CSE 375 Machine Learning and Pattern Recognition

13. Clustering

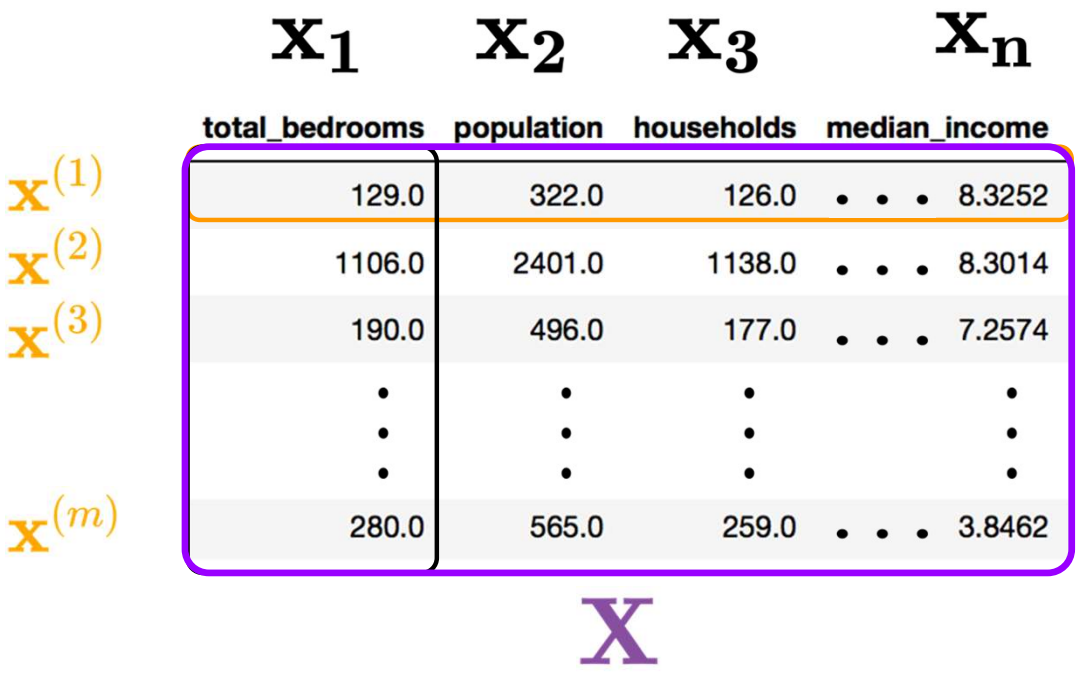
Slides Credit: N. Rich Nguyen

Contents

1. Discuss Unsupervised Learning methods using the Simpsons!
2. Represent “group” with similarity distance
3. Learn 2 clustering algorithms: partitional and hierarchical
4. Understand how k-means algorithm works

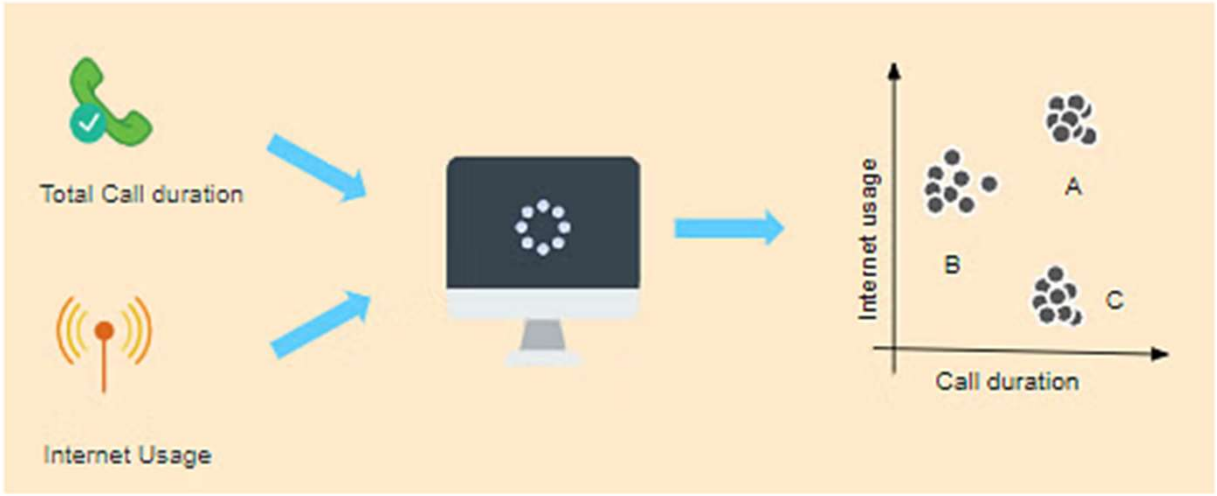
Unsupervised Learning

Unlabeled Dataset



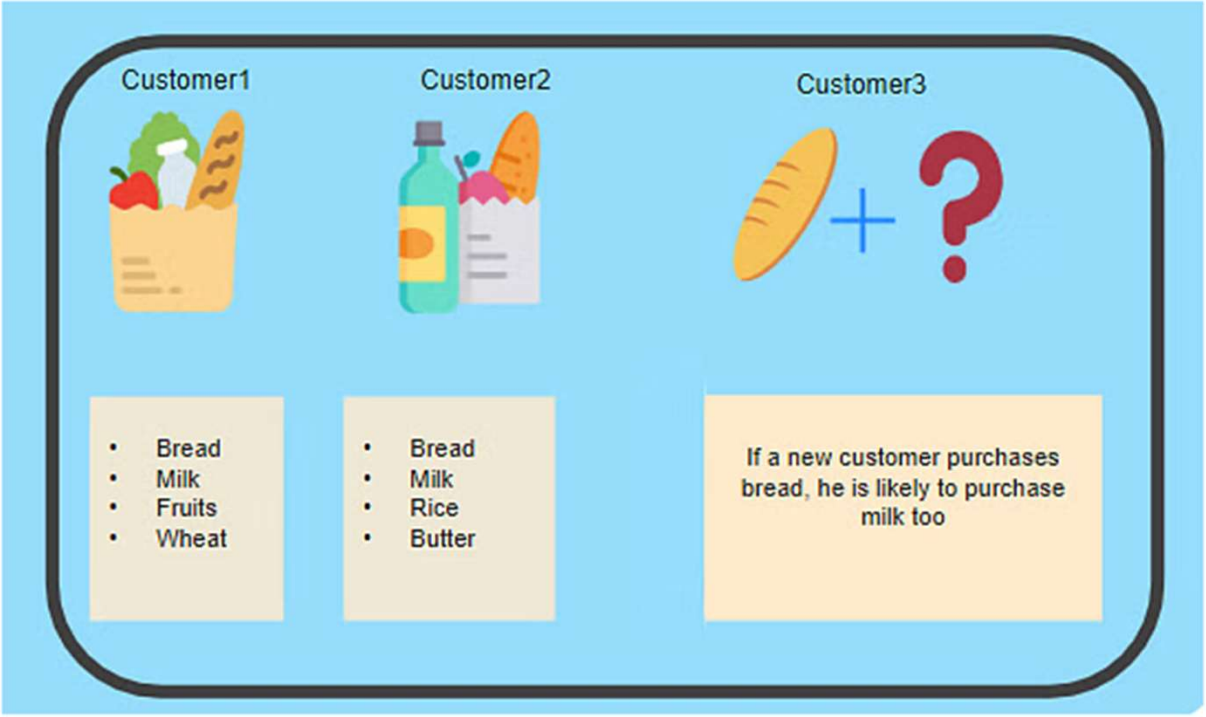
No y, no label, no annotation is given!

Identify Groups of Customers



https://www.simplilearn.com/tutorials/machine-learning-tutorial/supervised-and-unsupervised-learning#:~:text=Clustering%20%2D%20Unsupervised%20Learning&text=For%20example%2C%20finding%20out%20which,the%20customers%20with%20similar%20traits.

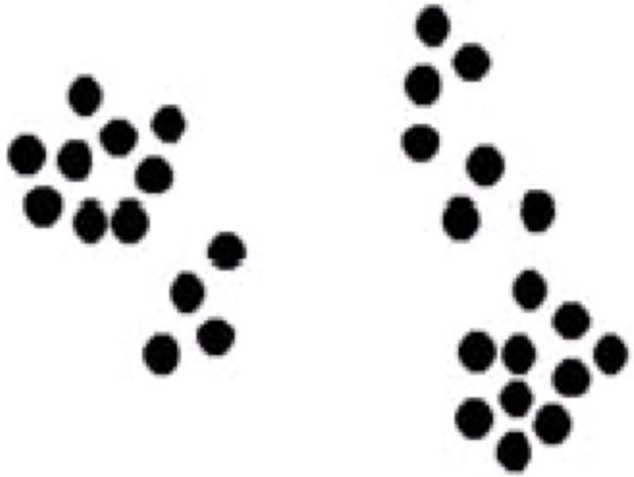
Identify Groups of Customers



Data analysis

Cancer researcher might assay gene expression levels in 100 patients with cancer. He or she might then look for subgroups among the breast cancer samples, or among the genes, in order to obtain a better understanding of the disease

An unlabeled cluster?



- Are there any clusters or groups? How many?
- What is each group? How to identify it?

What is clustering?

