

# **Advanced Data Analysis**

**DATA 71200**

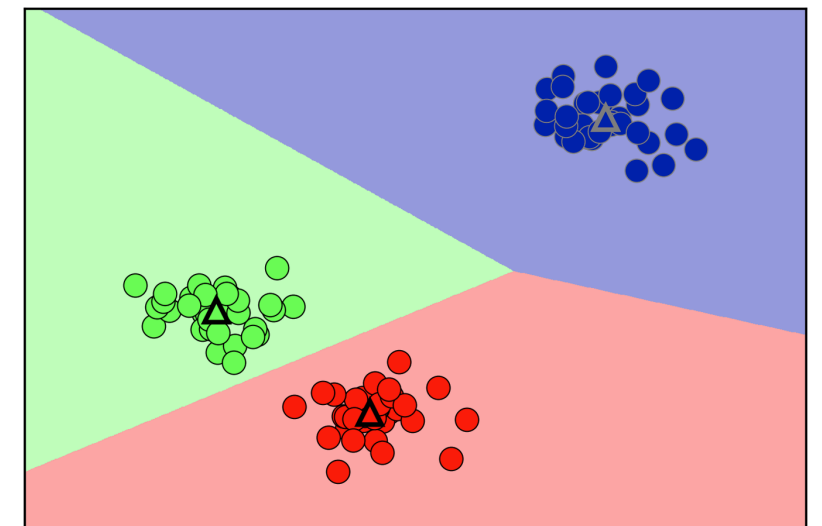
Class 15: Unsupervised Learning (Clustering)

# Clustering

- ▶ **Algorithms that assign data points to groups (especially for unlabeled data)**
  - In the absence of labels, evaluation is challenging
    - Often performed through visualization
- ▶ **Useful for**
  - Exploratory data analysis
  - Pre-processing data

# ***k*-Means**

- ▶ ***k* - number of clusters specified**
- ▶ **Finds cluster centers through an iterative process**
  - Assign data points to cluster with nearest cluster center
    - Initialized randomly for the first iteration
  - Update the cluster center with the assigned data points
  - Repeat until no updates are needed
- ▶ **Boundaries are determined by placement of cluster centers**



*Figure 3-24. Cluster centers and cluster boundaries found by the k-means algorithm*

