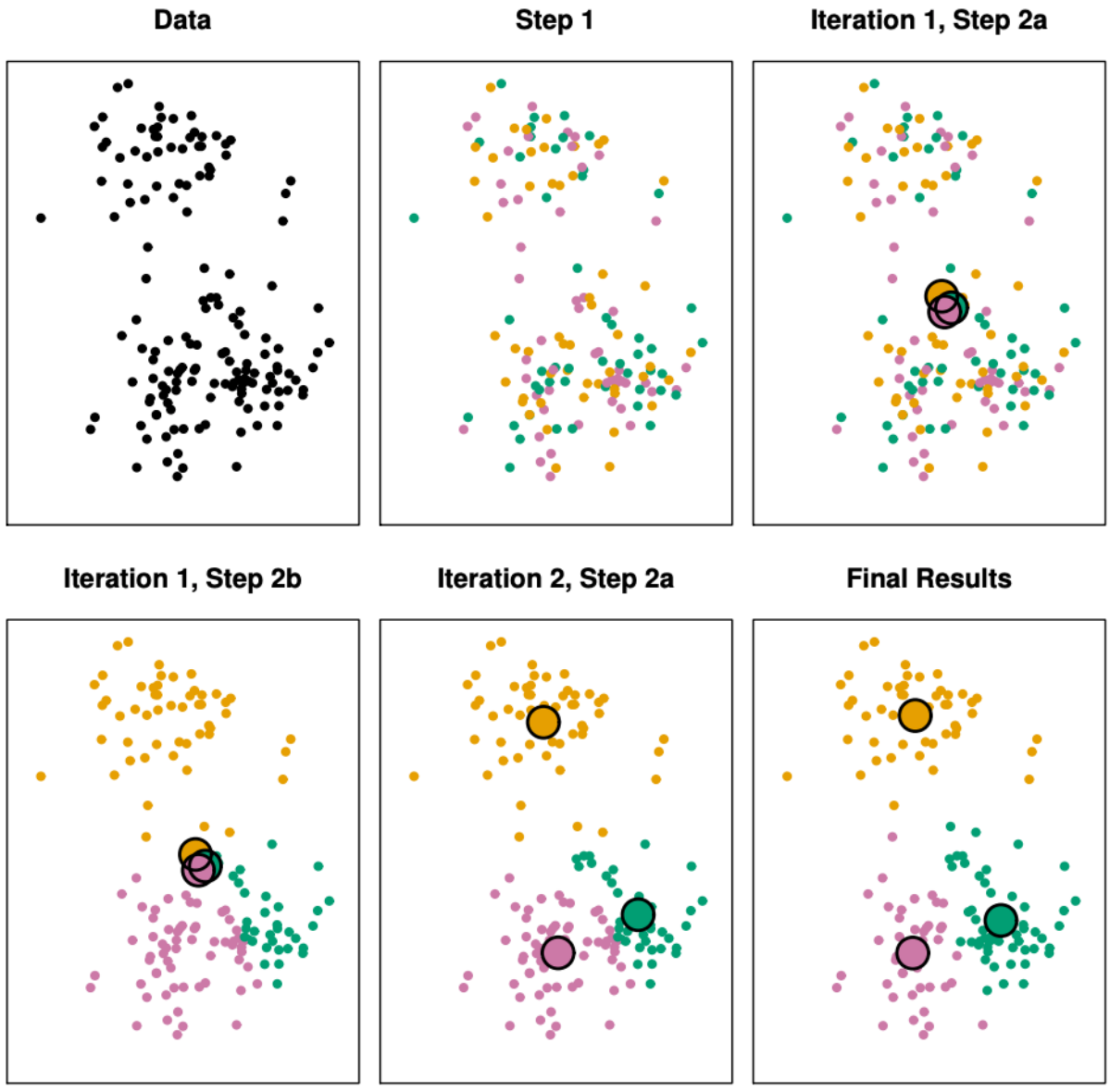
within-cluster-variation of cluster C_k

minimize within-cluster-variation of all ***K*** clusters

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:
   - 2.1 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.
   - 2.2 Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Example

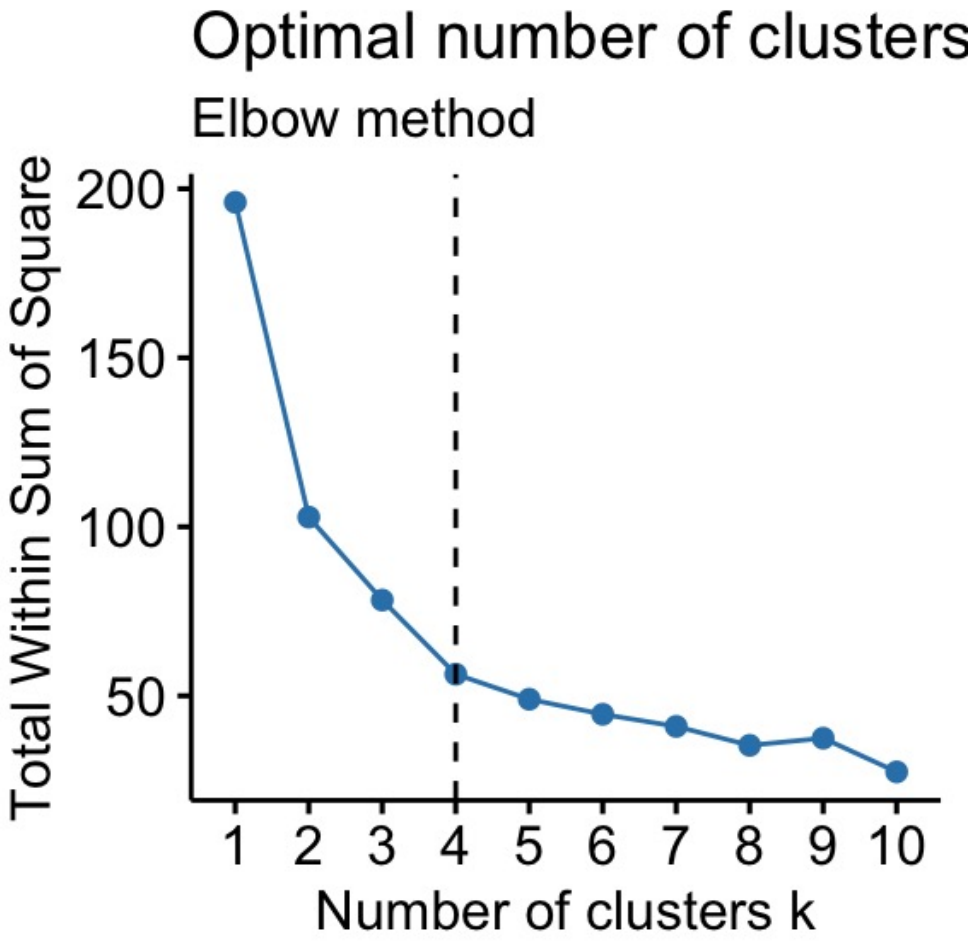

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Example: different starting values

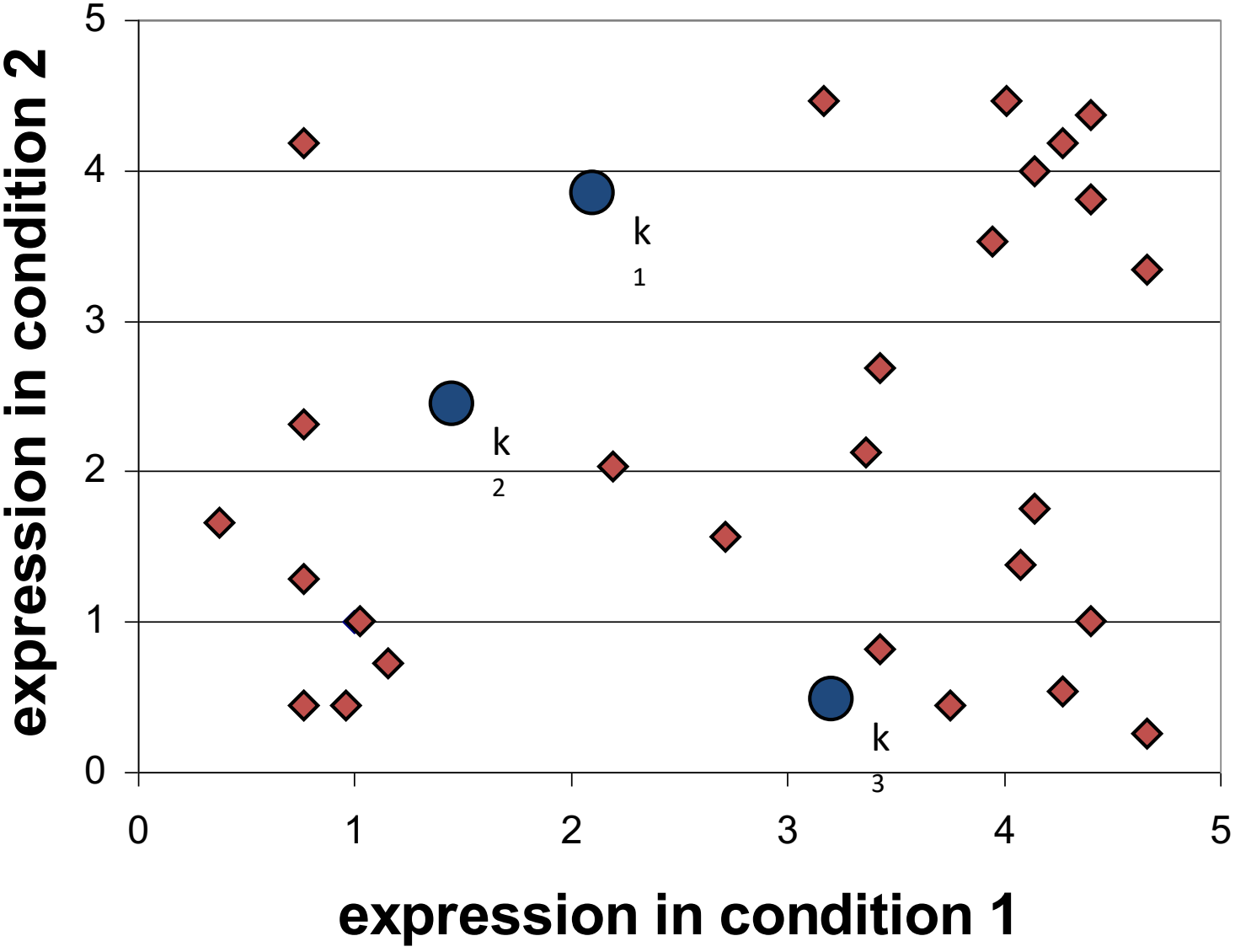
Choosing Number of Clusters K

- Compute clustering algorithm (e.g., k-means clustering) for **different values of K**. For instance, by varying **K** from 1 to 10 clusters.
- For **each K**, calculate the total within-cluster-variance (WCV).
- **Plot the curve** of WCV according to the number of clusters **K**.
- The location of a **bend (knee)** in the plot is generally considered as an indicator of the appropriate number of clusters.