Clustering is the process of grouping a set of objects into classes of similar objects

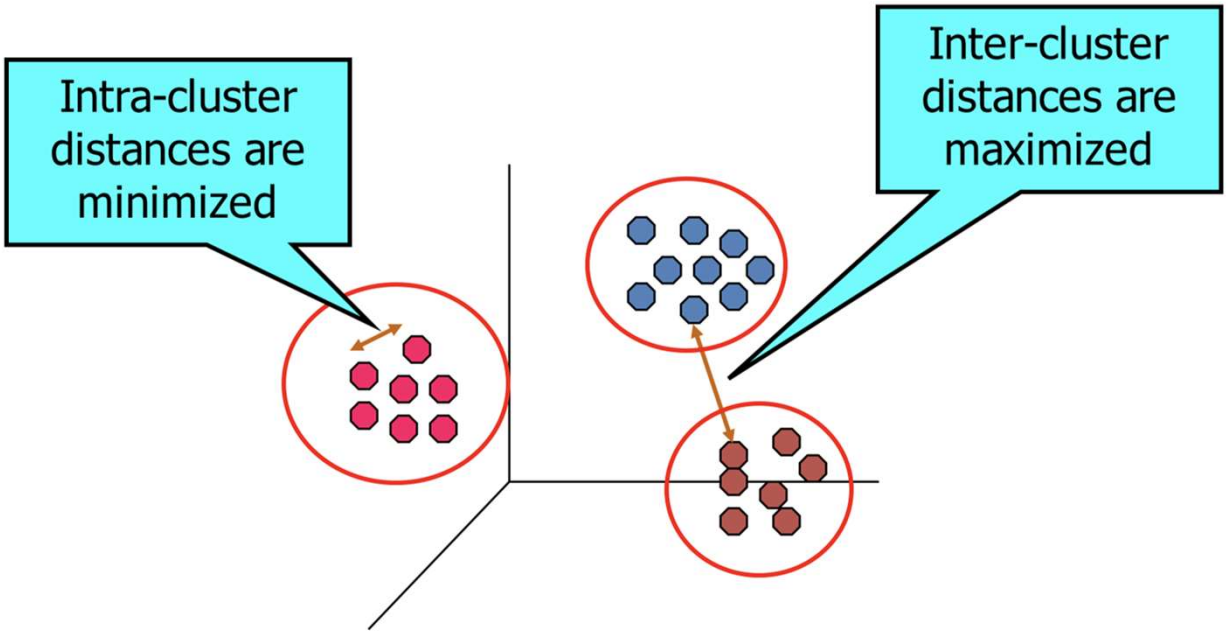
- High intra-class similarity
- Low inter-class similarity

As the **most common** form of unsupervised learning, it has many applications in Science, Engineering, Health,...

- Group genes that perform the same function
- Group customer based on their activity on your website
- Detect any anomaly/ defect in a product
- Segment images into parts according to their color and texture
- ...

How to find a good clustering?

Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another while different from (or unrelated to) the data points in other groups



Challenges of Clustering

- What is the natural grouping among these objects? (**groupness**)
- What makes objects "related"? (**distance of similarity**)
- Representation for objects? (**vector space/ normalization**)
- How many clusters? (**fixed number or data driven**)
- What algorithms to use? (**partitional or hierarchical**)
- Convergence? (**lack of formal theory**)

Clustering

- So clustering tries to find groups of ***similar*** samples
- But how to measure ***similarity***?.....

Similarity Measures

Similarity

- Hard to define, but we know it when we see it!
- Depends on representation and algorithm. Easier to think in terms of a distance between two vectors **a** and **b**.

Distance Measures

- **Symmetry**: D(**a**,**b**) = D(**b**,**a**)
  - Alex looks like Bob, then Bob looks like Alex
- **Self similarity**: D(**a**,**a**) = 0
  - Alex looks like Alex more than anyone else
- **Positivity Separation:** D(**a**,**b**) = 0 iff **a = b**
  - Otherwise you cannot tell anything apart
- **Triangular Inequality**: D(**a**,**b**) ≤ D(**a**,**c**) + D(**b**,**c**)
  - *Loosely*: Alex looks like Bob since Alex looks like Carl and Bob also looks like Carl

Minkowski (1864-1909) Metric

Suppose 2 objects **a** and **b** both have n features:

$$\mathbf{a} = (a_1, a_2, \ldots, a_n)$$
$$\mathbf{b} = (b_1, b_2, \ldots, b_n)$$

The Minkowski metric is defined by:

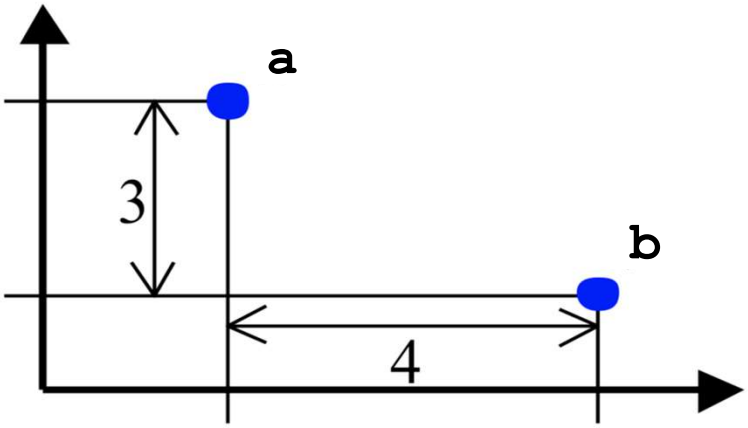
$$D(\mathbf{a}, \mathbf{b}) = \sqrt[p]{\sum_{i=1}^{n} |a_i - b_i|^p}$$

Most commonly used Minkowski Metrics:

1. Manhattan distance (p=1)
2. Euclidean distance (p =2)
3. "Sup" distance (p = ∞)

$$D(\mathbf{a}, \mathbf{b}) = \max_{1 \le i \le p} |a_i - b_i|$$

An Example



1: Euclidean distance: $\sqrt[2]{4^2+3^2}=5.$

2: Manhattan distance: $4+3=7.$

3: "sup" distance: $\max\{4,3\}=4.$

Hamming Distance

Manhattan distance is called the Hamming distance when all features are binary or discrete:

$$D_{Hamming}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} |a_i - b_i|$$

E.g., Gene Expression Levels Under 17 Conditions (1-High,0-Low)

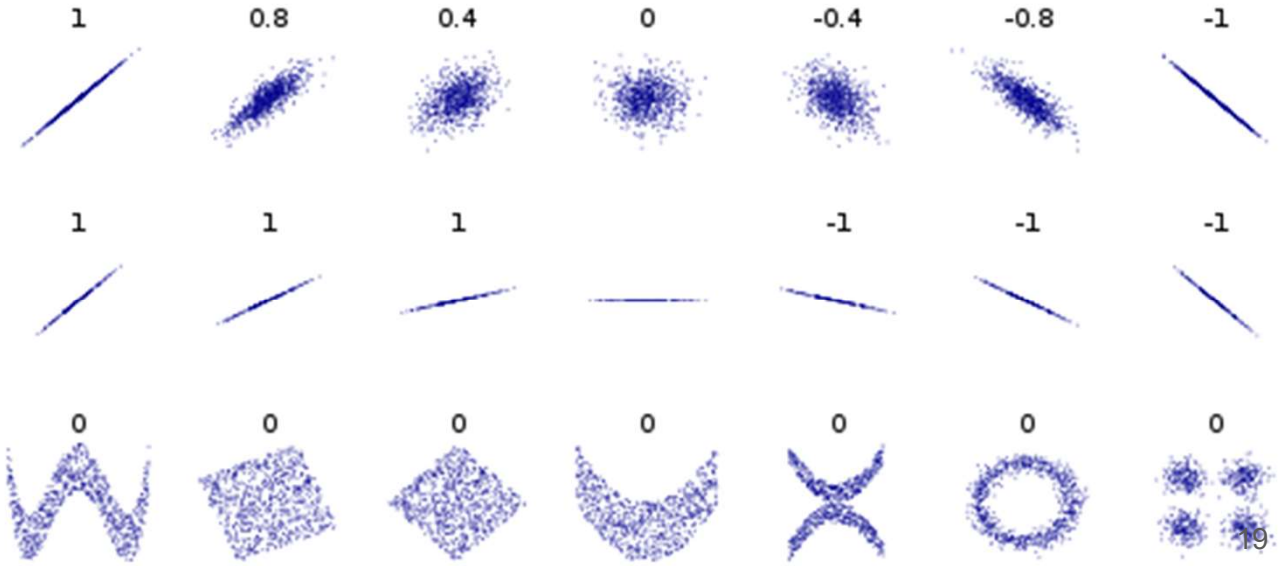
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
GeneA	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
GeneB	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

Hamming Distance: $\#(01) + \#(10) = 4 + 1 = 5$.

Correlation to measure similarity

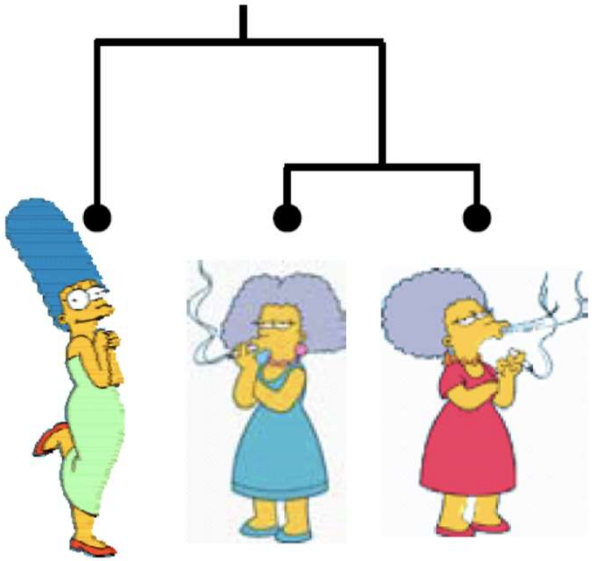
- Pearson Correlation Coefficient to measure the linear correlation between x and y
- Giving a value between -1 and 1 where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation (but you all know this already)

$$d(\mathbf{x}, \mathbf{y}) = \frac{\sum_i (x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sqrt{\sum_i (x^{(i)} - \bar{x})^2 \sum_i (y^{(i)} - \bar{y})^2}}$$



Edit/Transform Distance

To measure the similarity between two objects, **transform** one object into the other, and measure how much effort it takes.



The distance between Patty and Selma.

Change dress color, 1 point
Change earring shape, 1 point
Change hair part, 1 point

D(Patty,Selma) = **3**

The distance between Marge and Selma.