# $k$ -Means

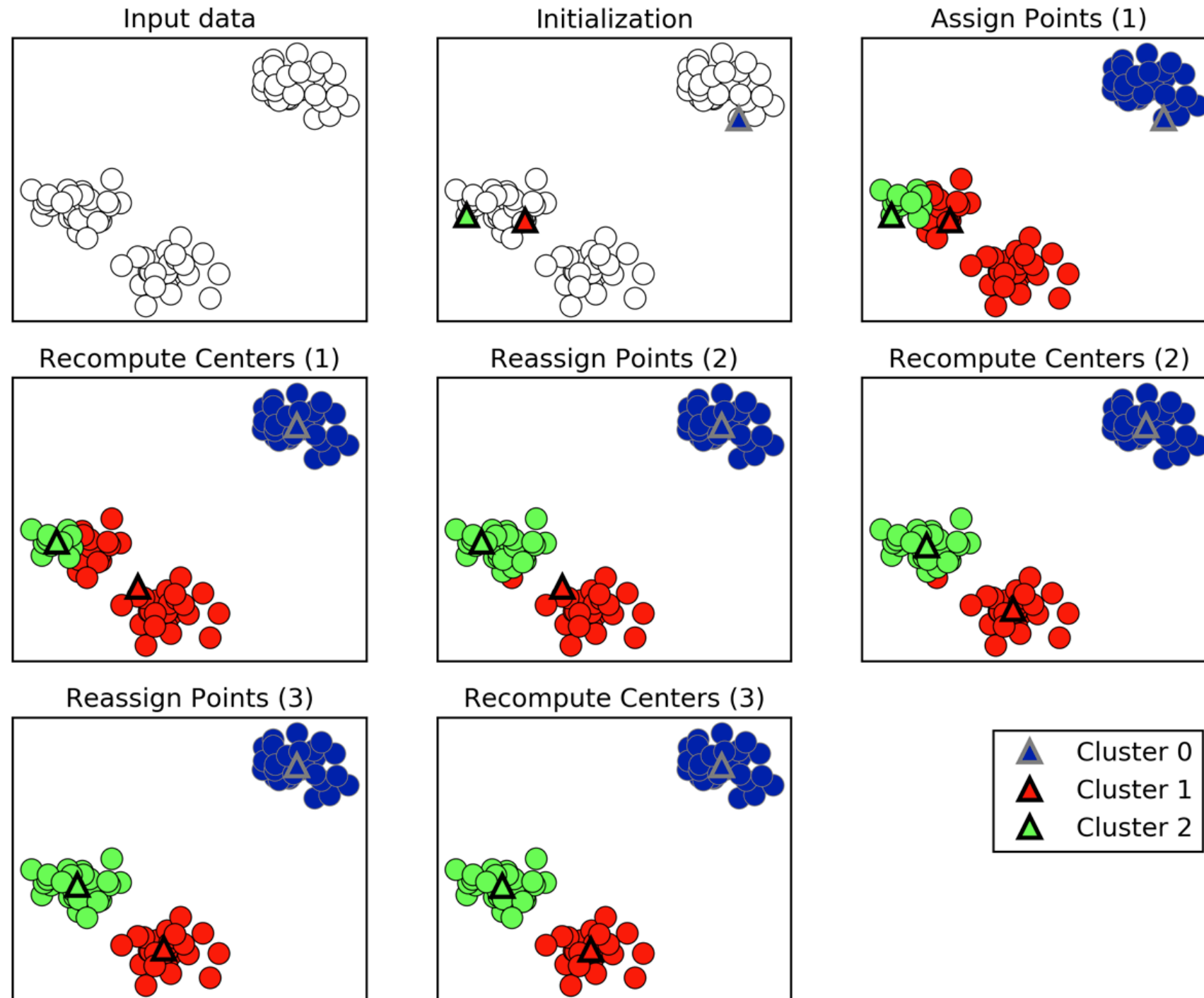


Figure 3-23. Input data and three steps of the  $k$ -means algorithm

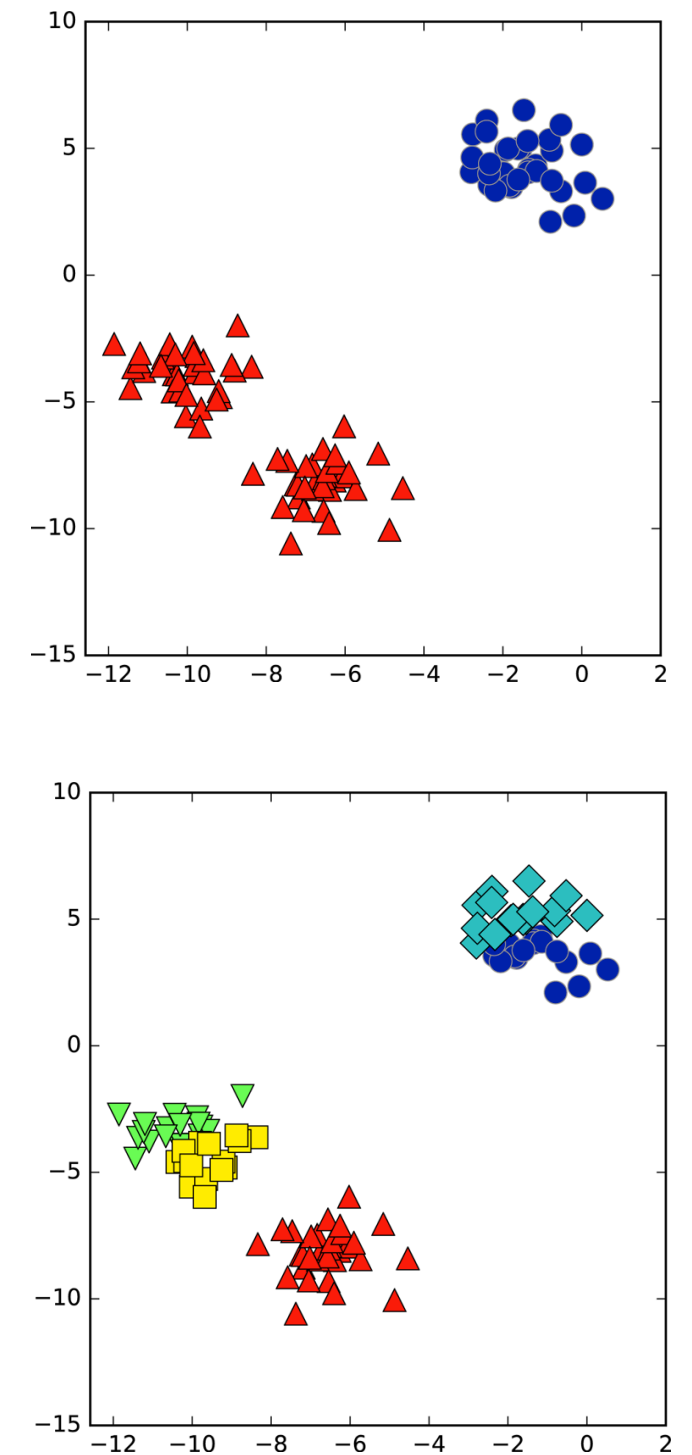


Figure 3-26. Cluster assignments found by  $k$ -means using two clusters (top) and five clusters (bottom)

# $k$ -Means

- ▶ Assumes the classes have the same width/diameter
- ▶ This causes issues with non-spherical clusters or clusters with complex shapes

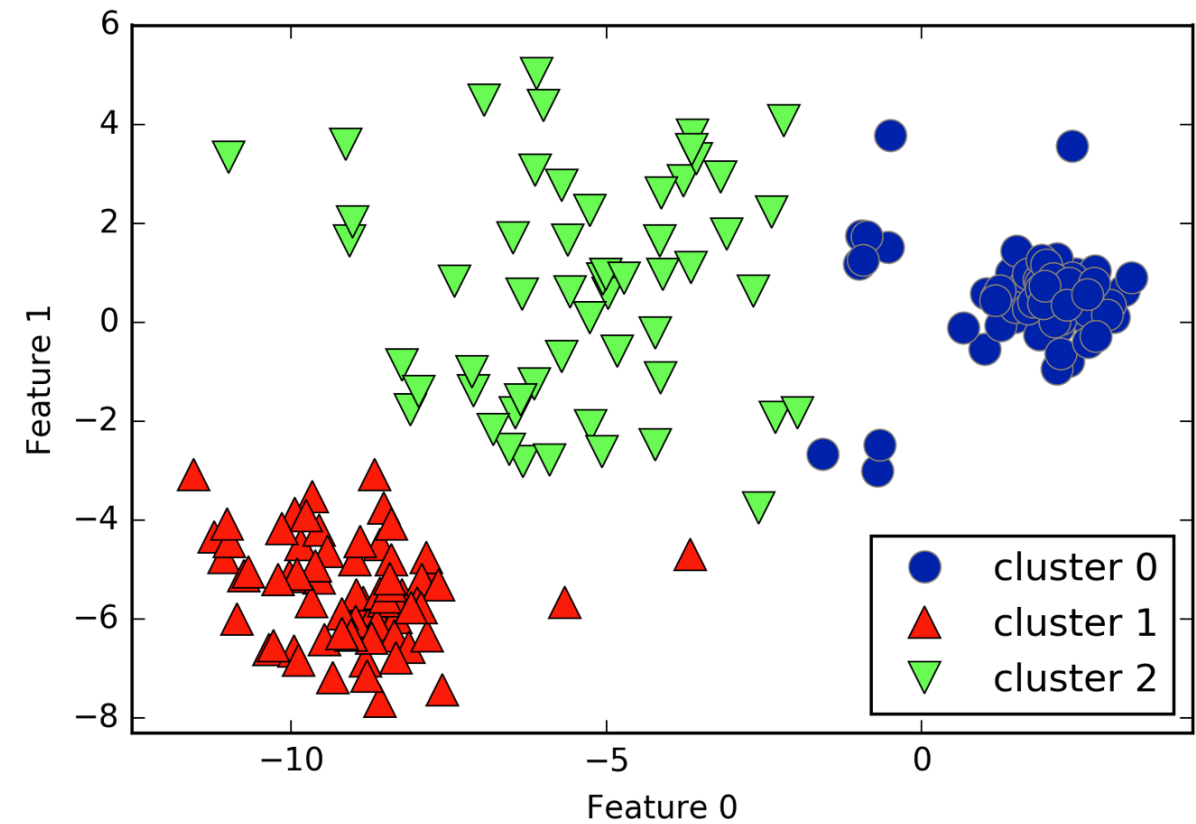


Figure 3-27. Cluster assignments found by  $k$ -means when clusters have different densities