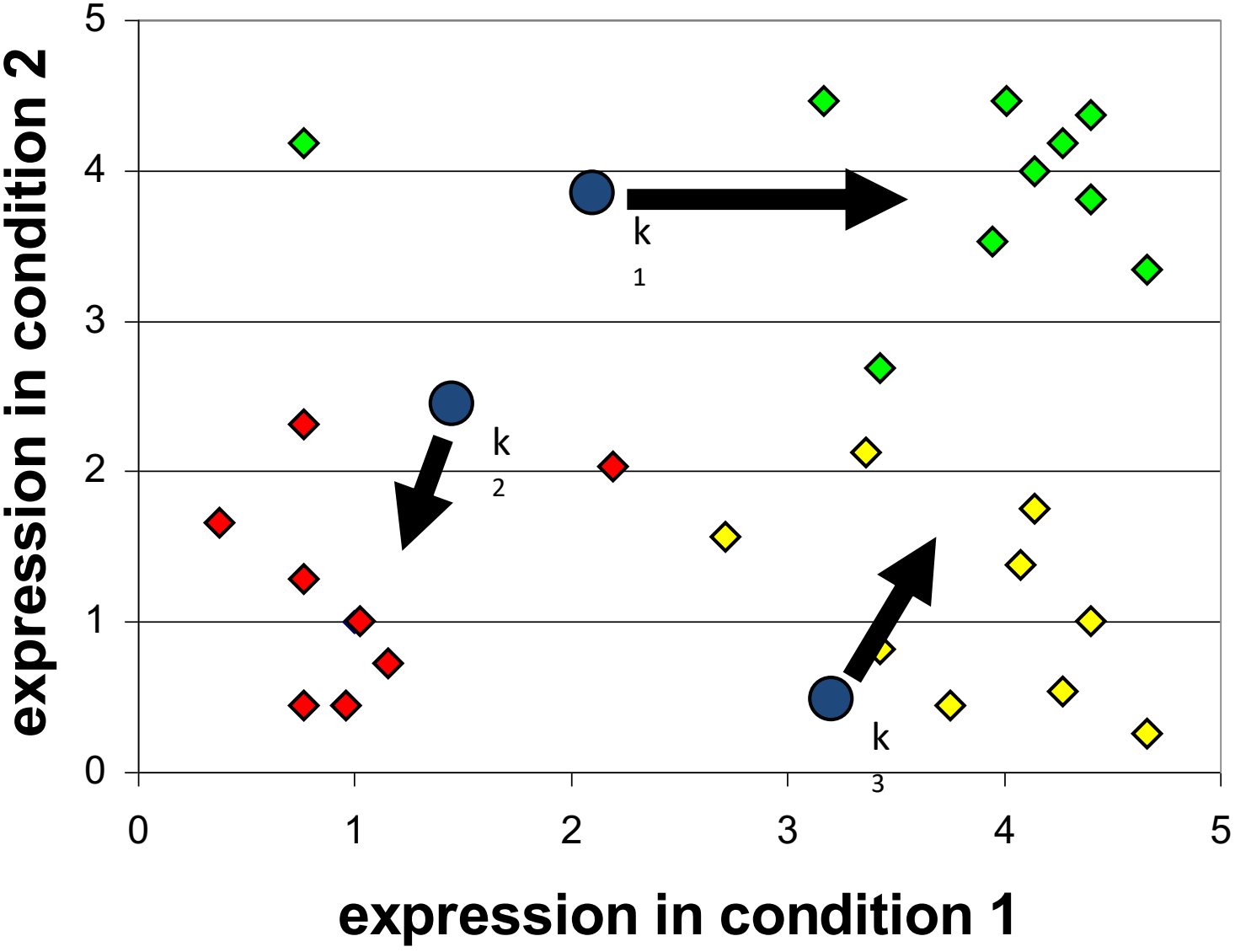
Choosing Number of Clusters K



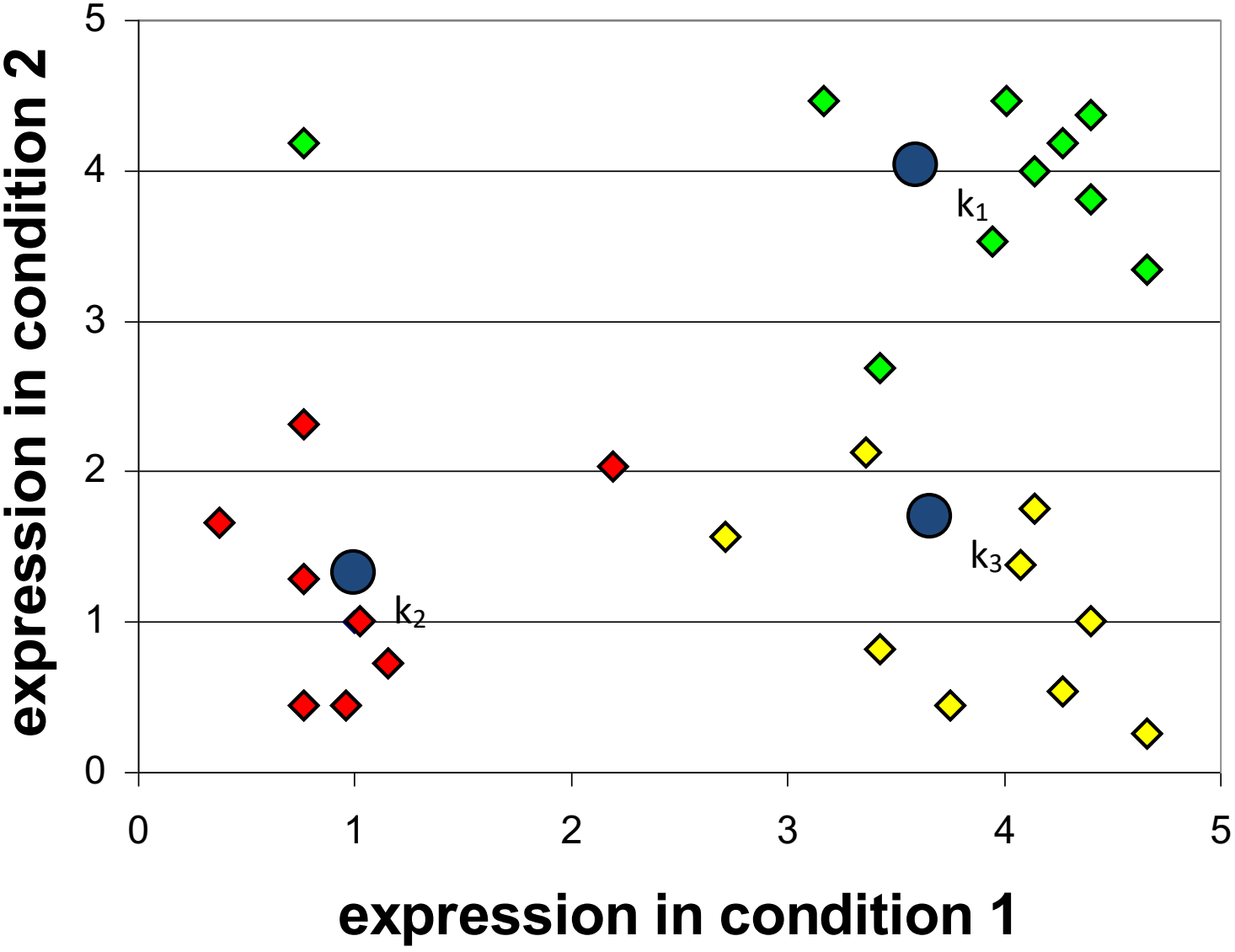
K-means iteration (1)