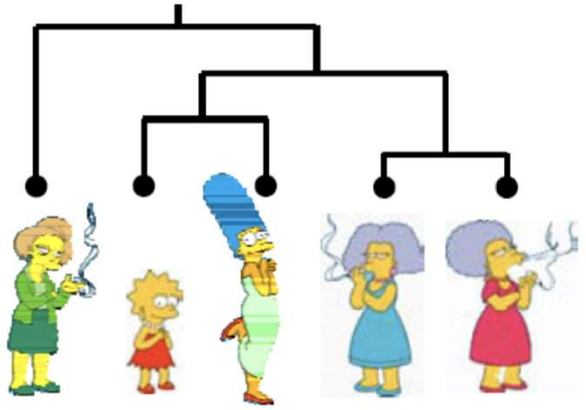
Change dress color, 1 point
Add earrings, 1 point
Decrease height, 1 point
Take up smoking, 1 point
Lose weight, 1 point

D(Marge,Selma) = **5**

Two types of Clustering Algorithms

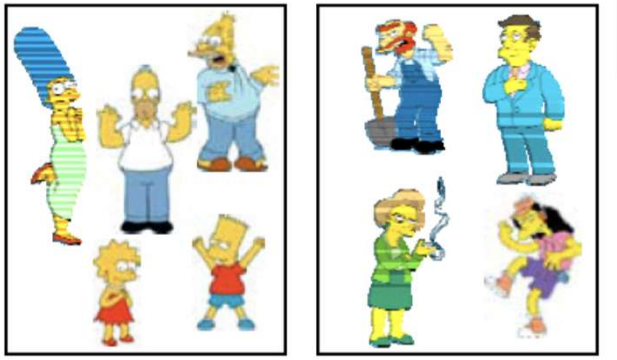
Hierarchical algorithms

- Bottom-up: agglomerative
- Top-down: divisive

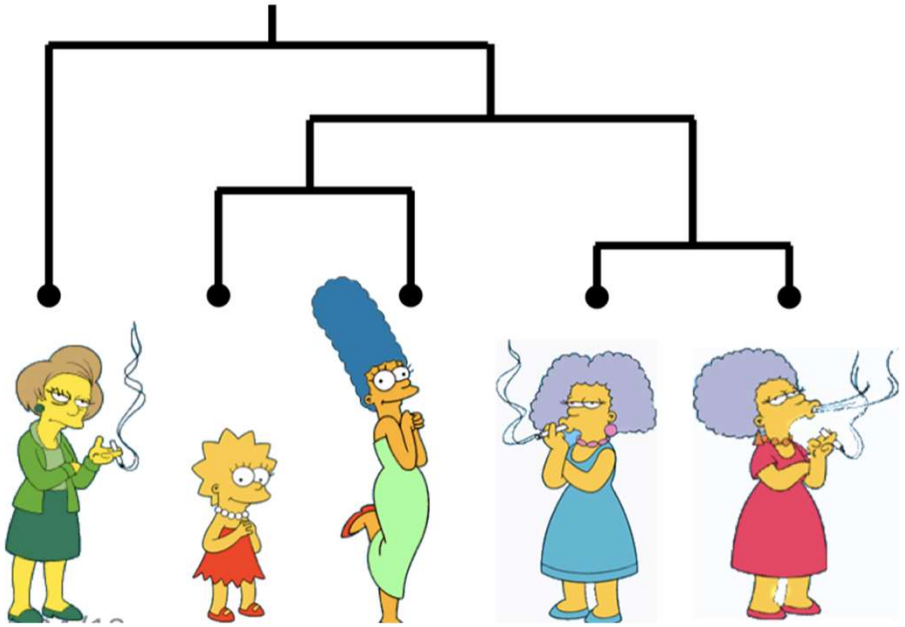


Partitional algorithms

- Usually start with a random (partial) partitioning
- Refine it iteratively (K-means)



Hierarchical Clustering



The number of dendrograms with n leafs
$= (2n - 3)!/[(2^{(n-2)})\ (n - 2)!]$

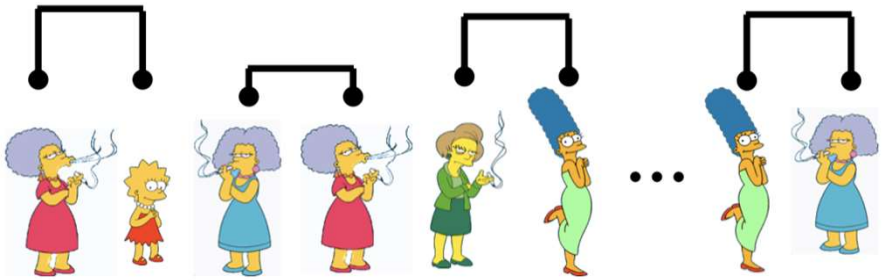
Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	…
10	34,459,425

Bottom-up: Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

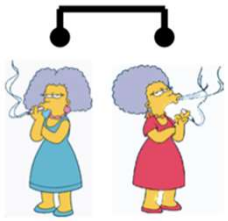
→ **A greedy local optimal solution!**

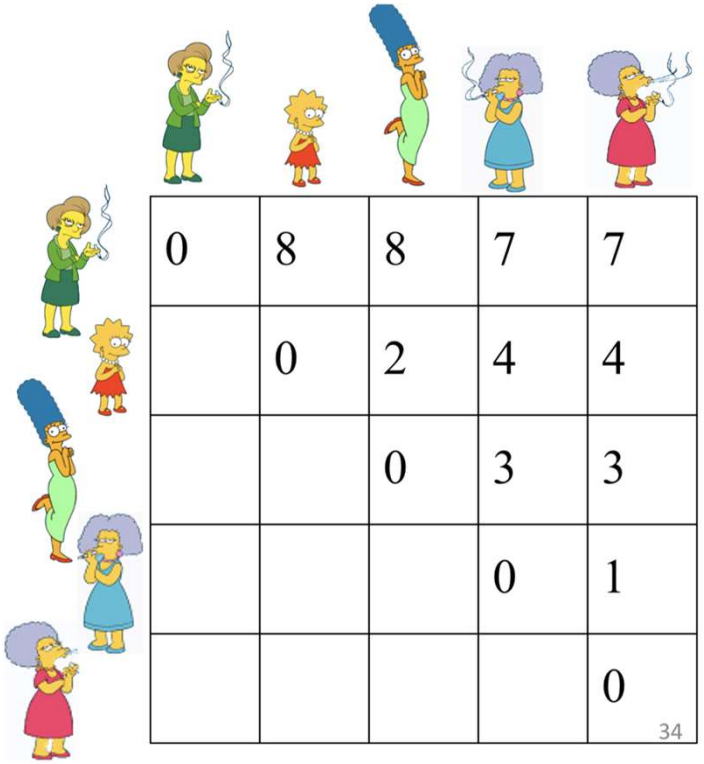
Bottom up Approach

- Begin with a distance matrix which contains the distances between every pair of objects
- Consider all possible merges



- Choose the best:



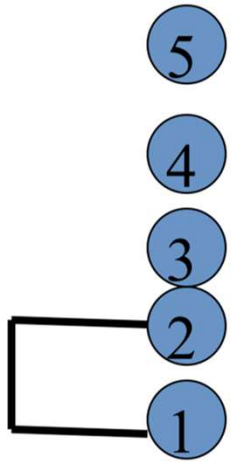


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

34

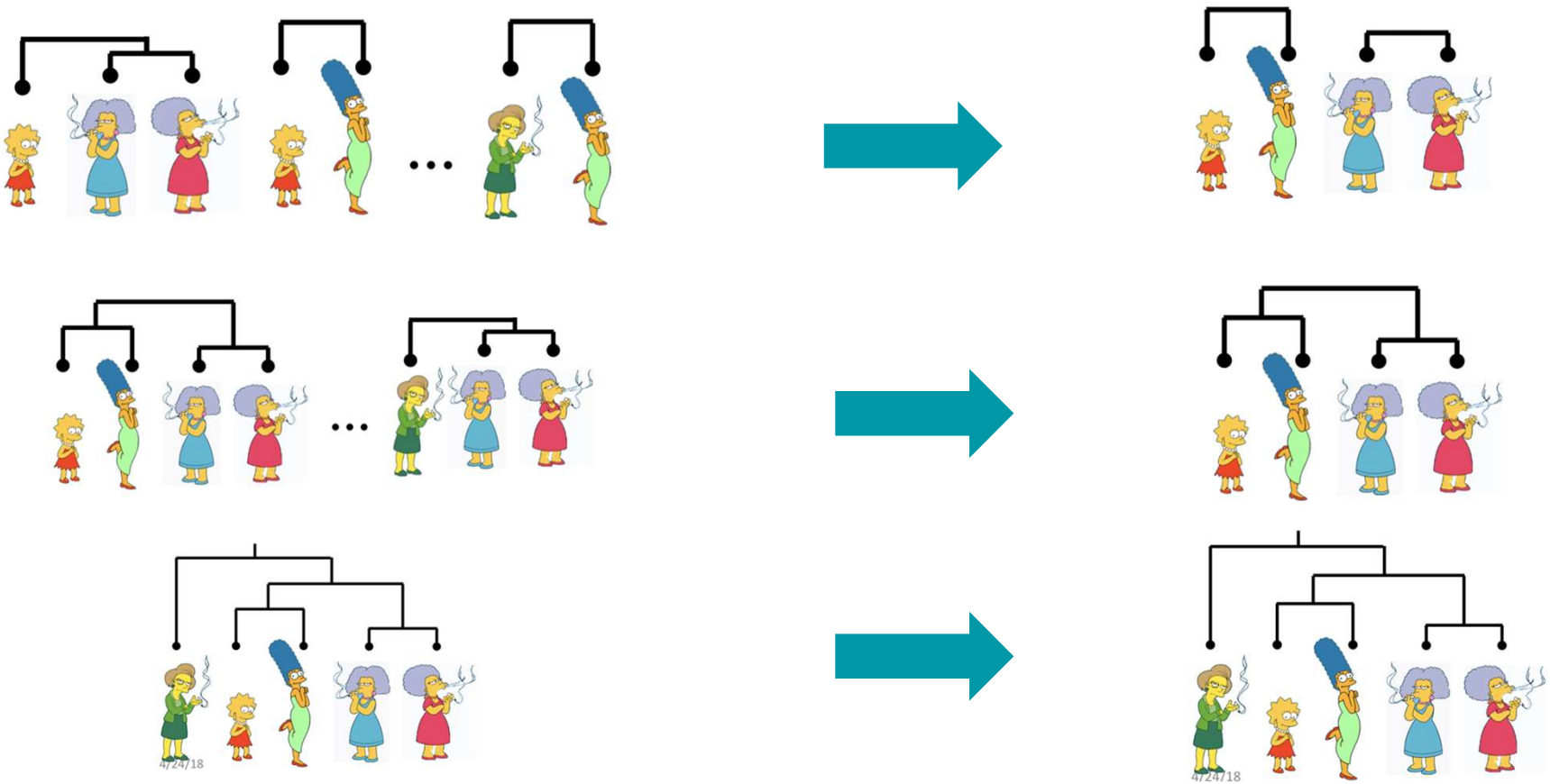
Numerical Example

	1	2	3	4	5
1	0				
2	2	0			
3	6	3	0		
4	10	9	7	0	
5	9	8	5	4	0



Bottom up Approach

- Continue to consider all possible mergers and choose the best:



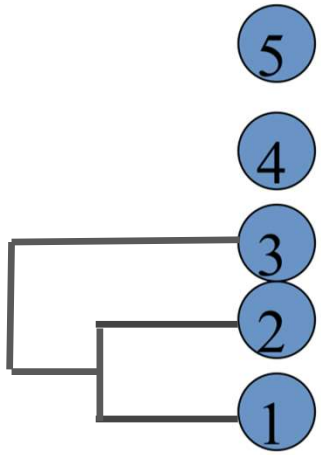
Numerical Example

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & & & & \\ 2 & 2 & 0 & & & \\ 3 & 6 & 3 & 0 & & \\ 4 & 10 & 9 & 7 & 0 & \\ 5 & 9 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|cccc} & (1,2) & 3 & 4 & 5 \\ (1,2) & 0 & & & \\ 3 & 3 & 0 & & \\ 4 & 9 & 7 & 0 & \\ 5 & 8 & 5 & 4 & 0 \end{array}$$

$$d_{(1,2),3} = \min\{ d_{1,3}, d_{2,3}\} = \min\{ 6,3\} = 3$$

$$d_{(1,2),4} = \min\{ d_{1,4}, d_{2,4}\} = \min\{ 10,9\} = 9$$

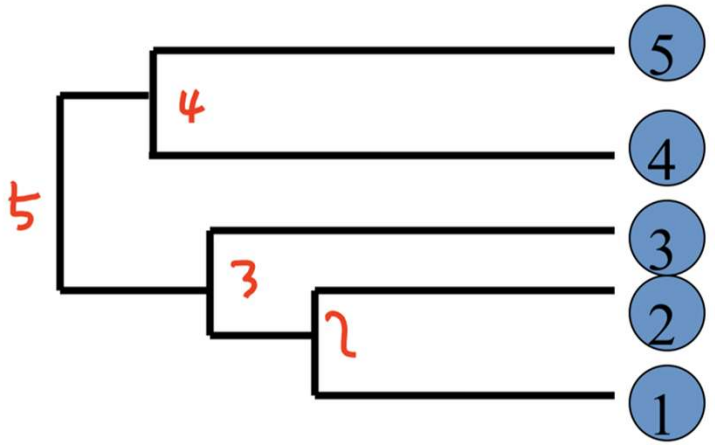
$$d_{(1,2),5} = \min\{ d_{1,5}, d_{2,5}\} = \min\{ 9,8\} = 8$$



Numerical Example

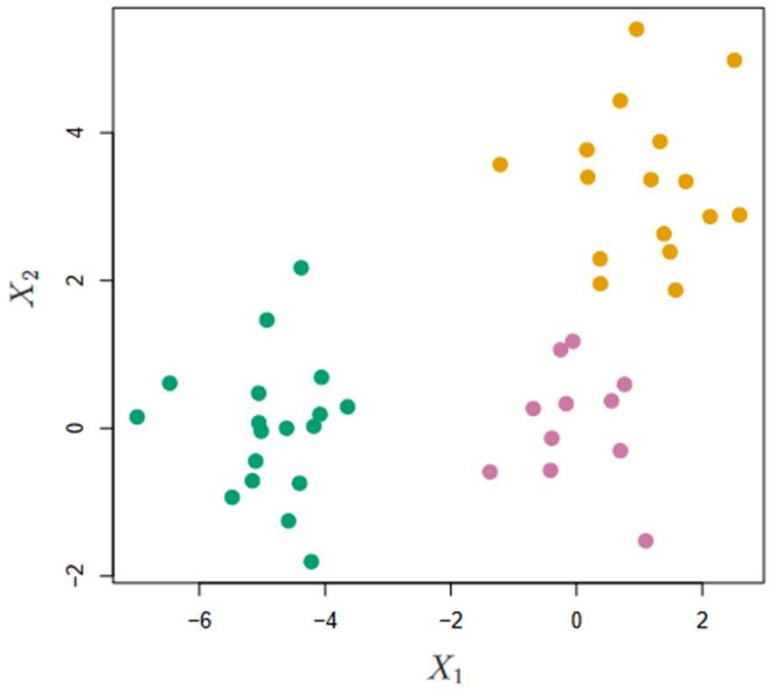
$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2,3) \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccc} (1,2,3) & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2,3),(4,5)} = \min\{ d_{(1,2,3),4}, d_{(1,2,3),5} \} = 5$$



Interpreting a Dendrogram

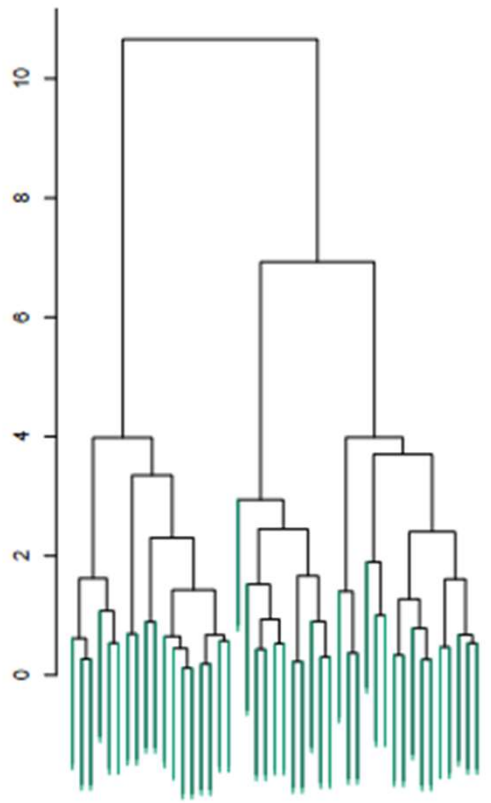
- Forty-five observations generated in two-dimensional space.
- In reality, there are three distinct classes, shown in separate colors. However, we
- will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.



ISLP: ch12

Interpreting a Dendrogram

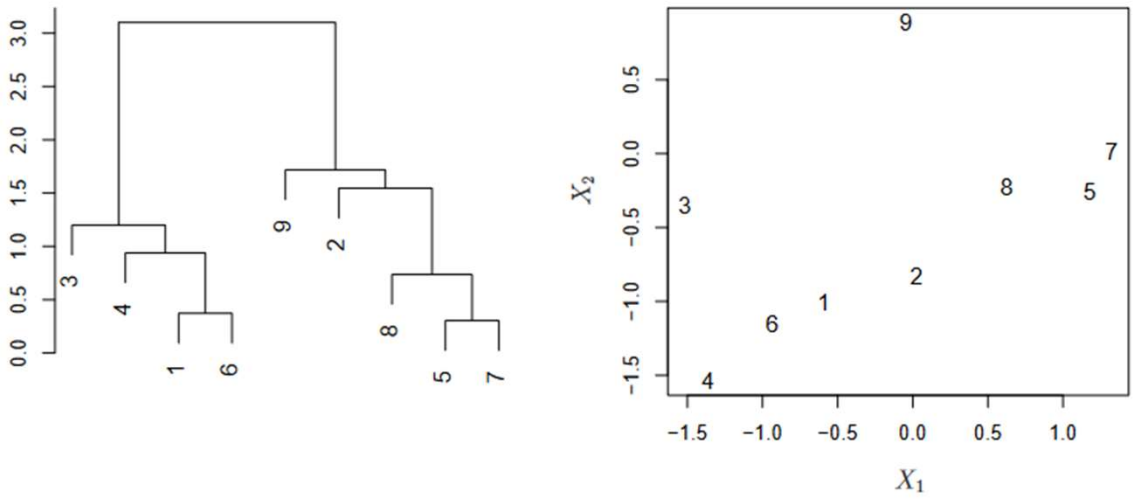
- observations that fuse at the very bottom of the tree are quite similar to each other, whereas observations that fuse close to the top of the tree will tend to be quite different.
- for any two observations, we can look for the point in the tree where branches containing those two observations are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



ISLP: ch12

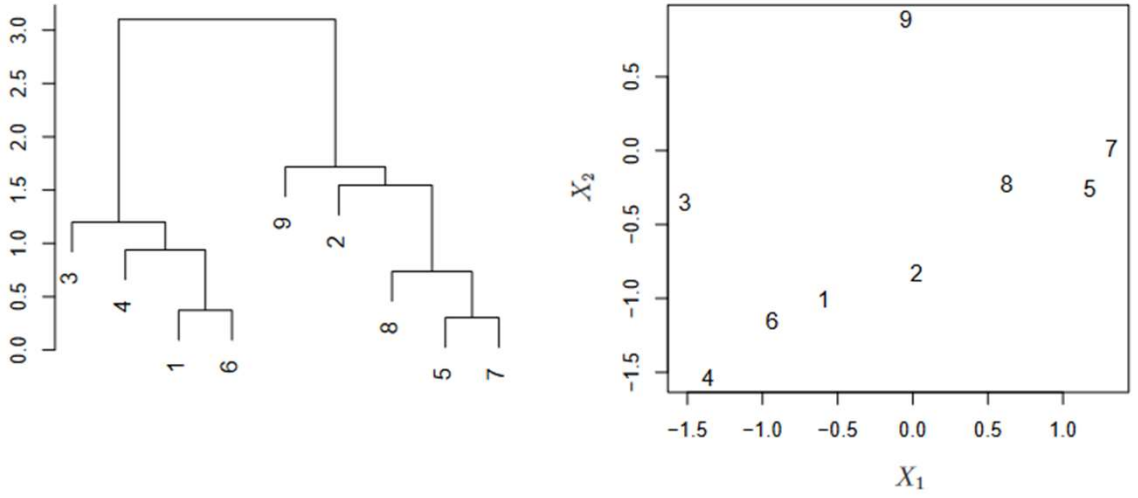
Interpreting a Dendrogram

- 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.
- we cannot draw conclusions about the similarity of two observations based on their proximity along the horizontal axis. Rather, we draw conclusions about the similarity of two observations based on the location on the vertical axis where branches containing those two observations first are fused.