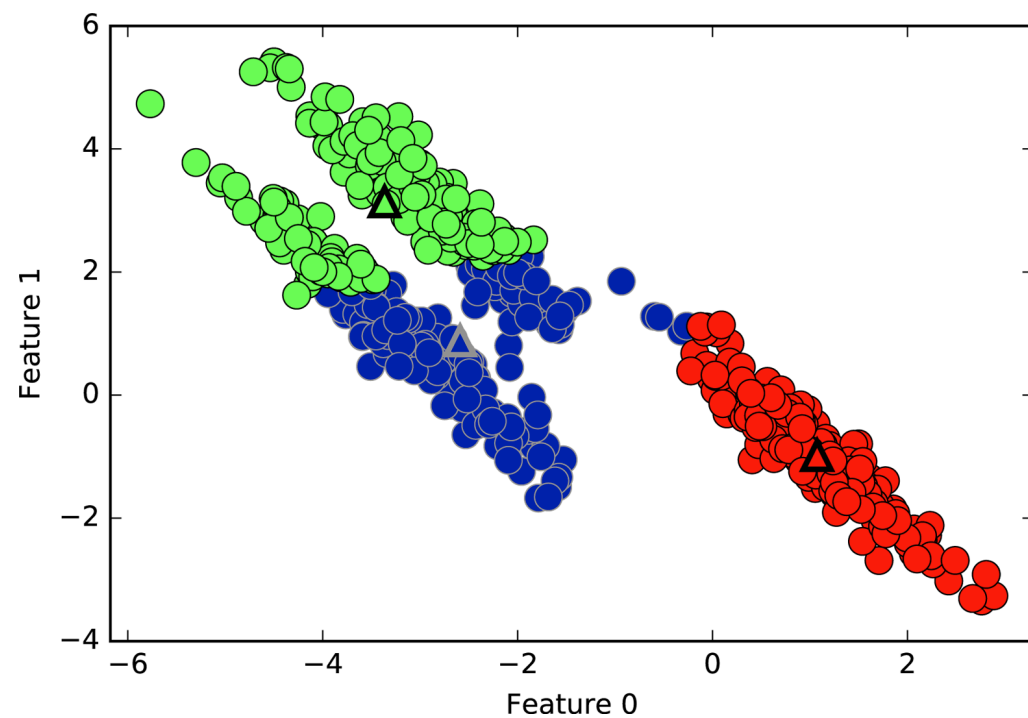


Figure 3-28.  $k$ -means fails to identify nonspherical clusters

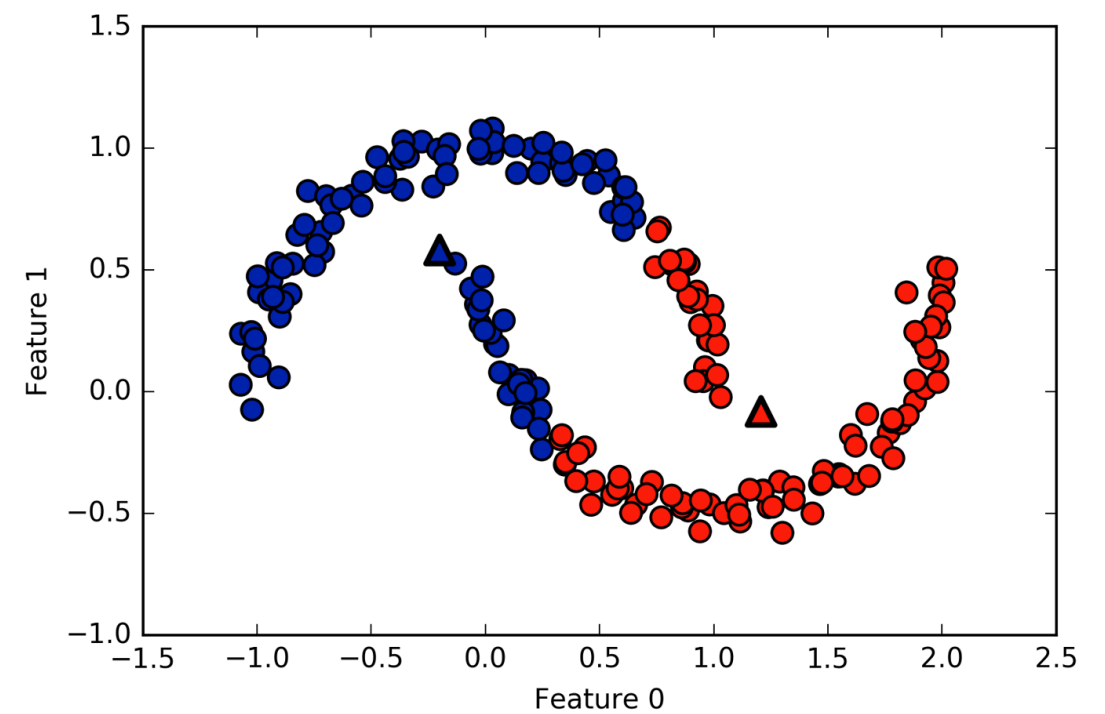


Figure 3-29.  $k$ -means fails to identify clusters with complex shapes

# $k$ -Means

- ▶ **The constant-width limitation can be partially overcome with a larger number of clusters**