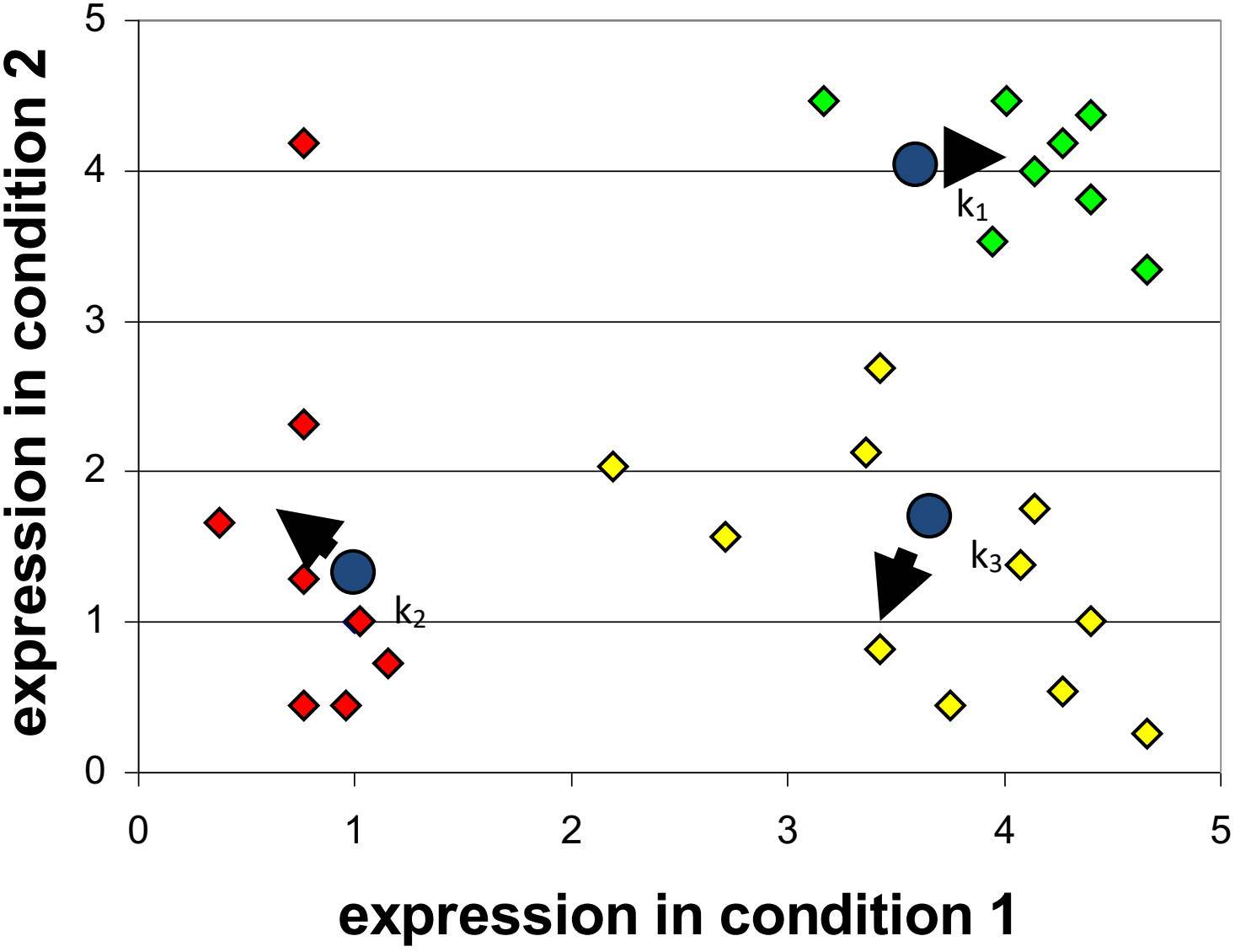
K-means iteration (2)



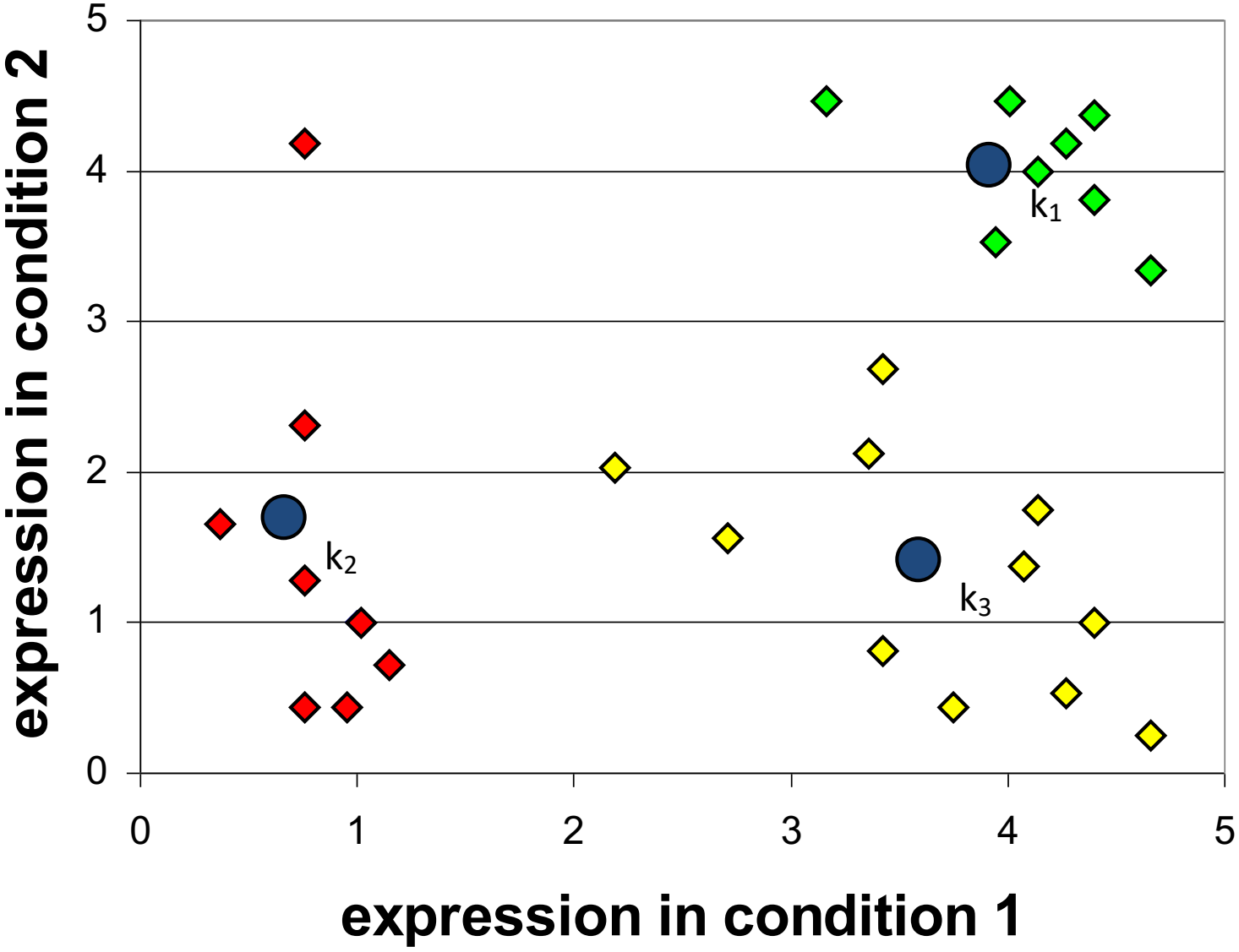
K-means iteration (3)



K-means iteration (4)



K-means iteration (5)



Hierarchical Clustering

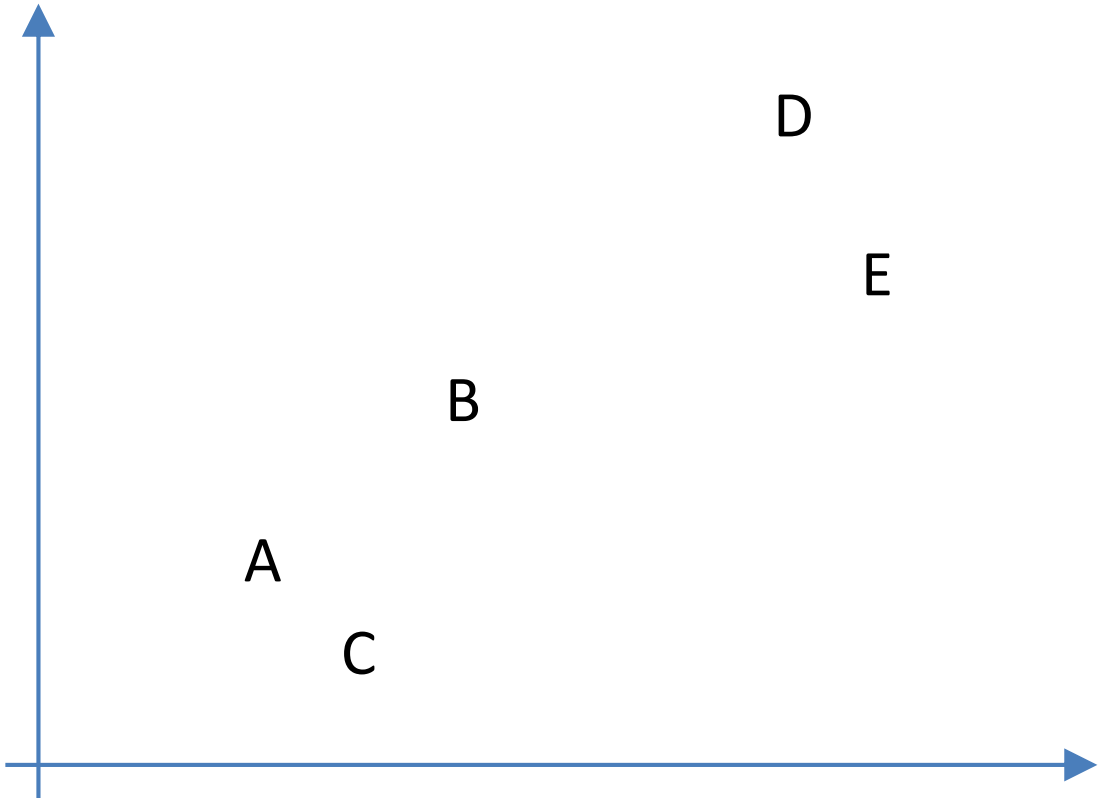
- K-means clustering requires us to pre-specify the number of clusters K. This can be a disadvantage.

- **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.