ISLP: ch12

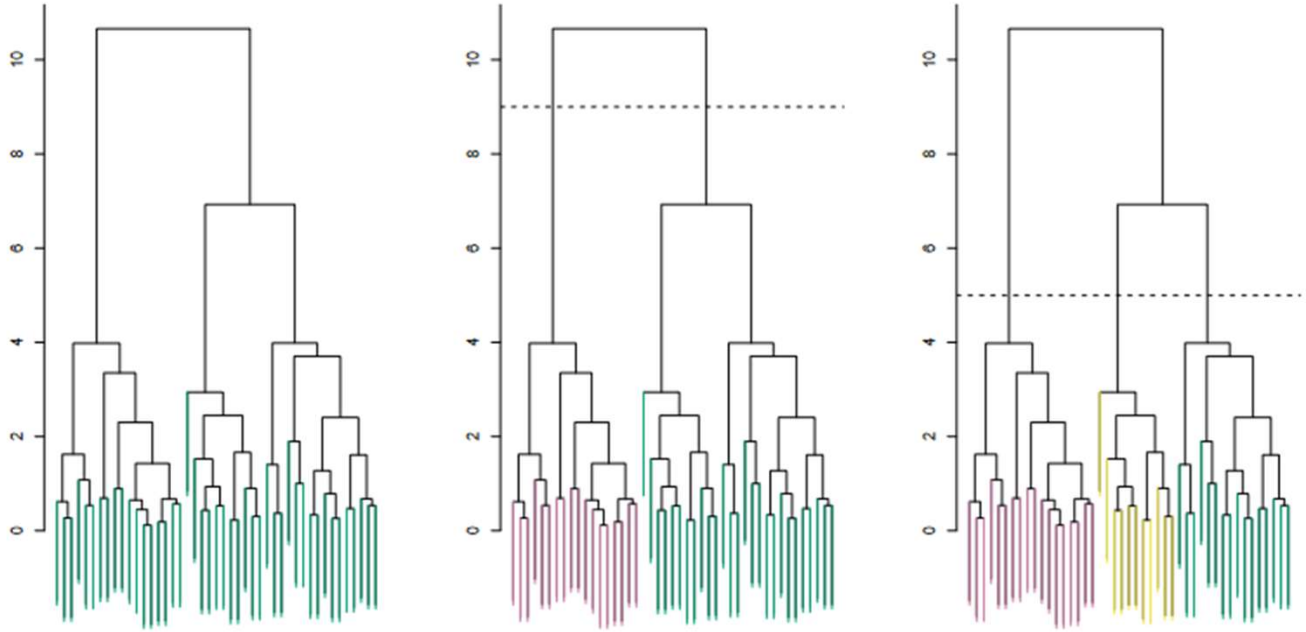
FIGURE 12.12. *An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.* Left: *a dendrogram generated using Euclidean distance and complete linkage. Observations* 5 *and* 7 *are quite similar to each other, as are observations* 1 *and* 6. *However, observation* 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.



ISLP: ch12

Interpreting a Dendrogram

- How to identify clusters?
- The height of the cut to the dendrogram serves the same role as the K in K-means clustering it controls the number of clusters obtained



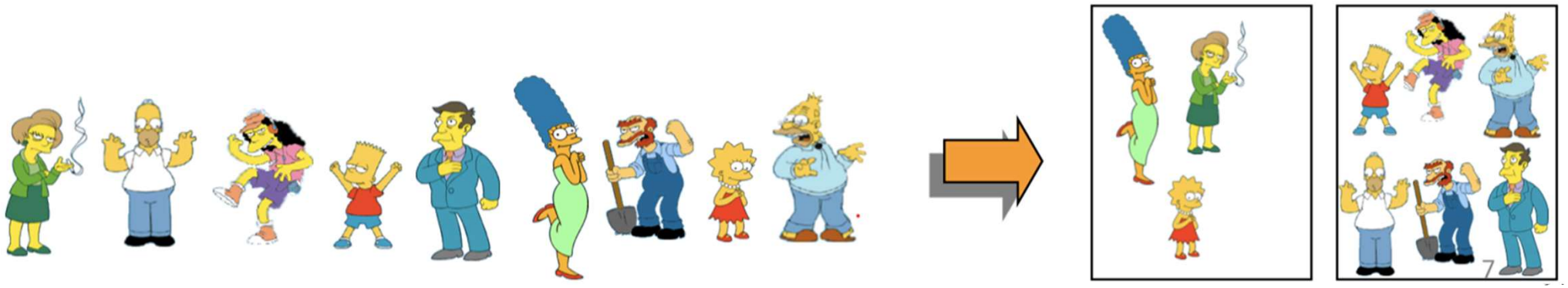
ISLP: ch12

Partition Clustering (K-means)

ISLP: ch12

Partitional Clustering

- Non-hierarchical
- Construct a partition of m objects into a set of k clusters
- User has to specify the desired number of clusters



Partitioning Algorithms

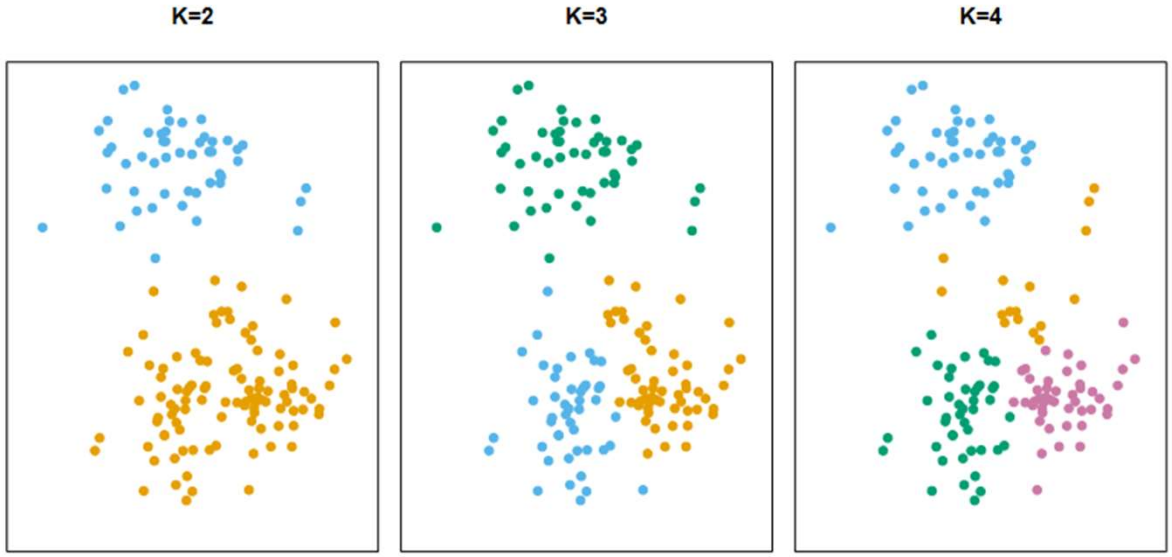
Given: a set of objects and the number k

Find: a partition of k clusters that optimizes a chosen partitioning criterion

- **Globally optimal:** exhaustively enumerate all partitions (often too expensive)
- **Effective Heuristic methods:** k-means

K-means Clustering

- K-means clustering is that a good clustering is one for which the within-cluster variation is as small as possible
- a local optimum of an optimization problem is a solution that is optimal within a neighboring set of candidate solutions.



ISLP: ch12

K-means

Algorithm 12.2 *K-Means Clustering*

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).