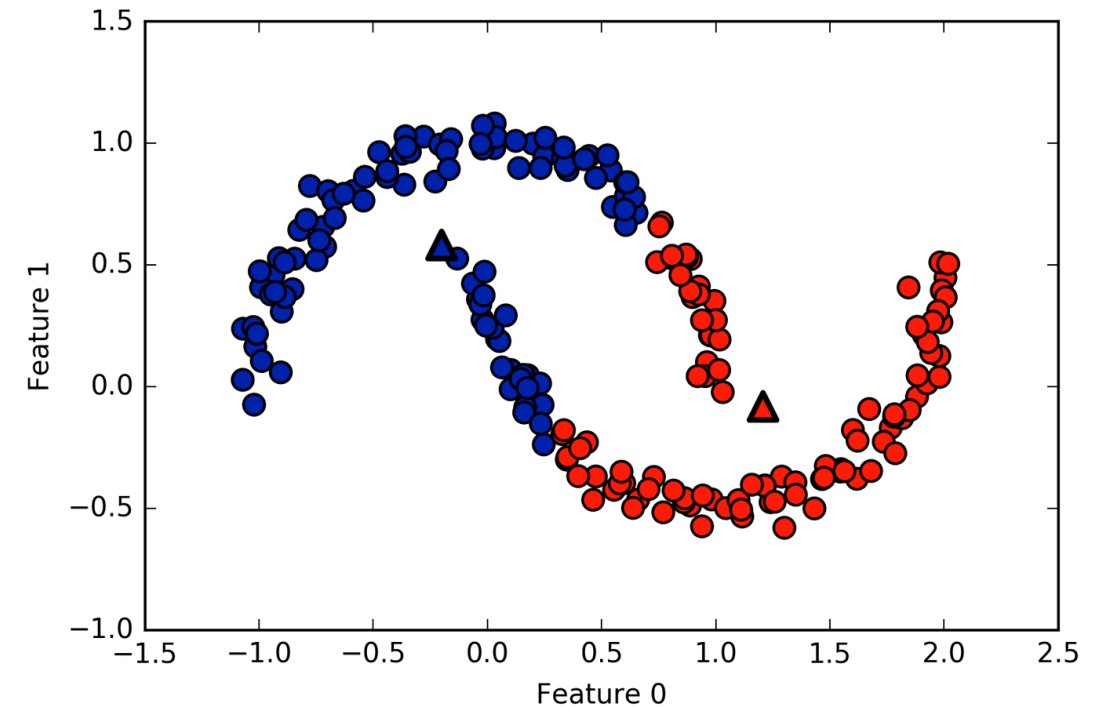


Figure 3-29.  $k$ -means fails to identify clusters with complex shapes

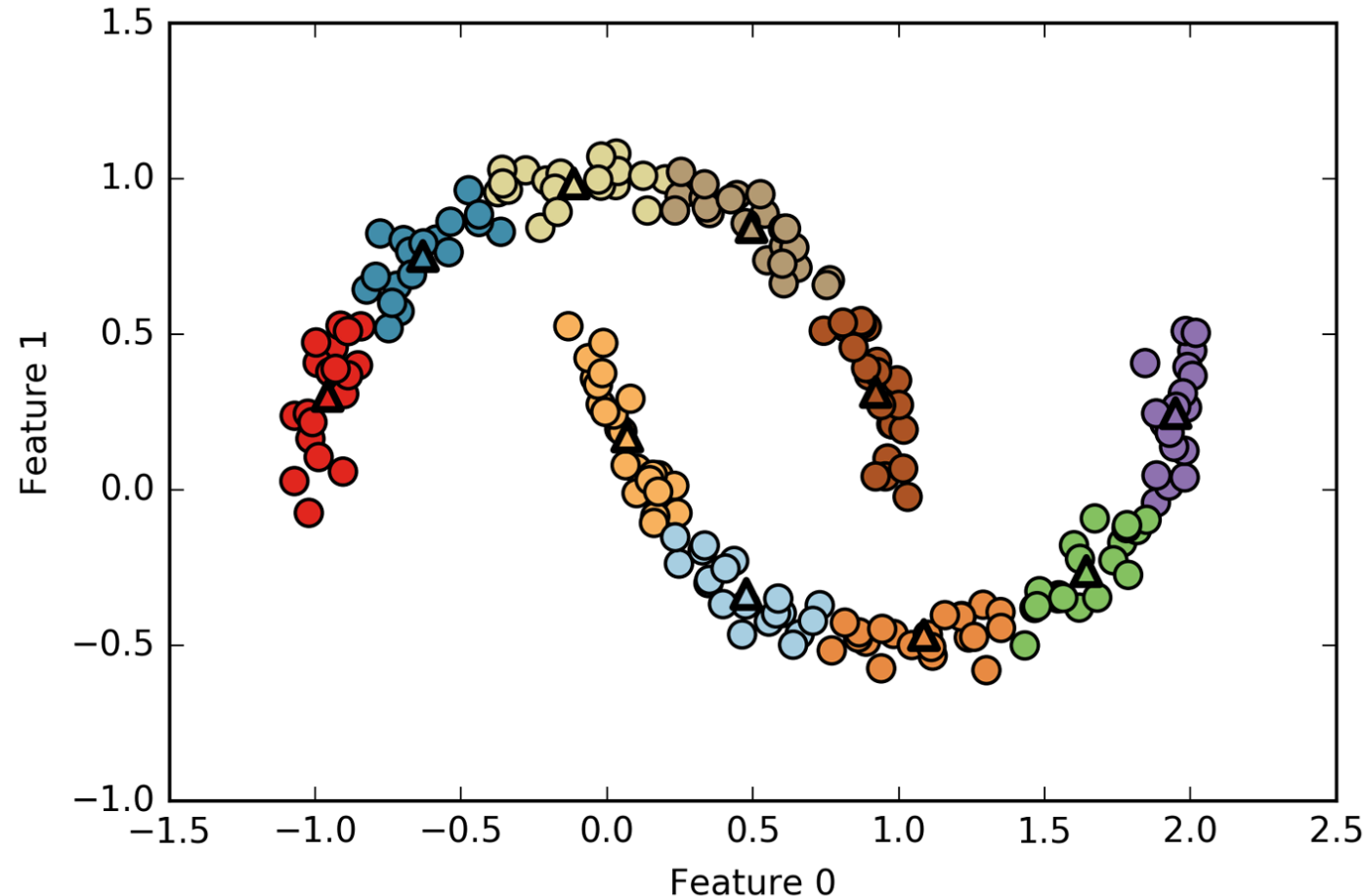


Figure 3-32. Using many  $k$ -means clusters to cover the variation in a complex dataset