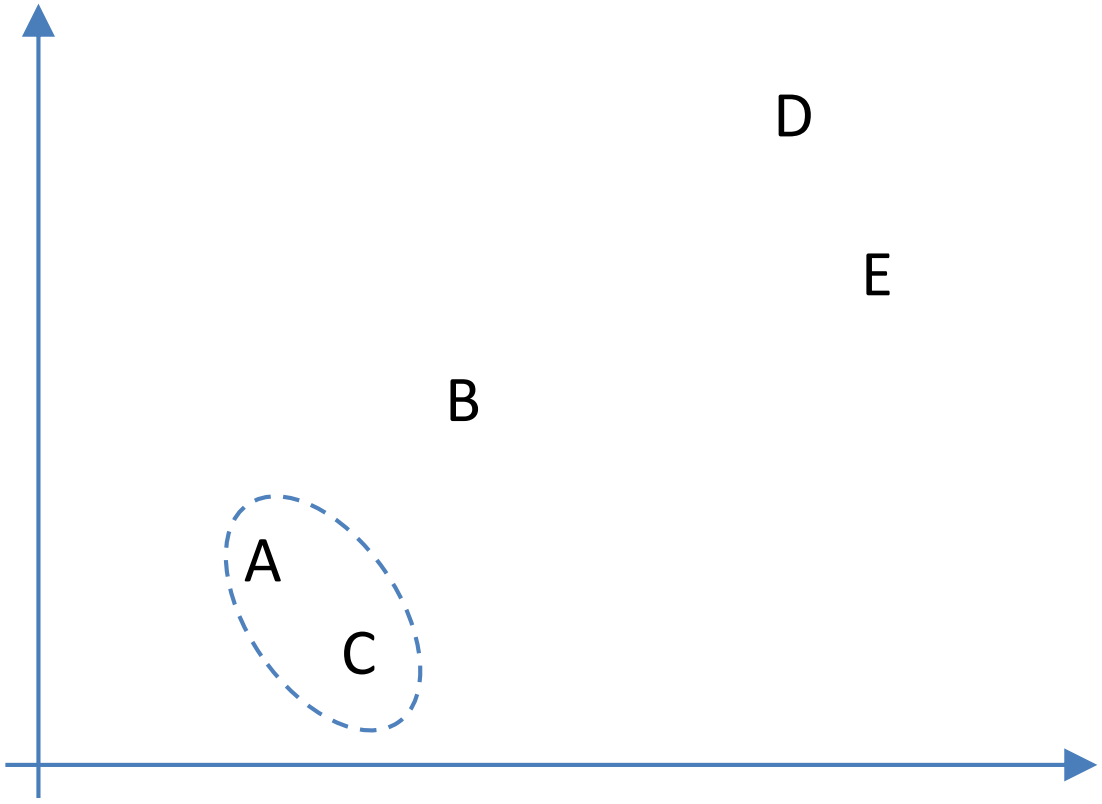
Hierarchical Clustering

Builds a hierarchy in a **bottom-up** fashion



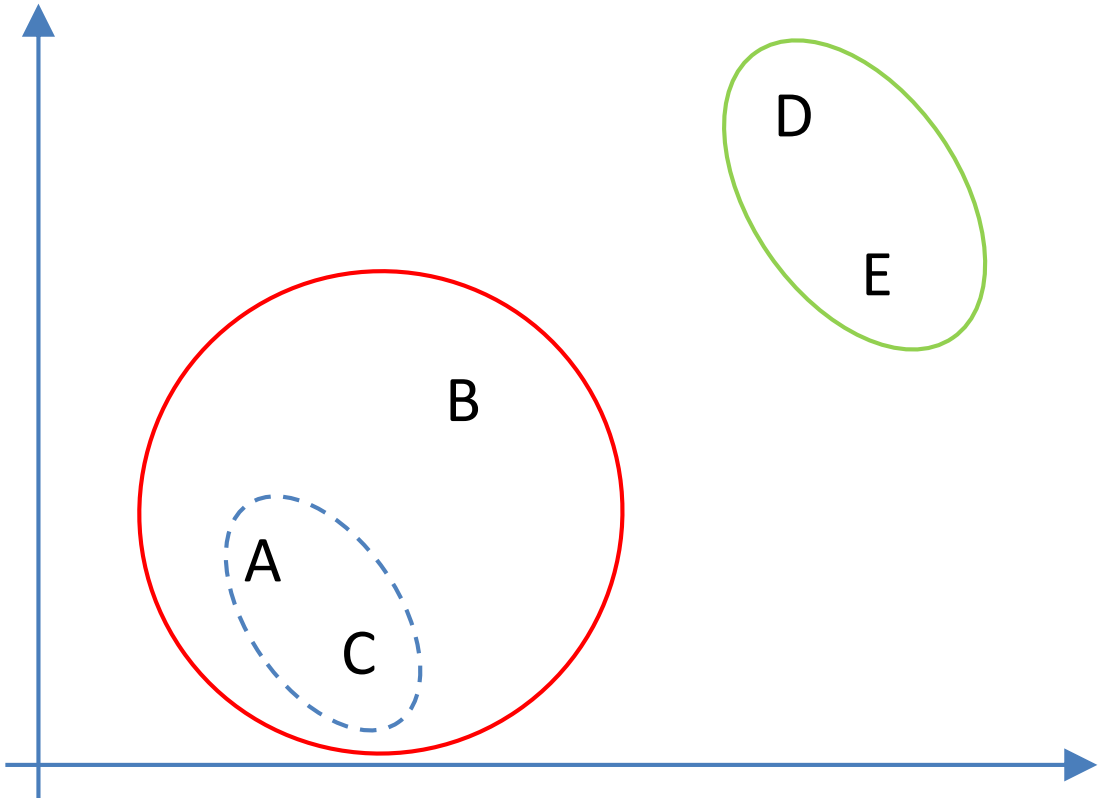
Hierarchical Clustering

Builds a hierarchy in a **bottom-up** fashion



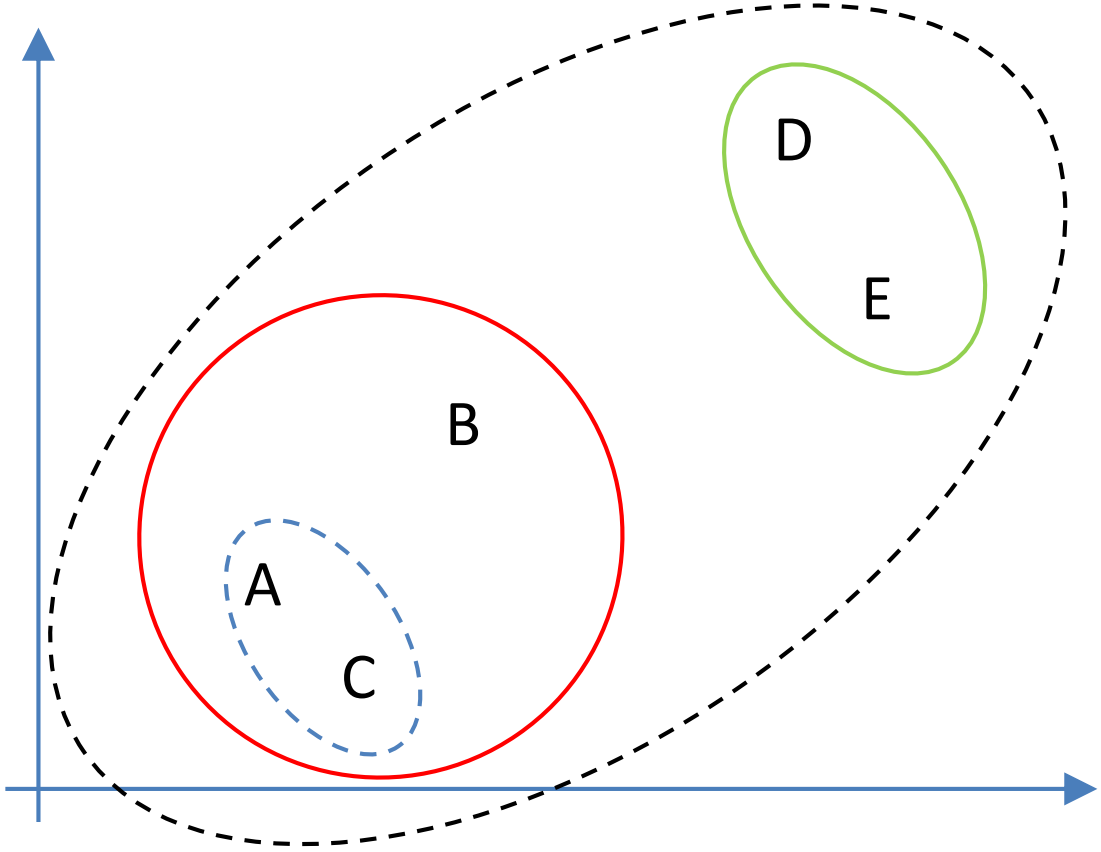
Hierarchical Clustering

Builds a hierarchy in a **bottom-up** fashion



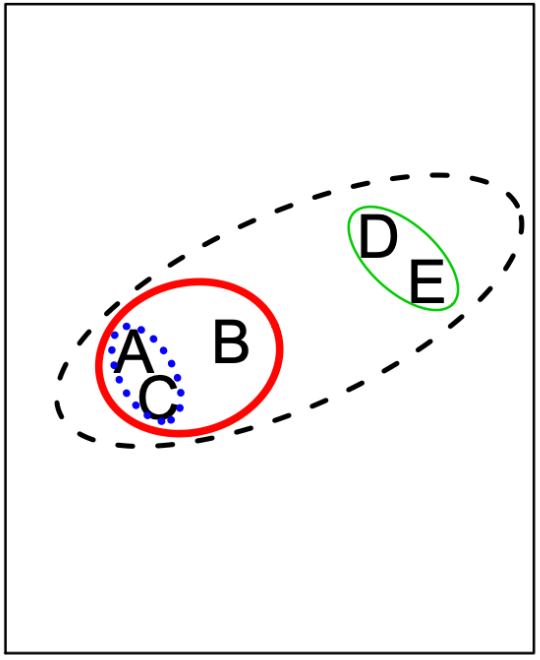
Hierarchical Clustering

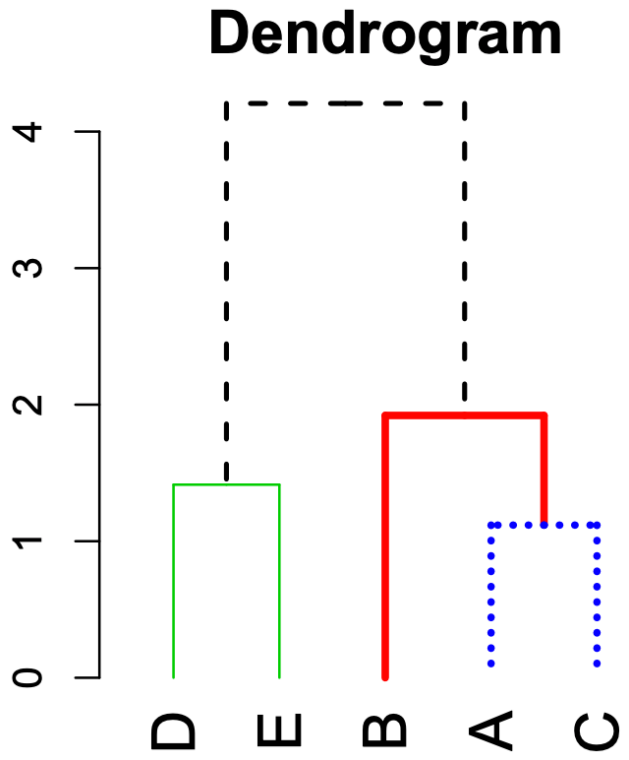
Builds a hierarchy in a **bottom-up** fashion



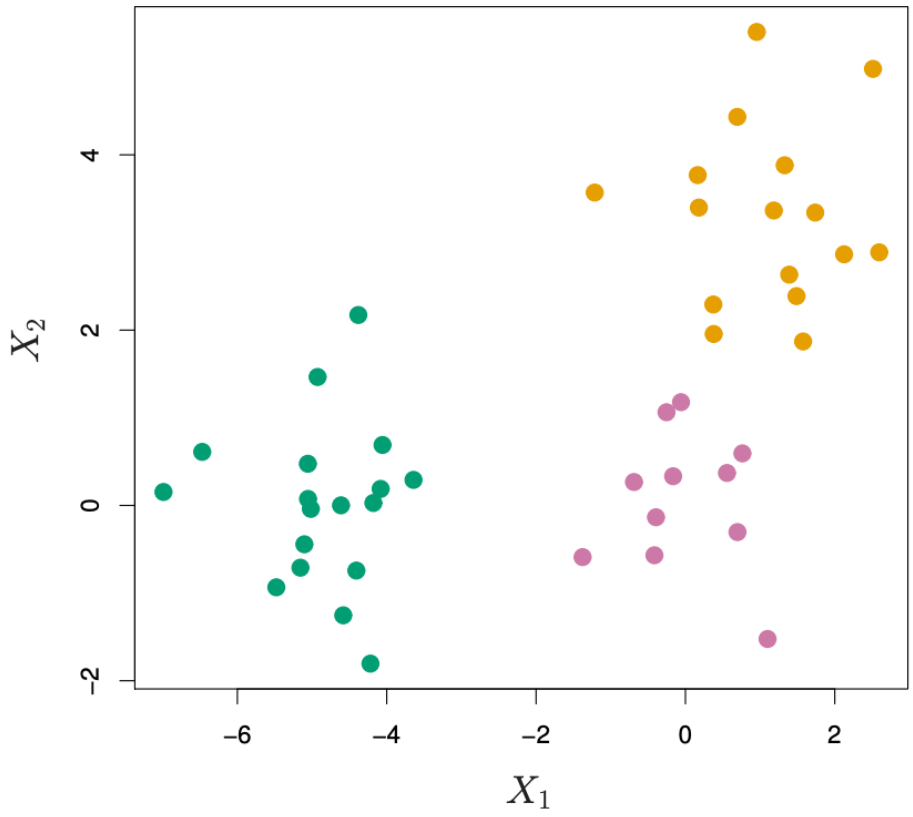
Hierarchical Clustering Algorithm