ISLP: ch12

K-means

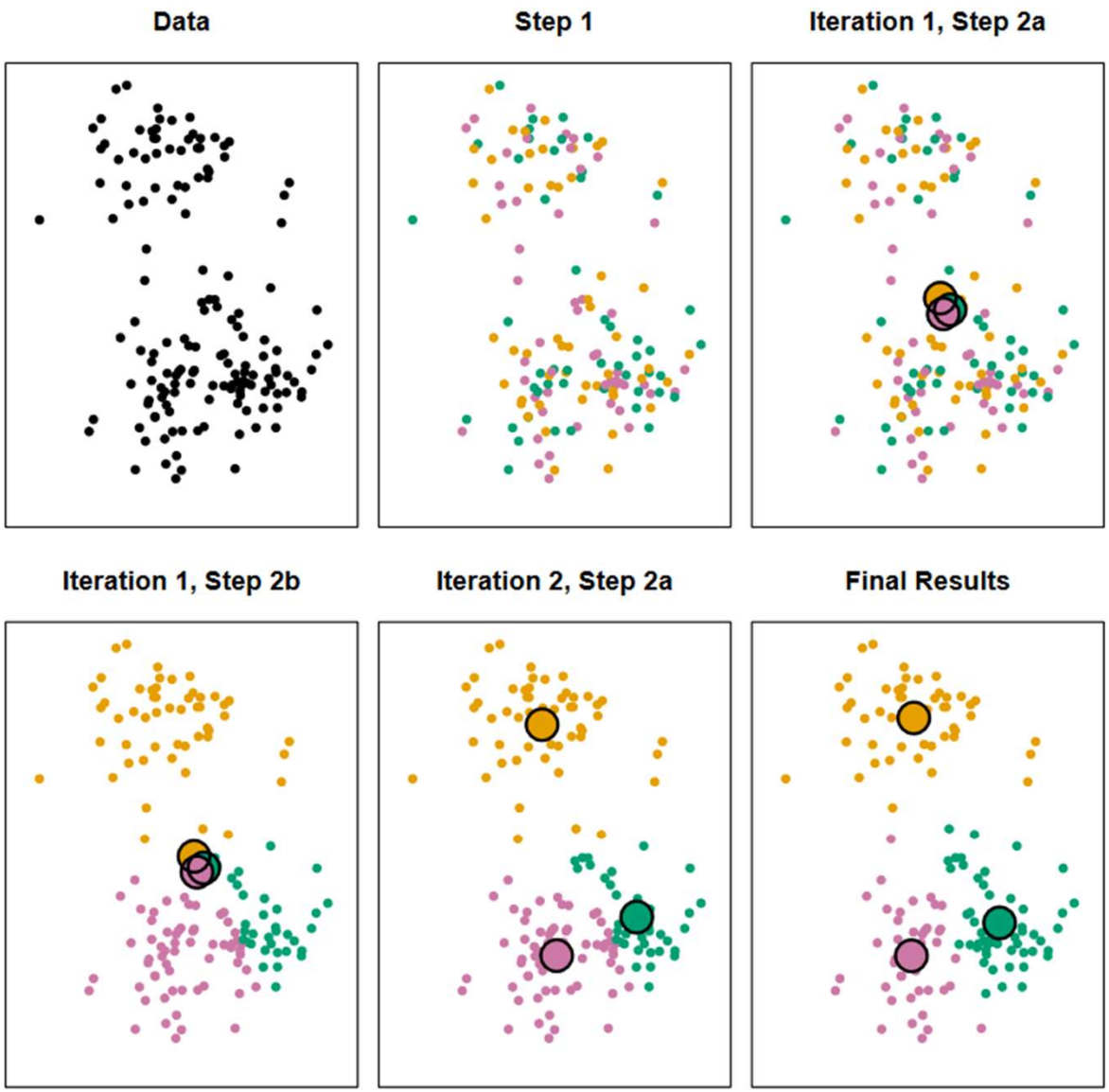
Algorithm 12.2 *K-Means Clust*

1. Randomly assign a number,
   These serve as initial cluste

2. Iterate until the cluster assi

   (a) For each of the K clu
   kth cluster centroid is
   observations in the kth

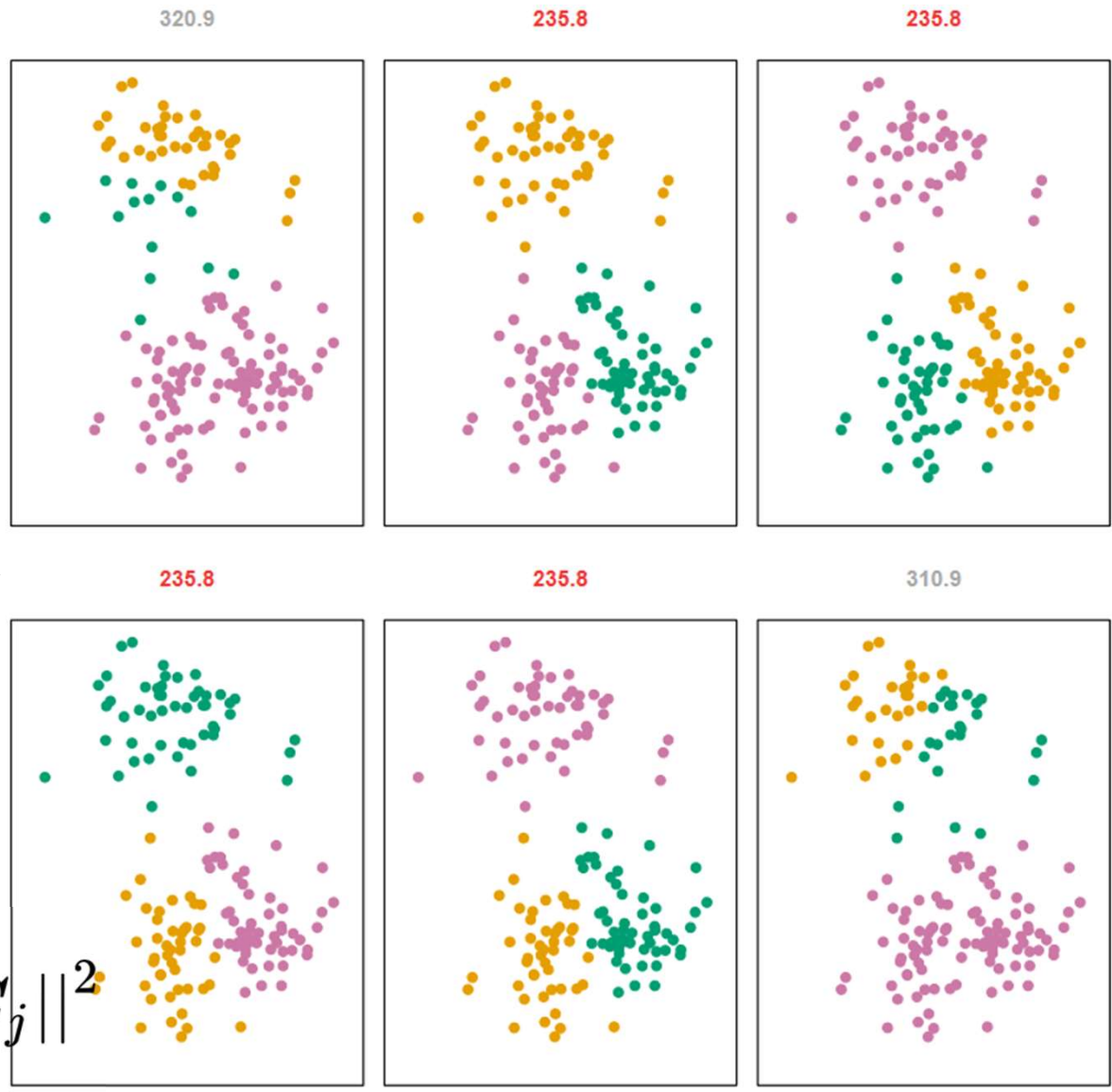
   (b) Assign each observatio
   (where *closest* is define



ISLP: ch12

K-means

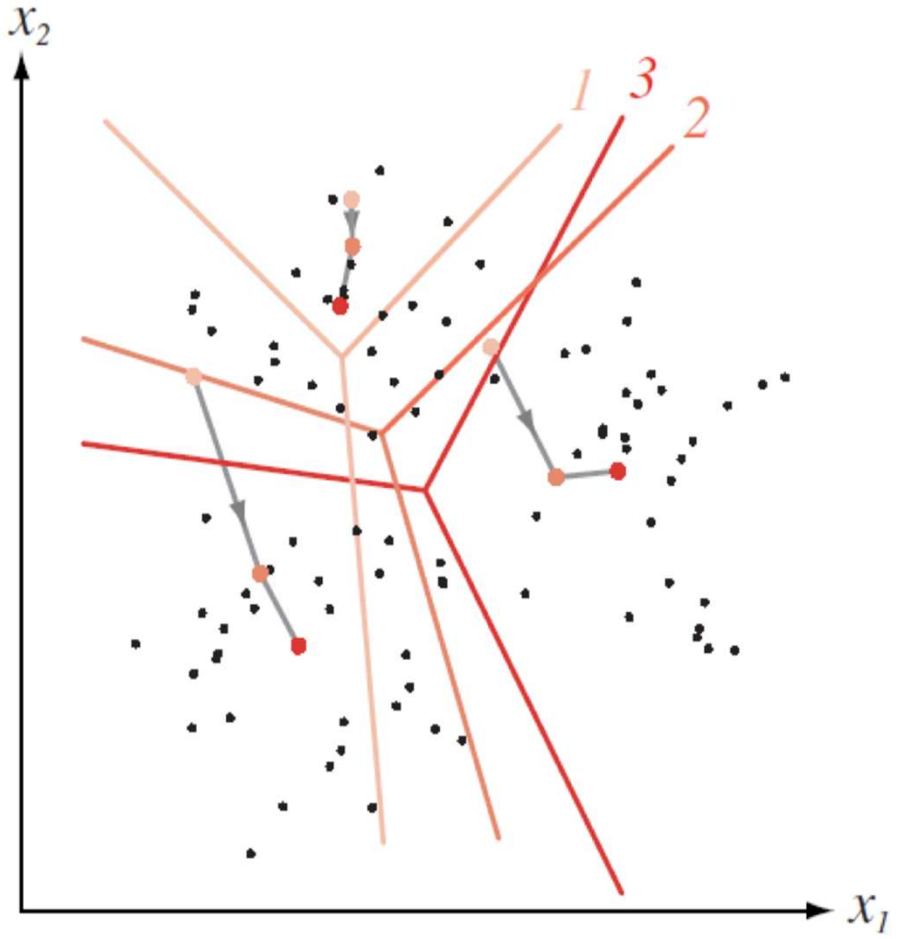
- K-means clustering performed six times on the data with K = 3, each time with a different random assignment of the first step
- Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters.
- Those labeled in red all achieved the same best solution, with an objective value of 235.8