# ***k*-Means**

- ▶ **Best Practices**

- Can run in batches on very large datasets

- ▶ **Strengths**

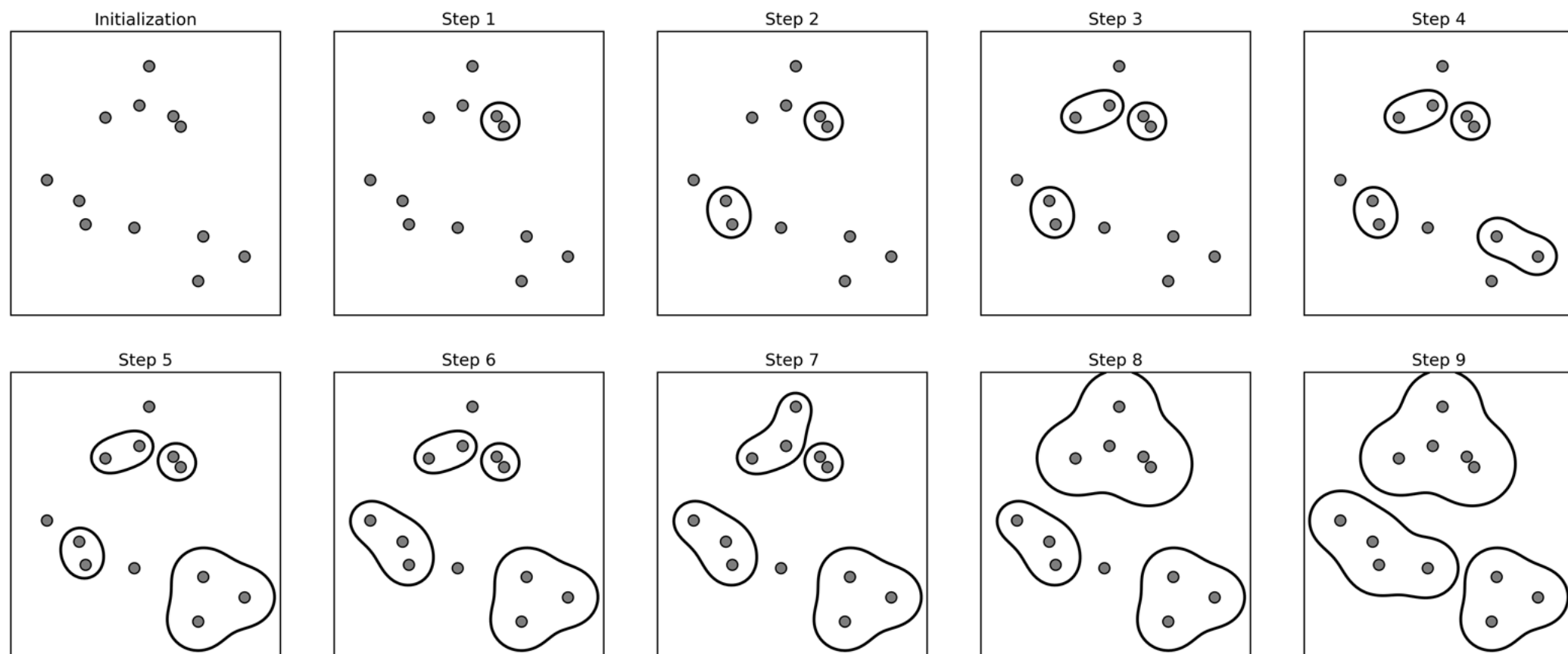
- Easy to understand
- Runs relatively quickly

- ▶ **Weaknesses**

- Based on random initialization
- Need to specify the number of clusters
- Clusters have consistent widths and shapes

# Agglomerative Clustering

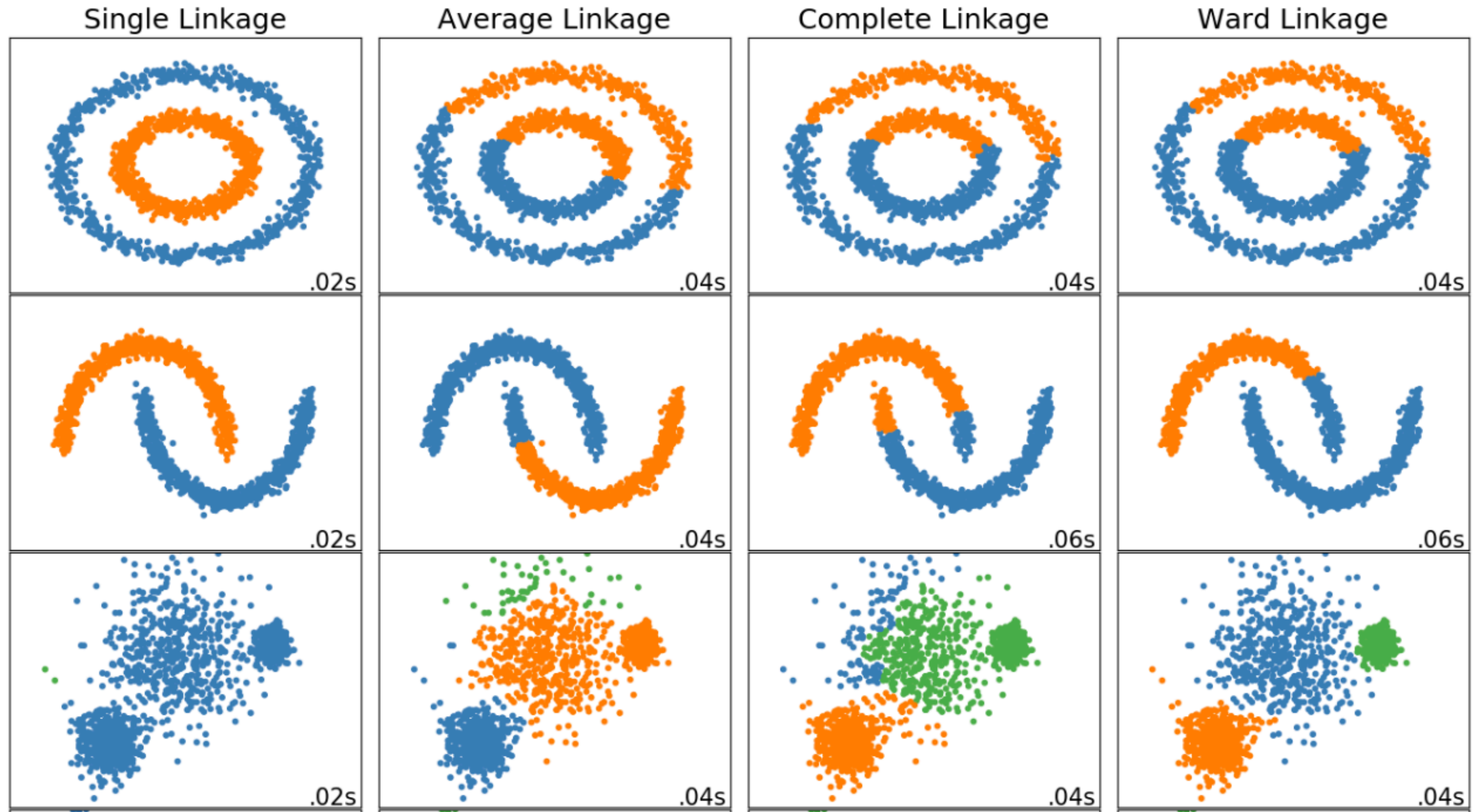
- ▶ **Starts by creating a cluster for each point**
- ▶ **Then amalgamates nearest clusters based on linkage criteria until the stopping criteria is reached**