- Start with **each point in its own cluster** *(leave nodes)*.
- Identify the **closest two clusters** and merge them.
- Repeat.
- End when all points are in a **single cluster** *(unique tree)*.



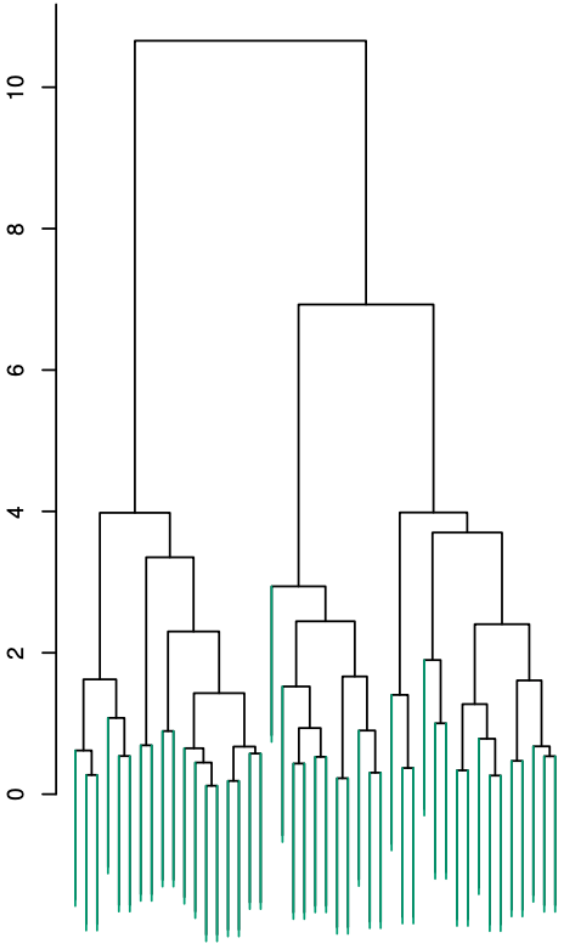


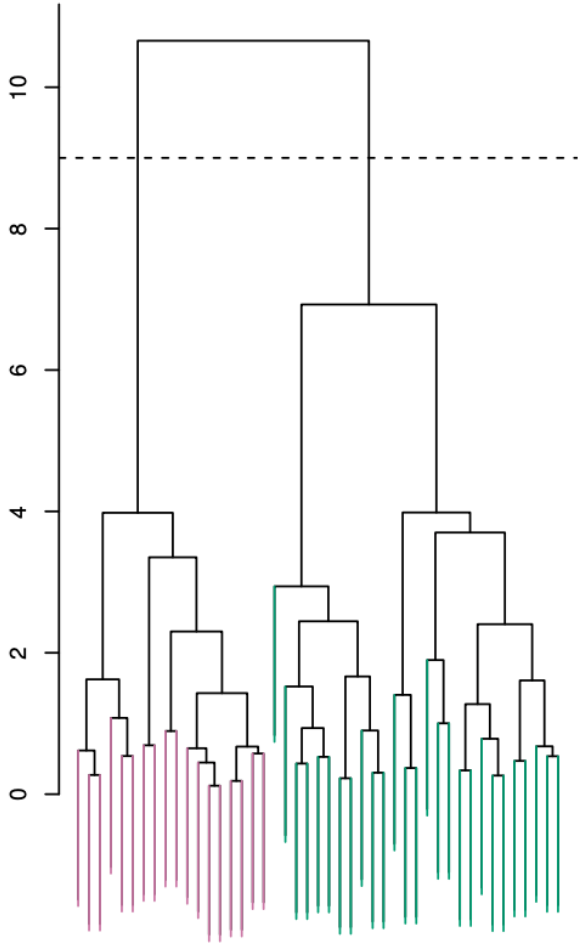
Example

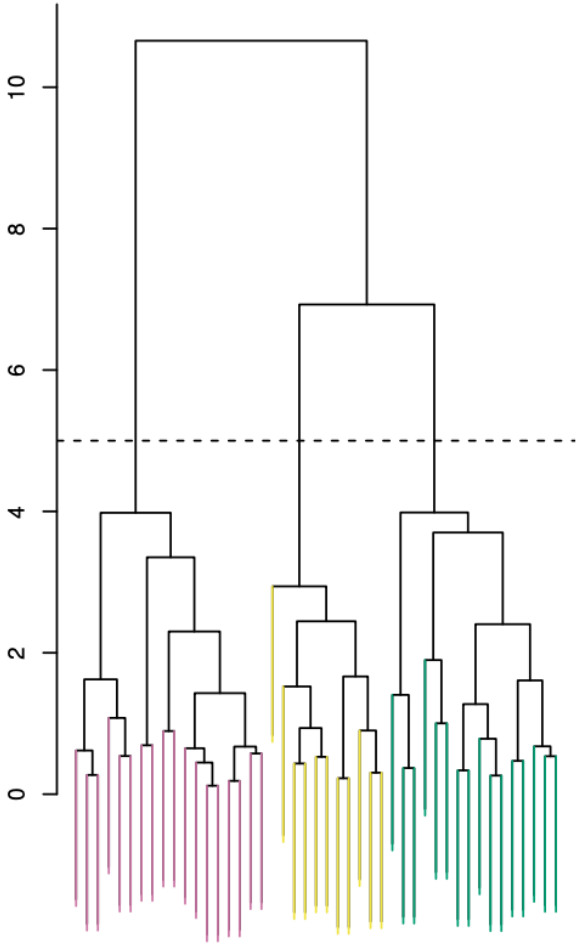


45 observations generated in 2-dimensional space. In reality there are **three distinct classes**, shown in separate colors. However, we will treat these **class labels as unknown** and will seek to cluster the observations in order to discover the classes from the data.