$$J = \arg\min_{C_j} \sum_{j=1}^{k} \sum_{i=1}^{m} ||x^{(i)} - C_j||^2$$

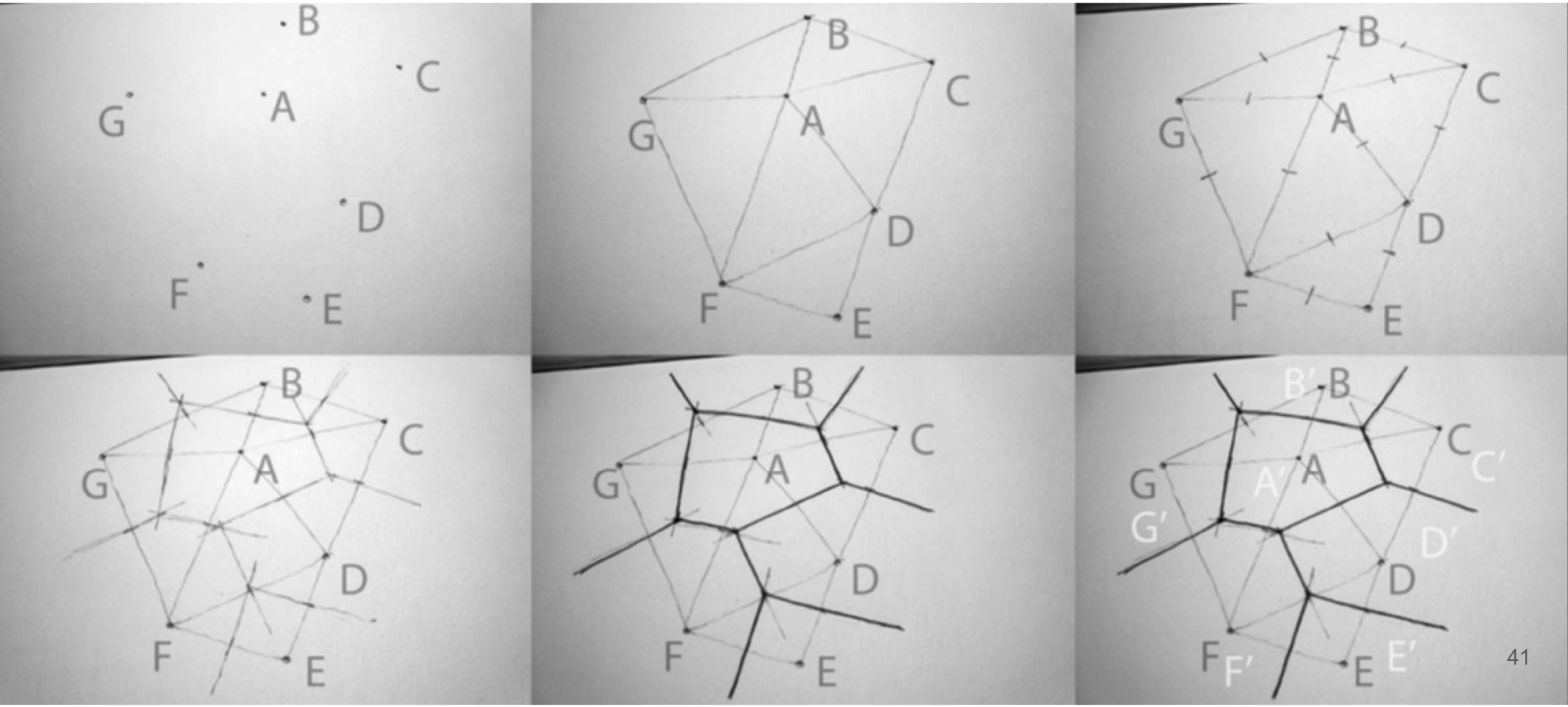
ISLP: ch12

How K-means partition

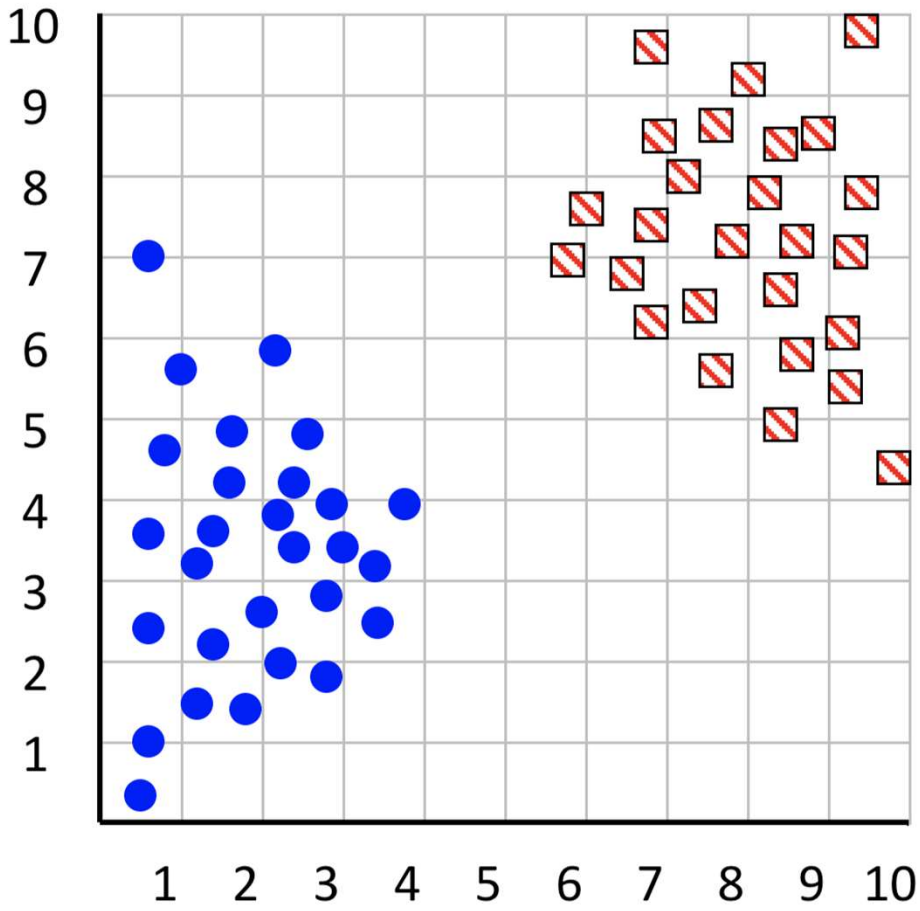


- When k centroids are set, they partition the whole data space into k mutually exclusive subspace to form a partition.
- A partition amounts to a Voronoi Diagram
- Changing positions of centroids leads to a new partitioning.

How to draw Voronoi Diagram



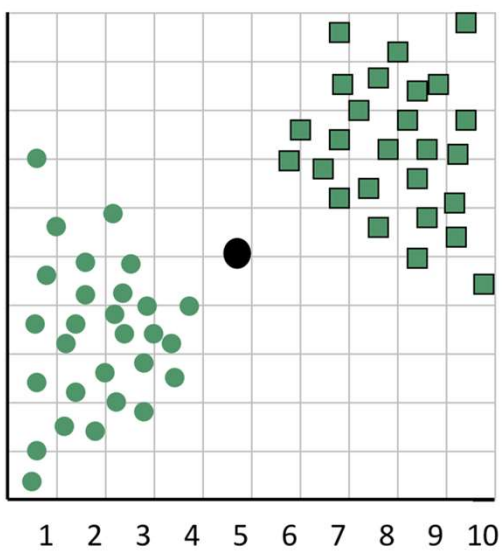
How to determine the right number of clusters



This is an **unsolved** problem, but we can still approximate by trying different k and measure the **total variation** following objective function:

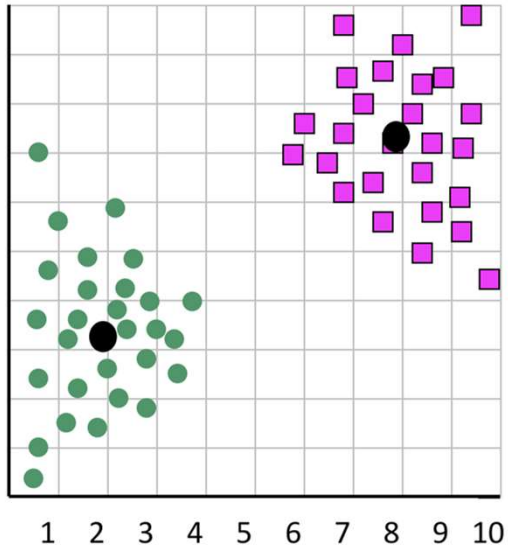
$$J = \arg\min_{C_j} \sum_{j=1}^{k} \sum_{i=1}^{m} ||x^{(i)} - C_j||^2$$