*Figure 3-33. Agglomerative clustering iteratively joins the two closest clusters*



# Agglomerative Clustering



uses the **minimum** of the distances between all observations of the two sets

uses the **average** of the distances of each observation of the two sets

uses the **maximum** distances between all observations of the two sets

minimizes the **variance** of the clusters being merged

# Agglomerative Clustering

- ▶ **Looking at all possible clusters simultaneously provides information about the hierarchical relationship of the clusters**
- ▶ **Dendrograms allow for visualization of multidimensional datasets, also providing information about cluster distance**

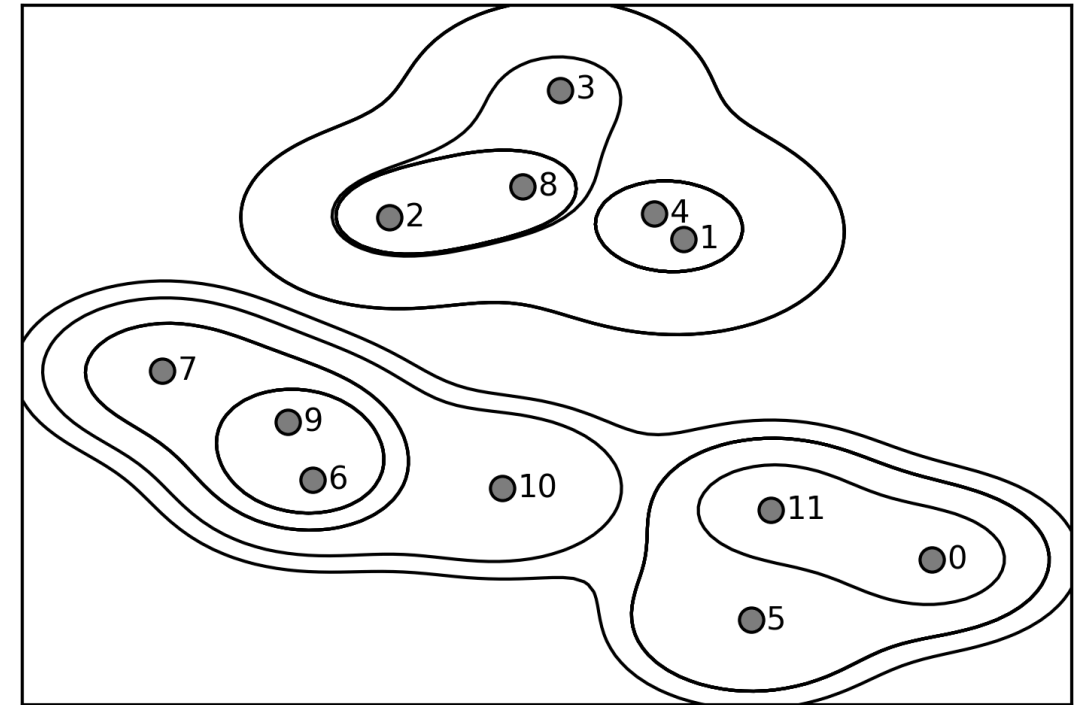


Figure 3-35. Hierarchical cluster assignment (shown as lines) generated with agglomerative clustering, with numbered data points (cf. Figure 3-36)

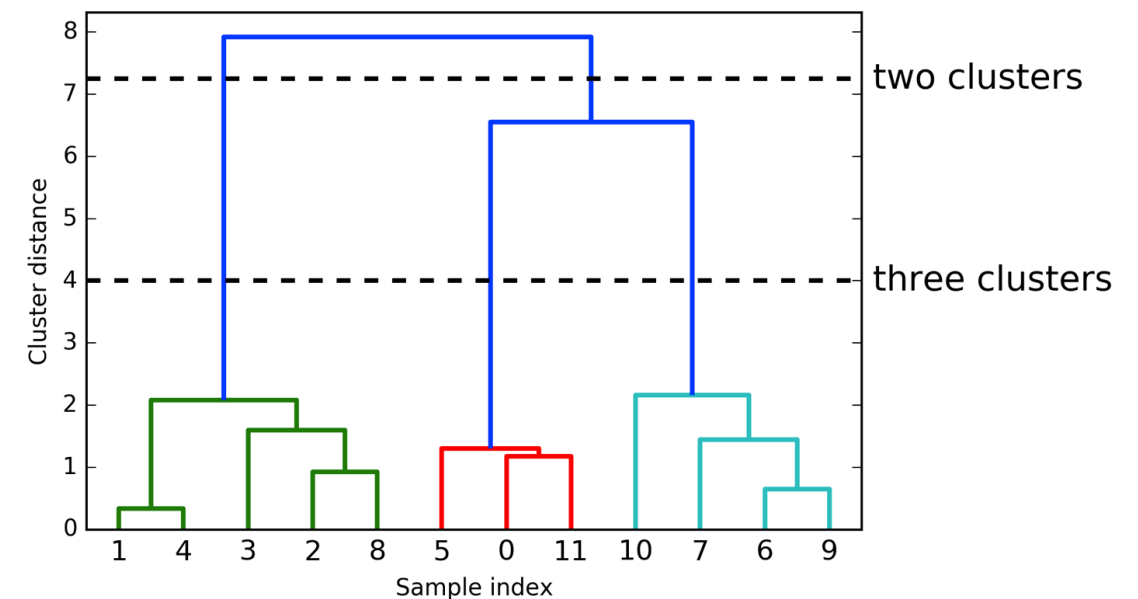


Figure 3-36. Dendrogram of the clustering shown in Figure 3-35 with lines indicating splits into two and three clusters

# Agglomerative Clustering

## ▸ **Parameters**

- Linkage criteria: ward, average, complete
- Stopping criteria: number of clusters