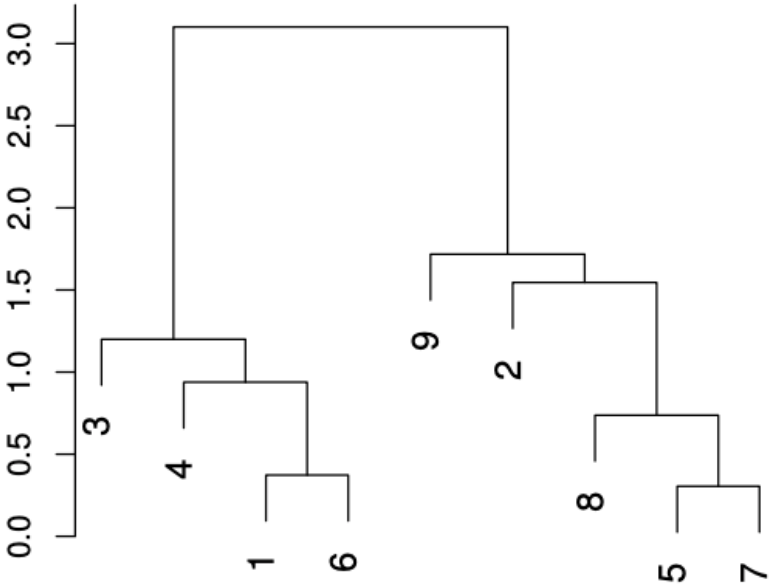
Example

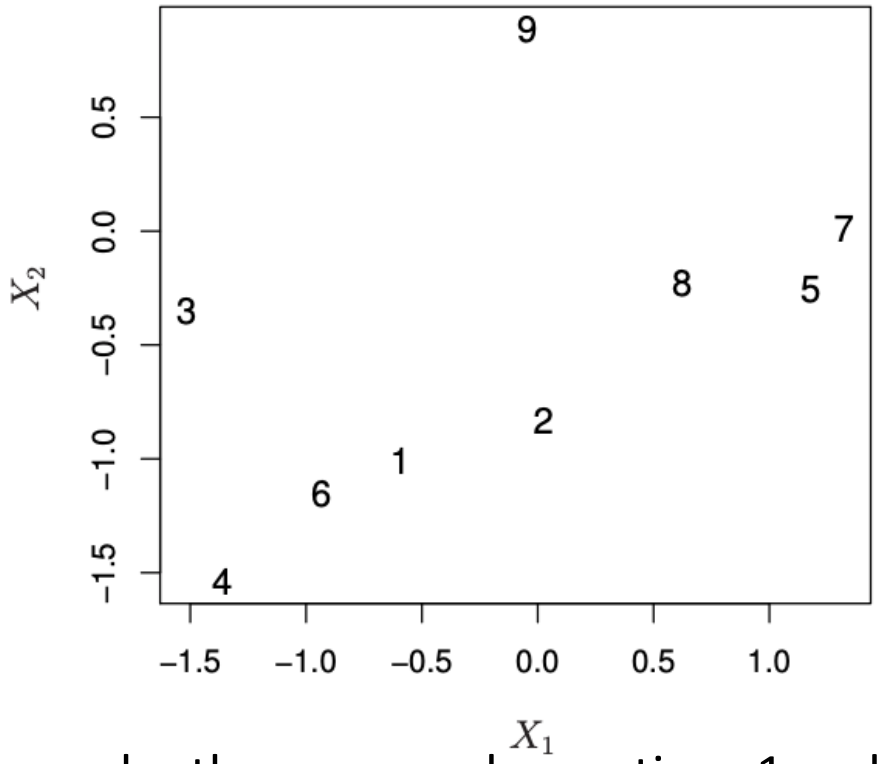






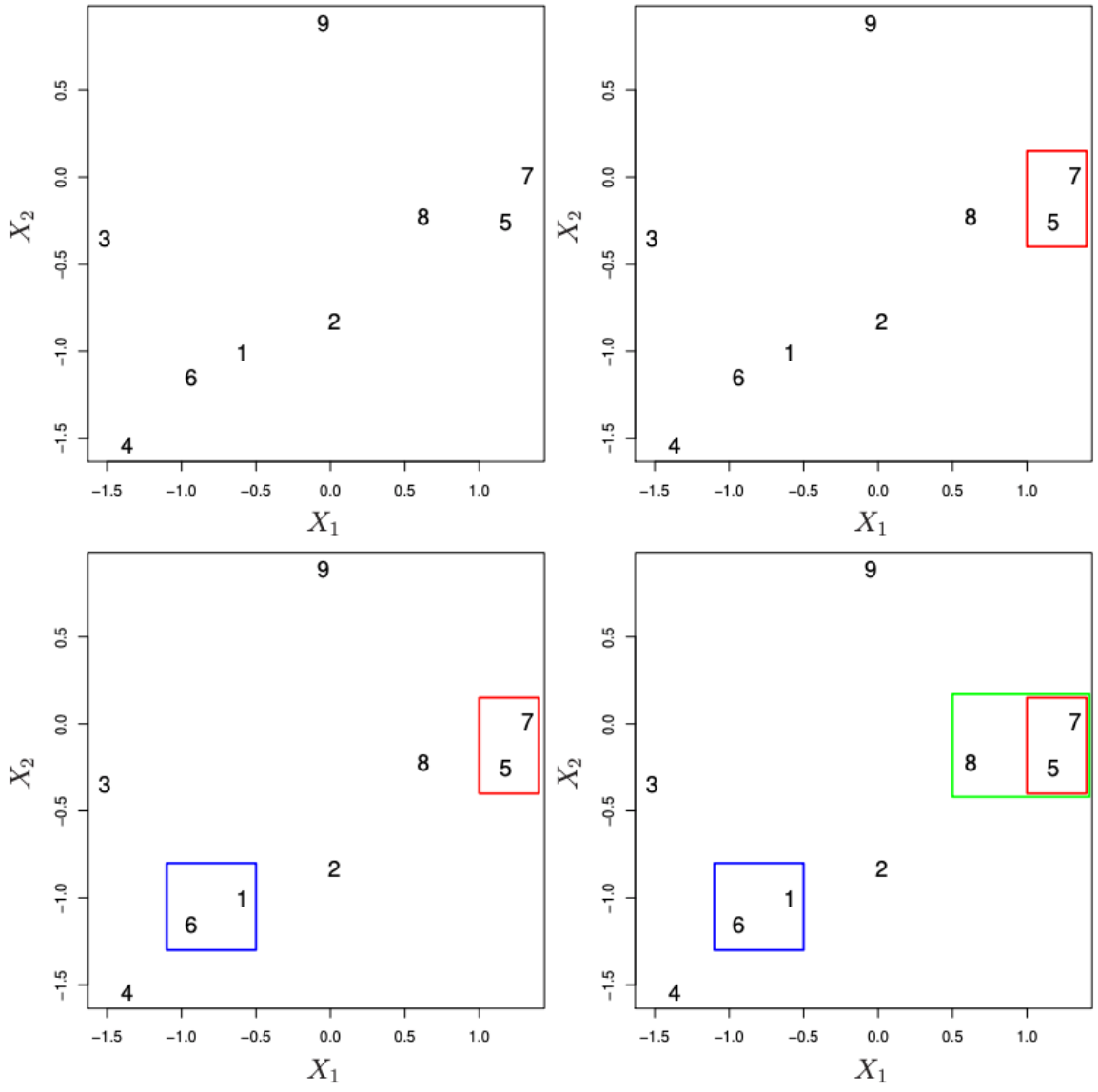
Another example





- 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.

Another example



Clustering Distance Measures

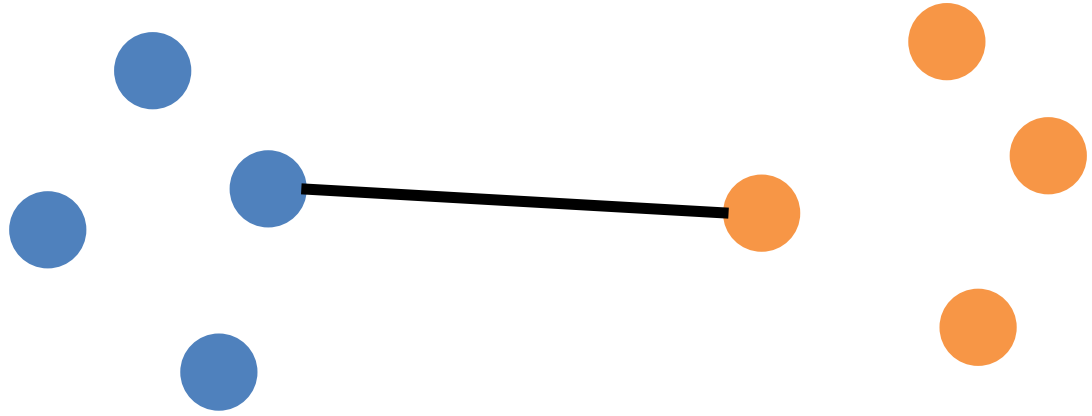
Type of linkage:

- Single link
- Complete link
- Average Link
- Centroids

Single link

Distance between closest elements in clusters

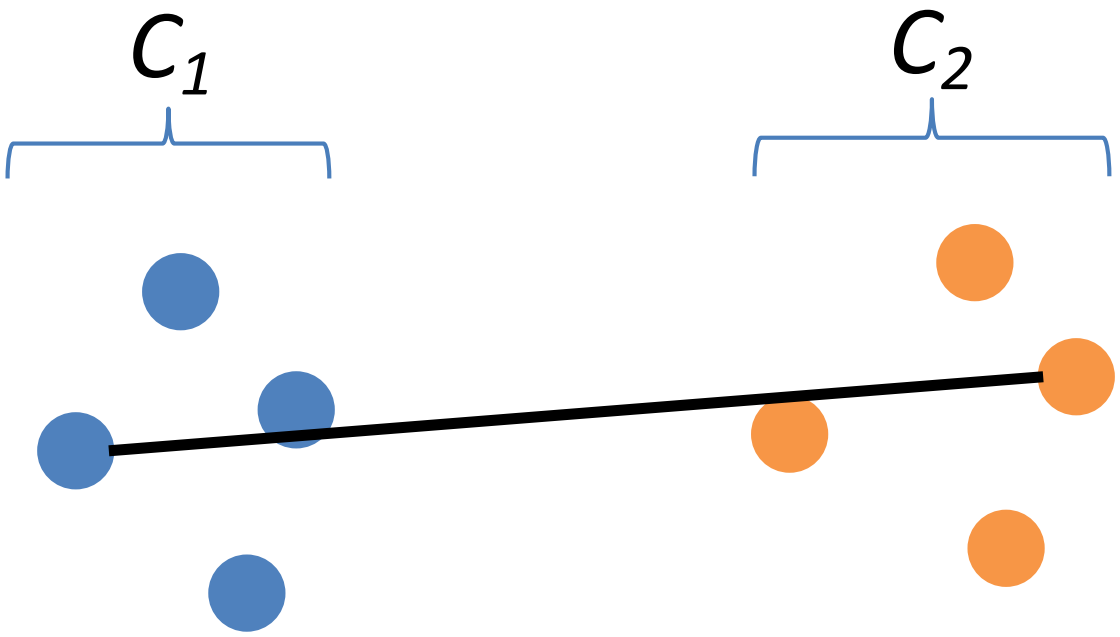
$$D(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} D(x_1, x_2)$$



Complete link

Distance between farthest elements in clusters

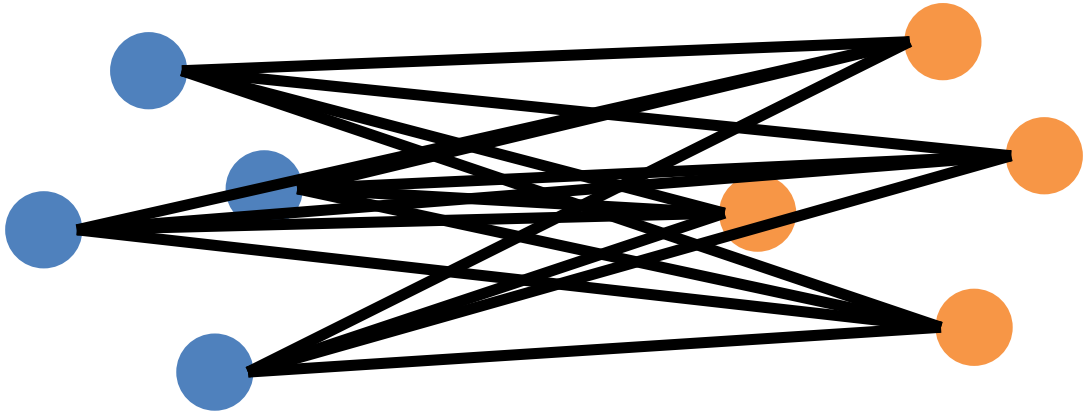
$$D(C_1, C_2) = \max_{x_1 \in C_1, x_2 \in C_2} D(x_1, x_2)$$



Average link

Average of all pairwise distances

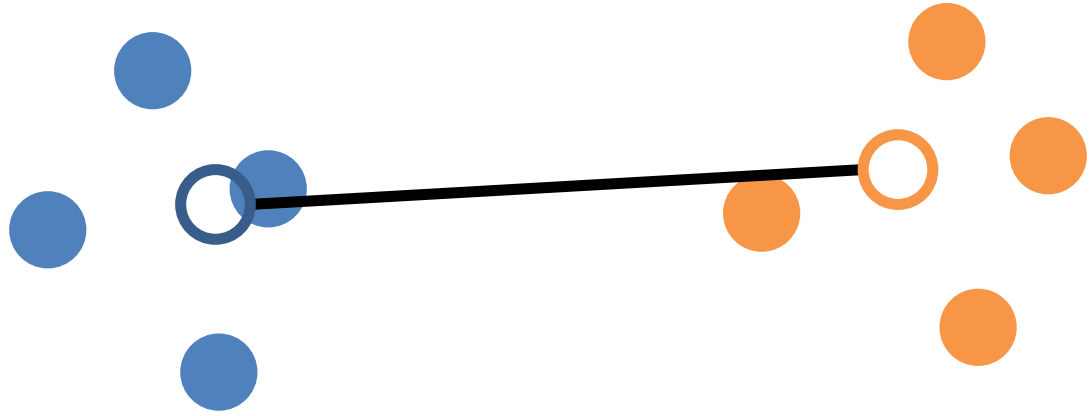
$$D(C_1, C_2) = \frac{1}{|C_1|}\frac{1}{|C_2|}\sum_{x_1 \in C_1}\sum_{x_2 \in C_2} D(x_1, x_2)$$



Centroid link

Distance between centroids of two clusters

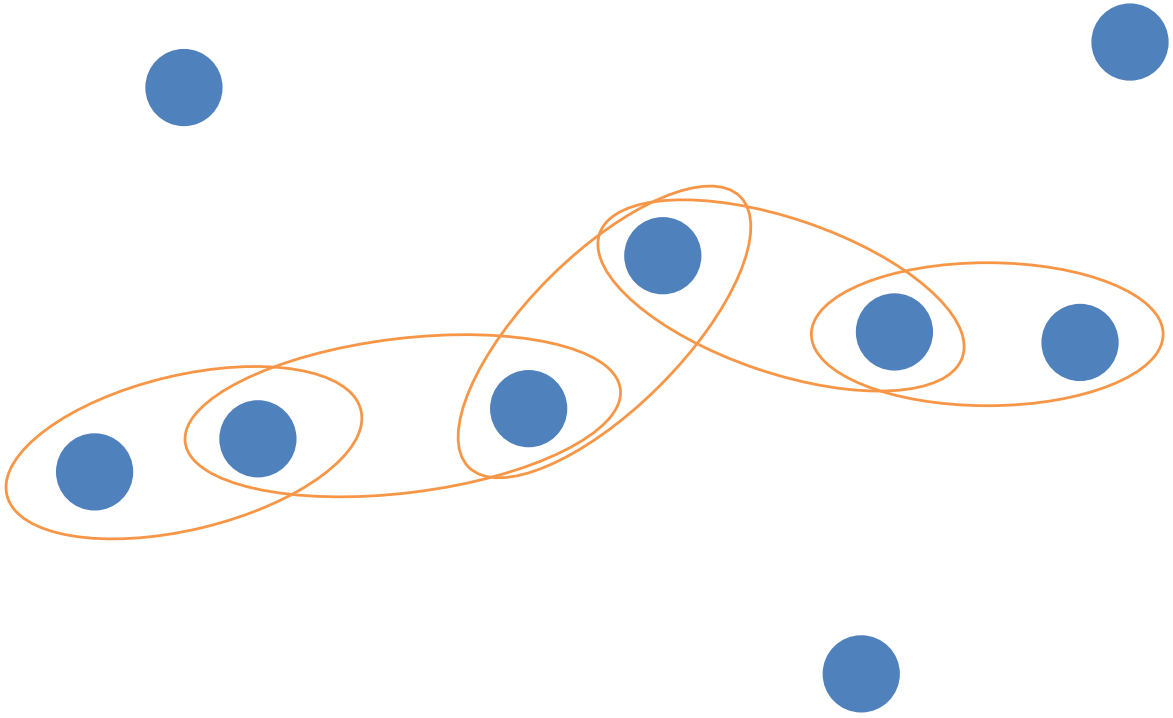
$$D(C_1, C_2) = D\left(\left(\frac{1}{|C_1|}\sum_{x_1 \in C_1} \overline{x}\right), \left(\frac{1}{|C_2|}\sum_{x_2 \in C_2} \overline{x}\right)\right)$$



Comparison

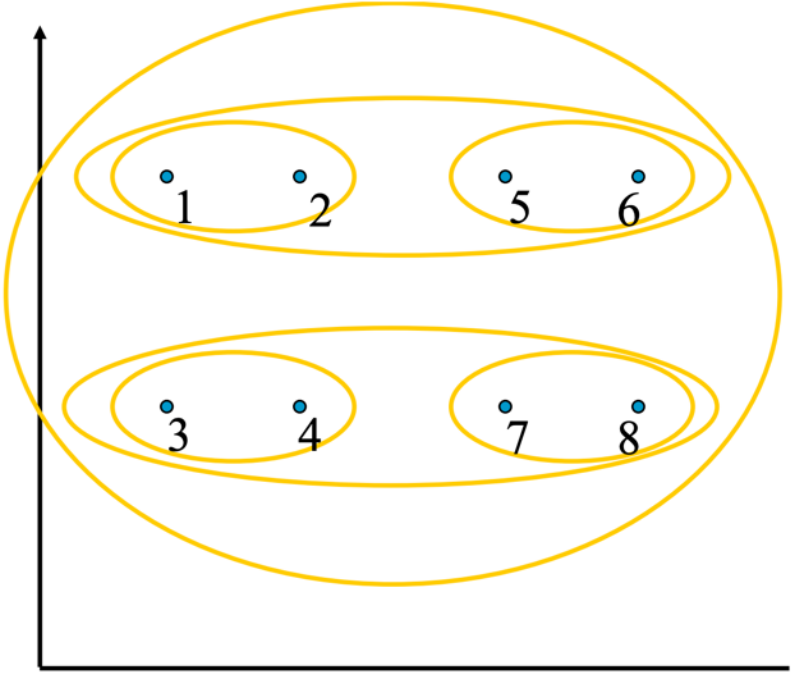
- **Complete link**: largest distance
- **Average link**: average distance
- **Single link**: smallest distance

Example: **Single link** can produce **long stretched** clusters (A→B→C→D…)