Different values of K



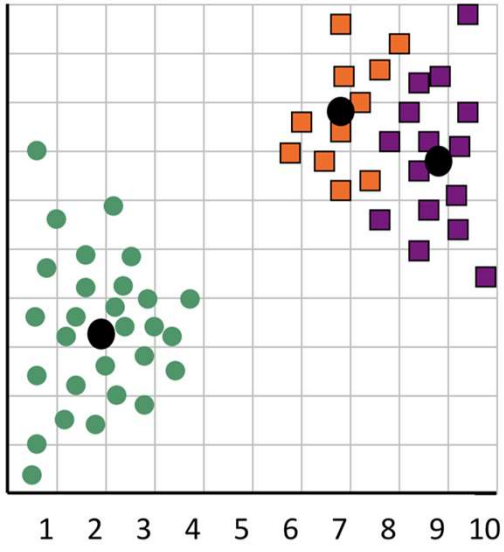
$k = 1$

$\Rightarrow J = 873.0$



$k = 2$

$\Rightarrow J = 173.1$



$k = 3$

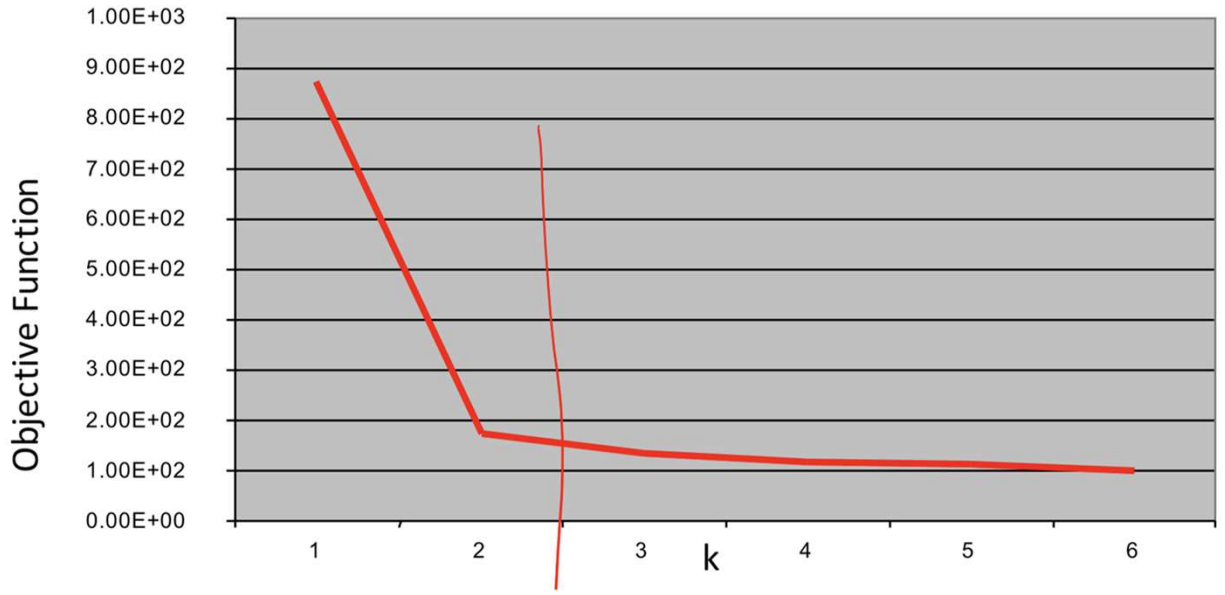
$\Rightarrow J = 133.6$

$k = m$

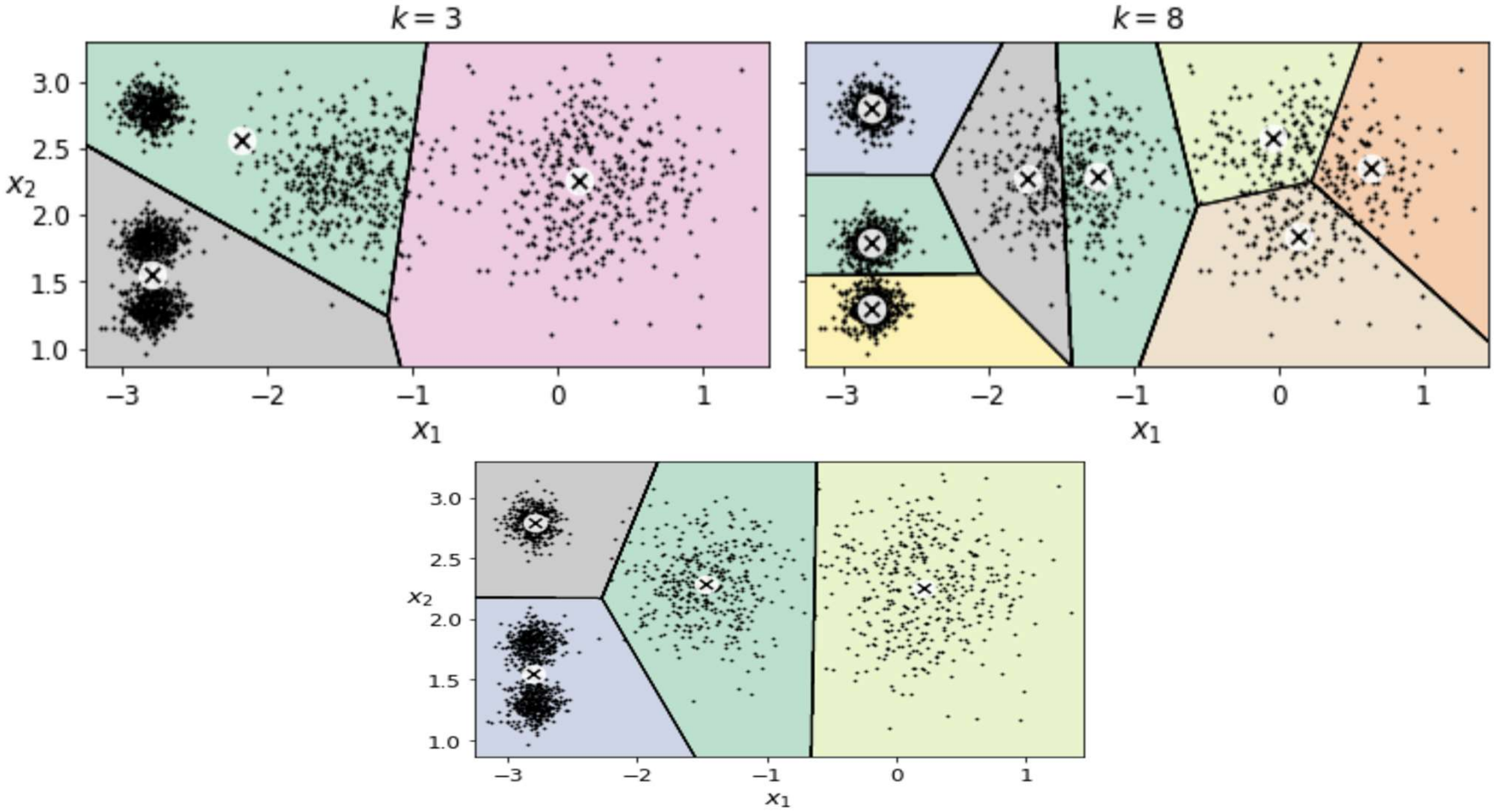
$\Rightarrow J = 0$

Finding the “Elbow”

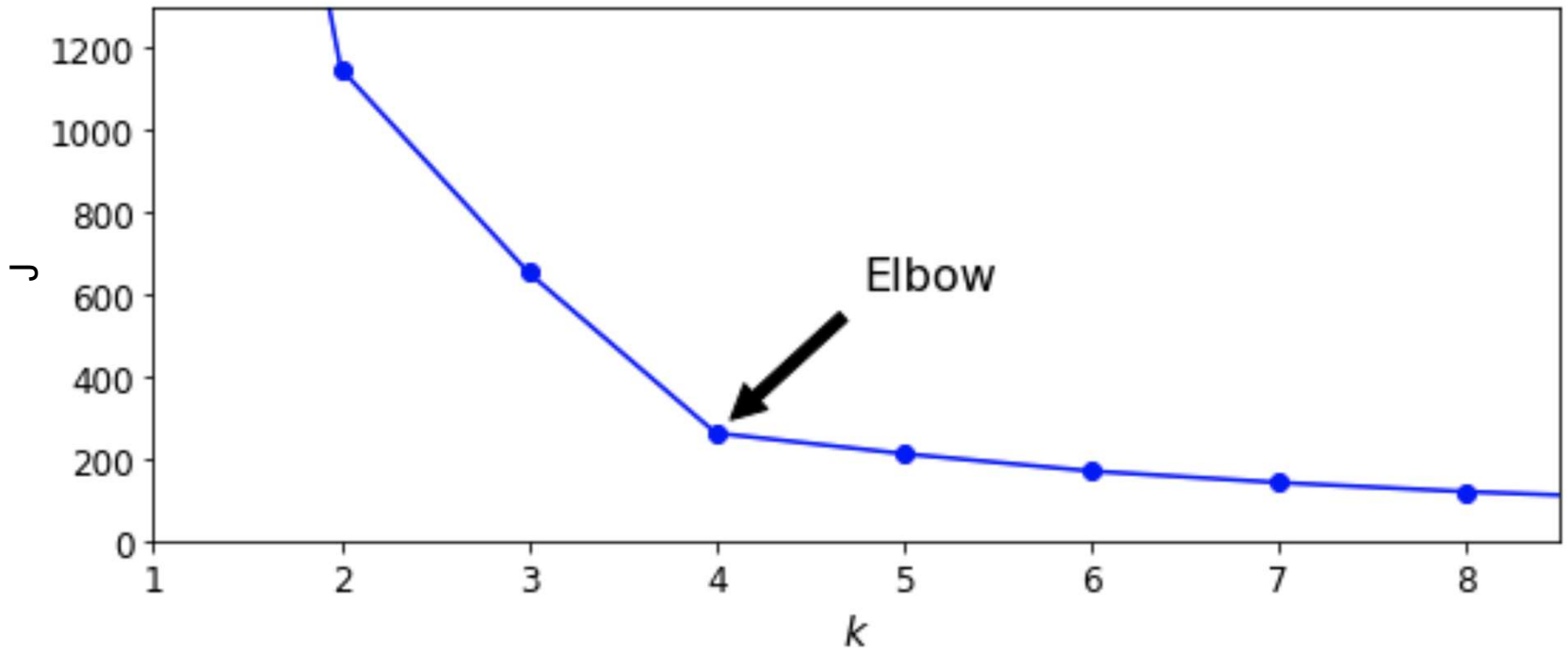
- Plot the objective function for k from 1 to 6
- The abrupt change at k=2 highly suggestive of two clusters in the data, but the results are not always as clear cut in this toy example



Example



Elbow Curve Example



Convergence

When K-means ever reach a fixed point?

- A state in which clusters **no longer change**

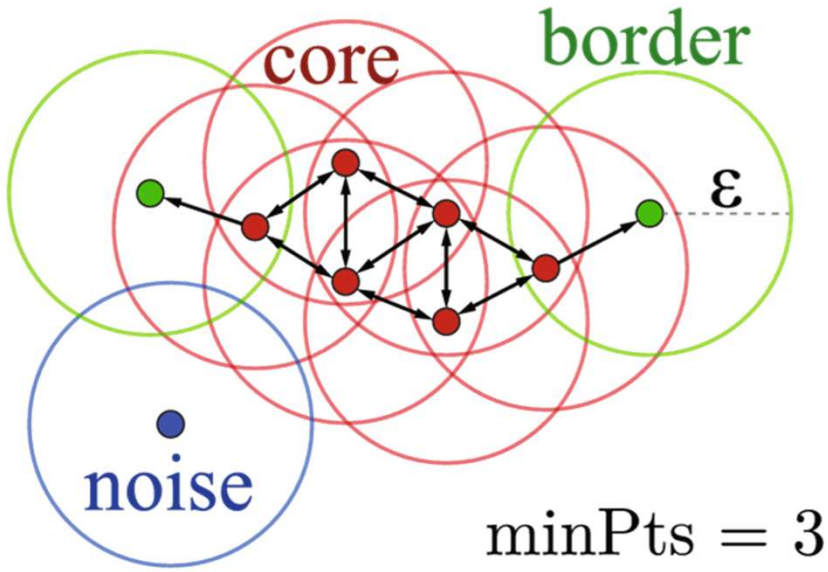
K-means is a special case of a general procedure known as **Expectation Maximization (EM)** algorithm.

- EM is known to converge
- Number of iterations could be large

DBSCAN and other clustering methods

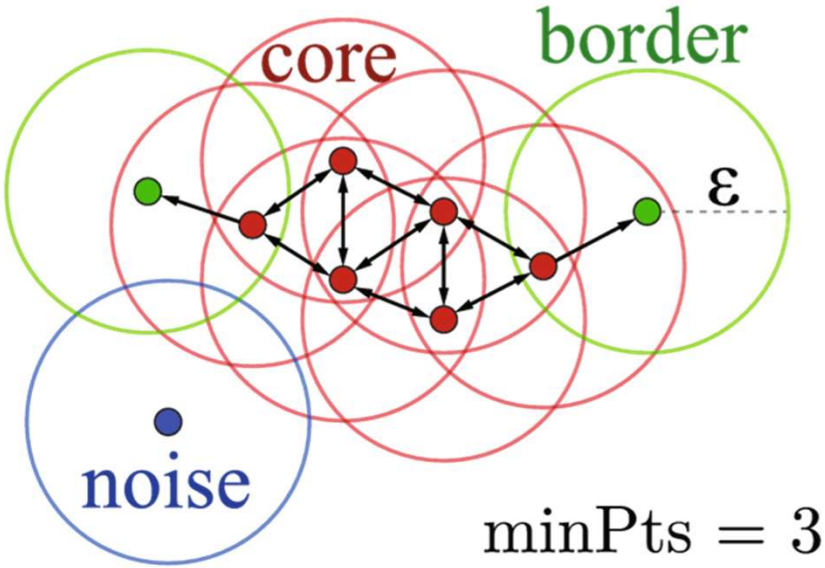
DBSCAN

- Density-based spatial clustering of applications with noise (DBSCAN) is another clustering algorithm which is based on local density estimation.
- Simple yet powerful algorithm which allows identification of arbitrary shapes
- Works well if all the clusters are dense enough and if they are well separated by low-density region.



DBScan Algorithm

- For each data point, count how many points are located within a small distance ε (epsilon) from it. This region is called ε-neighborhood.
- If a data point has at least min_samples in its ε-neighborhood, then it is considered a **core point**.
- All data points in the neighborhood of a core point belong to the same cluster
- Any point that is not a core point and does not have one in its neighborhood is considered an **anomaly** (**noise**?)



DBScan vs K-means

- DBSCAN works with cluster of arbitrary shapes and sizes, K-Means assumes spherical clusters.
- DBSCAN does not assign noise and outlier points to any cluster → robust to noise

- DBSCAN has more parameters than K-Means
- DBSCAN is not efficient for large datasets

Other clustering algorithms

- **BIRCH** (Balanced Iterative Reducing and Clustering using Hierarchies): designed for large m, and small n; builds a tree structure
- **Mean-shift**: place a circle centered on each data point, then compute the mean of all instances located within it, and shift the circle so that it is centered on the mean. Iterate until the all the circles stop moving
- **Affinity Propagation:** uses a voting system were data points vote for similar points to be their representatives
- **Spectral Clustering:** takes a similarity matrix between the data points and creates a low-dimensional embedding from it

Summary

- ✓ Discuss Unsupervised Learning methods using the Simpsons!
- ✓ Represent “group” with distance of similarity
- ✓ Learning some clustering algorithms: partitional and hierarchical
- ✓ See in detail how K-means algorithm works

Acknowledgements

Big thanks to these professors for providing part of the content for this lecture:

- Yanyun Qi, UVA
- Eric Xing, CMU
- Rong Jin, MSU