## ▸ **Strengths**

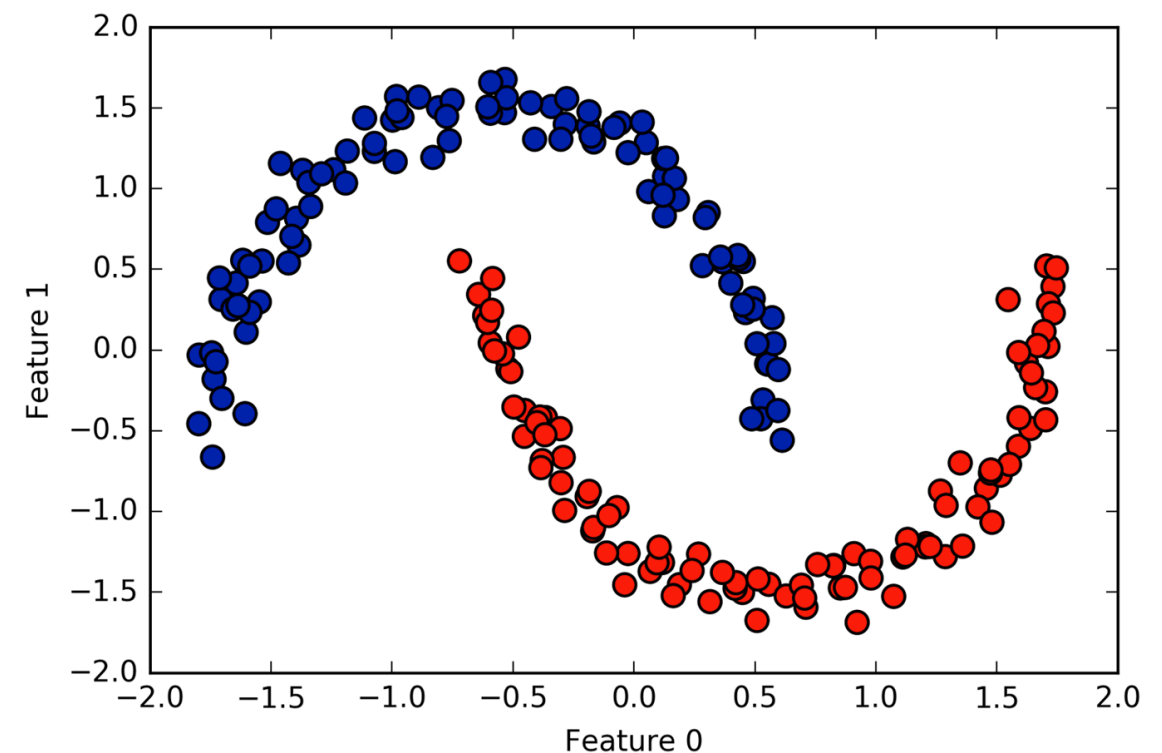
- Easy to understand/visualize

## ▸ **Weaknesses**

- Not able to make prediction on new data
- In scikit-learn you need to specify the number of clusters

# DBSCAN

- ▶ **Density-based spatial clustering of applications with noise**
- ▶ **Do not need to specify the number of clusters**
- ▶ **Attempts to distinguish between densely and sparsely populated areas of the data space**
  - ▶ Core points - cluster centers
  - ▶ Boundary points - within a cluster
  - ▶ Noise



*Figure 3-38. Cluster assignment found by DBSCAN using the default value of  $\epsilon=0.5$*

# DBSCAN

- ▶ **Procedure (repeated until clusterable data has been addressed)**
  - ▶ **Select a data point and check how many other data points are within the specified distance**
  - ▶ **If there are as many as the specified minimum number, data point is considered a core sample**
    - ▶ **Data points within the minimum distance are boundary points**
  - ▶ **If there are multiple core samples within the specified distance, they are merged into a single cluster and their neighbors are also visited**
- ▶ **If points aren't clustered, they are classified as noise**

# DBSCAN

- ▶ Increasing eps results in more points per cluster
- ▶ Increasing min\_samples results in more being classified as noise