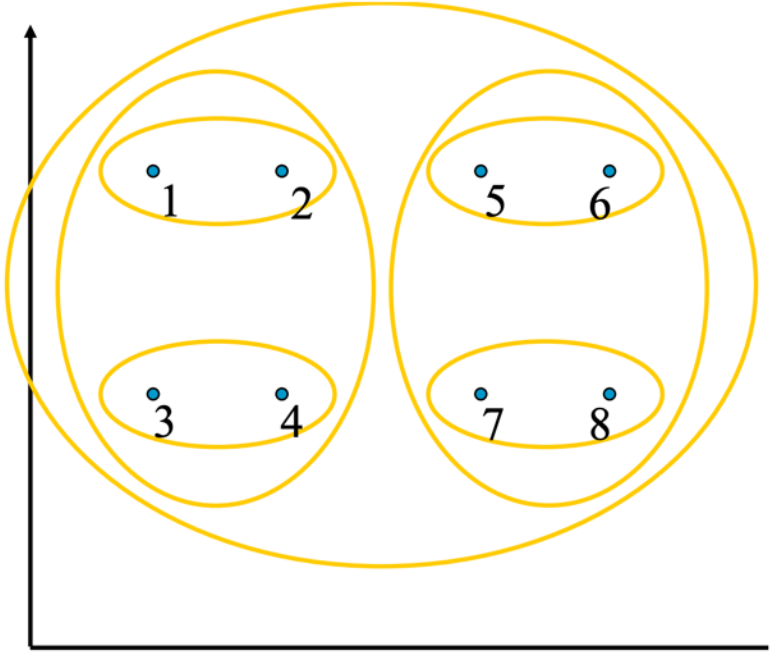


Clustering behavior

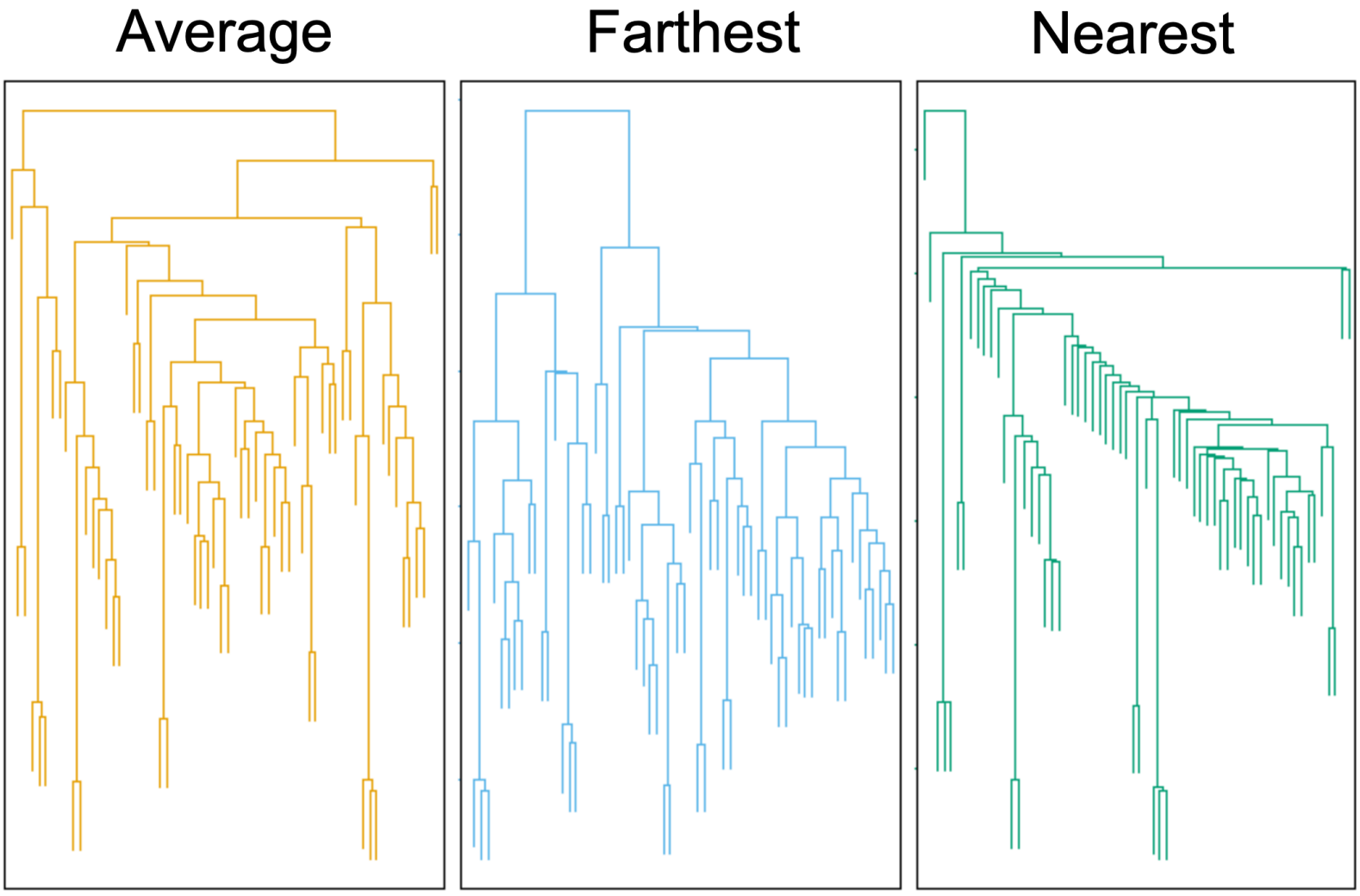
Closest pair
(single-link clustering)



Farthest pair
(complete-link clustering)



Clustering behavior



Mouse tumor data from [Hastie *et al.*]

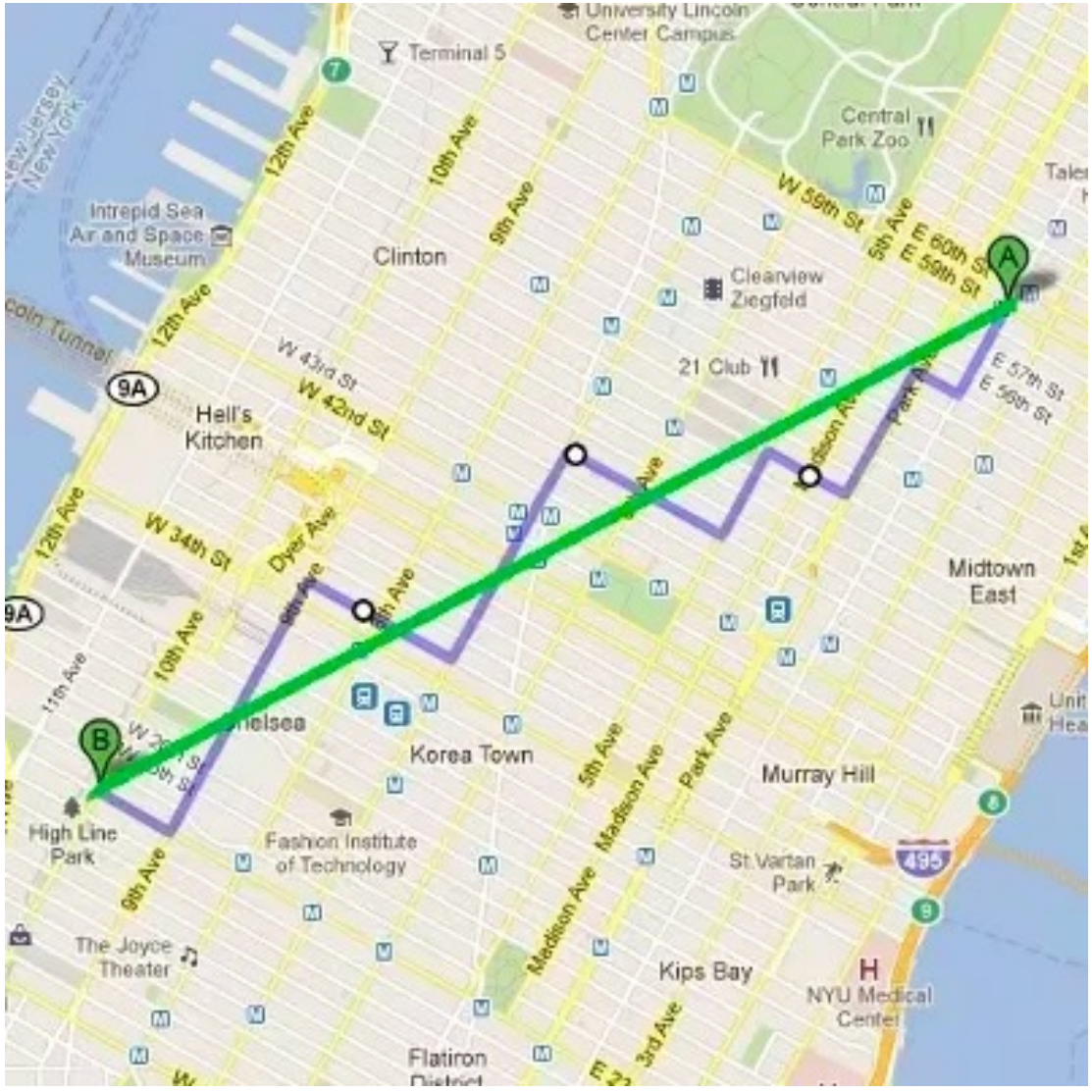
Distance Measures

Given vectors $x = (x_1, \ldots xn), y = (y_1, \ldots, yn)$

Euclidean distance: $D_E(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - yi)^2}$

Manhattan distance: $D_M(x, y) = \sum_{i=1}^{n} |xi - yi|$

Euclidean vs. Manhattan distance



Summary

Unsupervised learning

Clustering

- K-means clustering
  - Within-cluster-variation
- Hierarchical clustering
  - Single, complete, average, centroid link
  - Distance measures
    - Euclidean, Manhattan

Q&A

Thank you