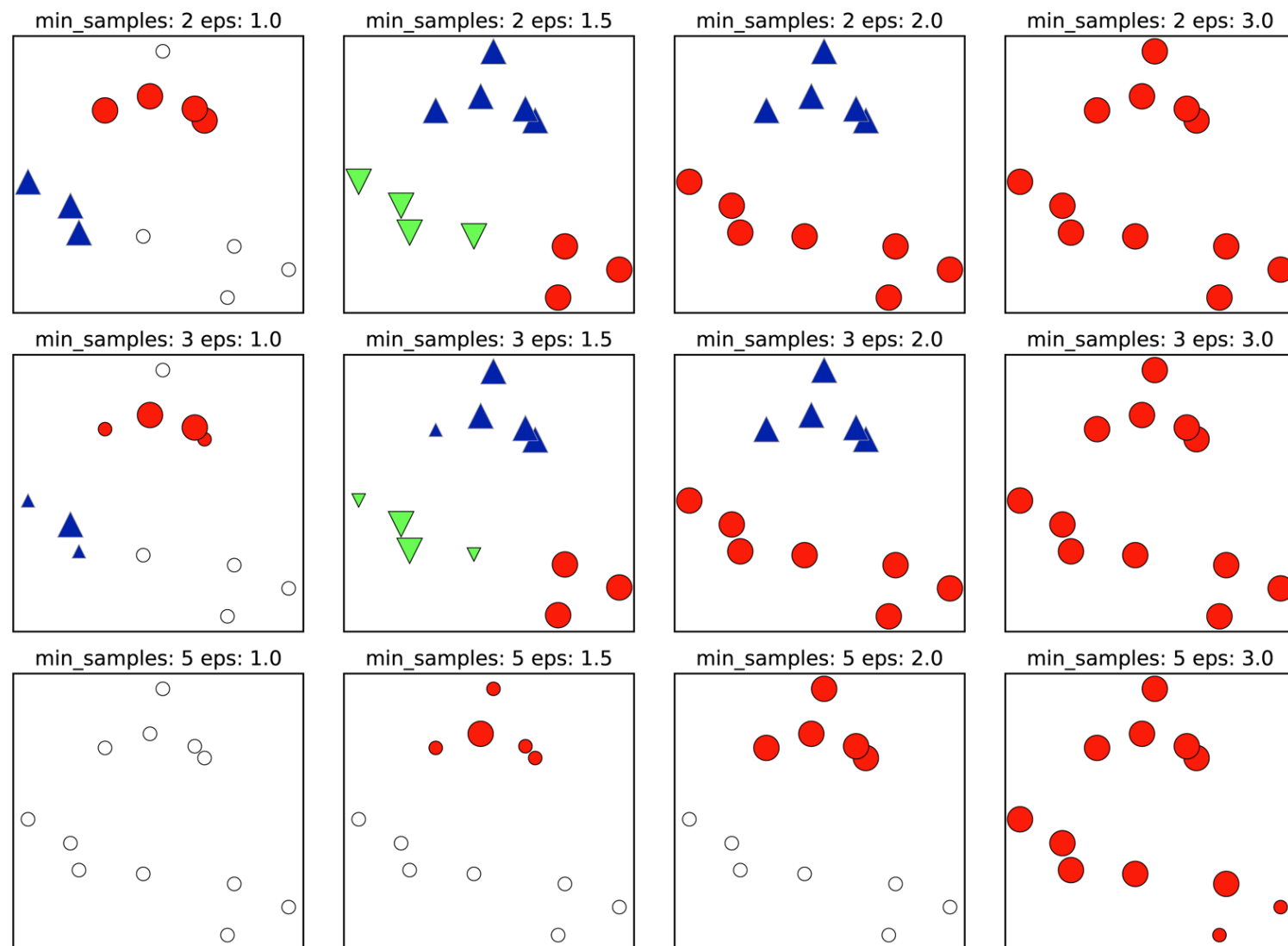


Figure 3-37. Cluster assignments found by DBSCAN with varying settings for the `min_samples` and `eps` parameters

# DBSCAN

## ▸ **Best Practices**

- Scaling data can improve clustering results with DBSCAN

## ▸ **Parameters**

- `eps` - determines distance the algorithm looks for data points
- `min_samples` - determines the minimum number of data points within `eps` distance necessary to form a cluster

## ▸ **Strengths**

- Able to cluster complex shapes

## ▸ **Weaknesses**

- Cluster assignment depends on order the points are visited
- Results sensitive to the settings of `min_samples` and `eps`



# Evaluating Clustering

## With Ground Truth: Adjusted Rand Index (ARI)

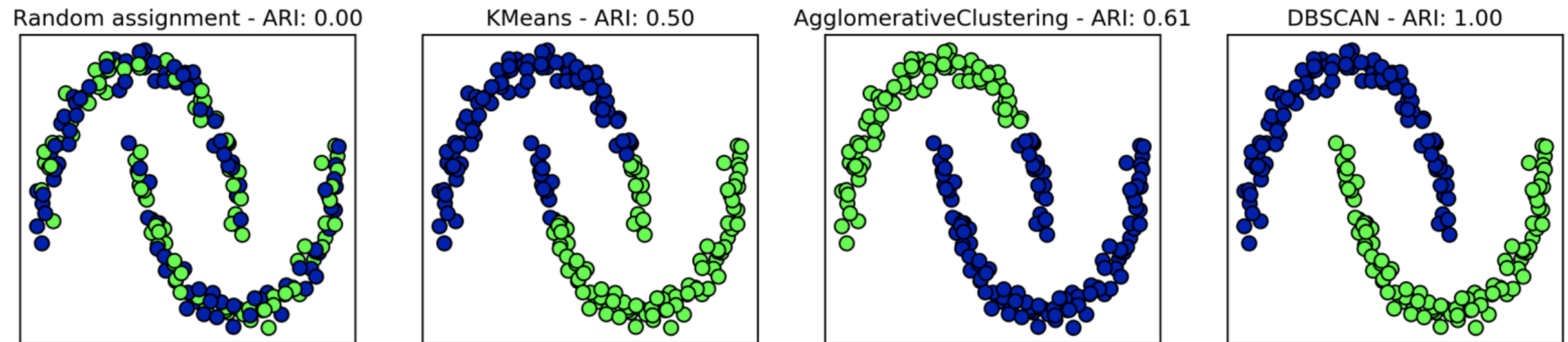


Figure 3-39. Comparing random assignment, k-means, agglomerative clustering, and DBSCAN on the two\_moons dataset using the supervised ARI score

## With No Ground Truth: Silhouette Coefficient

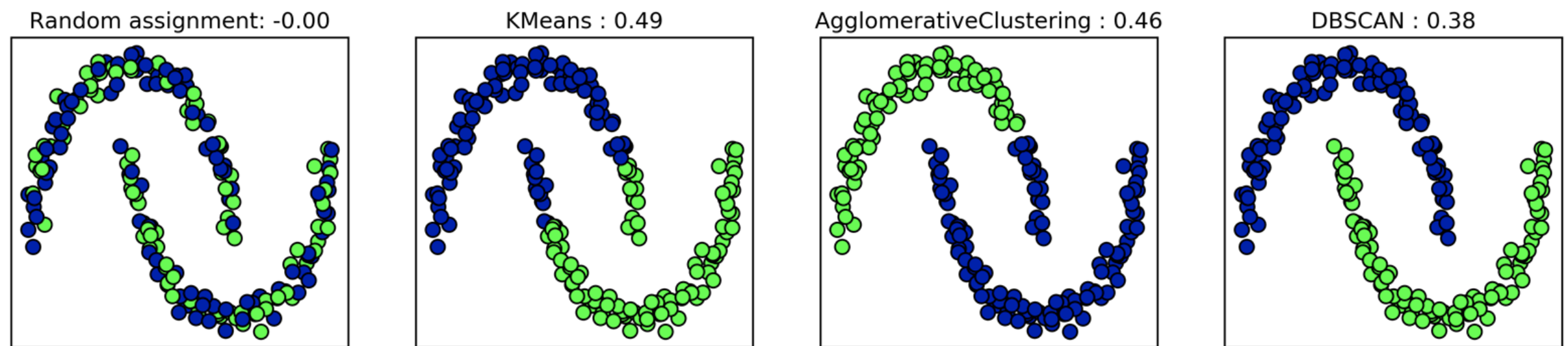


Figure 3-40. Comparing random assignment, k-means, agglomerative clustering, and DBSCAN on the two\_moons dataset using the unsupervised silhouette score—the more intuitive result of DBSCAN has a lower silhouette score than the assignments found by k-means