Machine Learning I (DATS 6202)

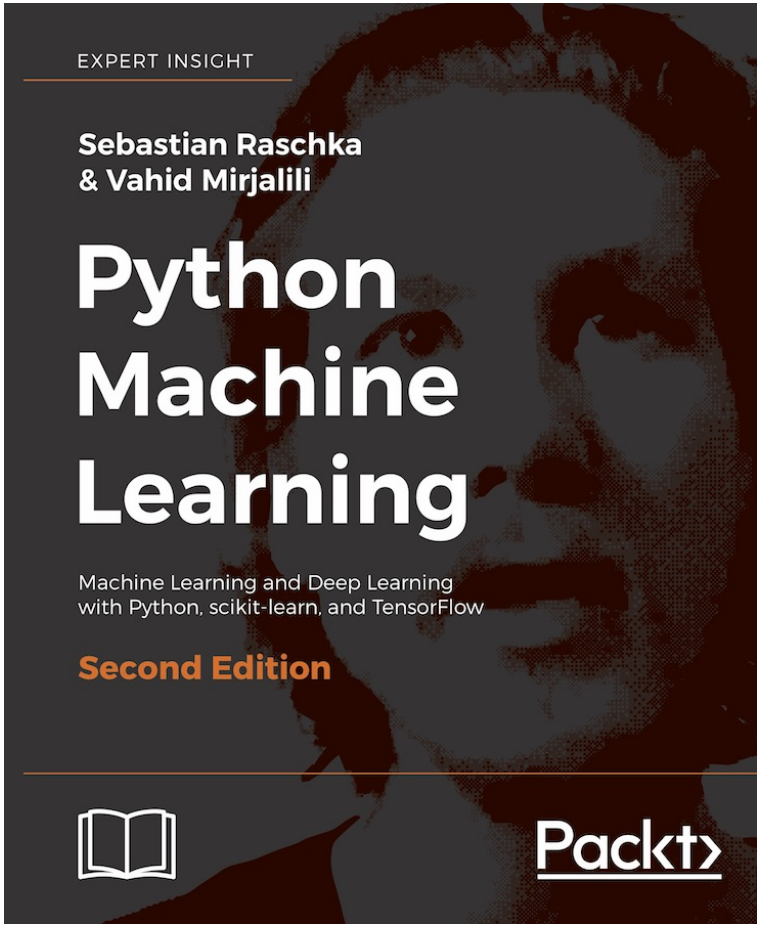
Clustering

Yuxiao Huang

Data Science, Columbian College of Arts & Sciences
George Washington University
yuxiaohuang@gwu.edu

November 12, 2018

Reference



Picture courtesy of the website of the book code repository and info resource

Reference

- This set of slices is an excerpt of the book by Raschka and Mirjalili, with some trivial changes by the creator of the slides
- Please find the reference to and website of the book below:
  - *Raschka S. and Mirjalili V. (2017). Python Machine Learning. 2nd Edition.*
  - `https://sebastianraschka.com/books.html`
- Please find the website of the book code repository and info resource below:
  - `https://github.com/rasbt/python-machine-learning-book-2nd-edition`

Overview

1. Clustering
2. Prototype-based clustering
3. Hierarchical clustering
4. Density-based clustering

Supervised learning VS unsupervised learning

- To this point we have been discussing supervised learning, which
  - requires data where the target value is known
  - aims to predict the target value of a new sample
- Supervised learning can be divided into two groups
  - regression: when the value is continuous
  - classification: when the value is discrete
- A different approach of learning, unsupervised learning,
  - allows data where the target value is unknown
  - aims to detect the hidden structure in the data

Clustering

- As one type of unsupervised learning, clustering analysis divides data into clusters (groups), where data in the same cluster are more similar to each other than to those from different clusters
- There are three types of clustering:
  - prototype-based clustering
  - hierarchical clustering
  - density-based clustering

Prototype-based clustering

- In Prototype-based clustering, each cluster is represented by a prototype, which can be
  - the centroid (mean) of continuous data
  - the medoid (mode) of discrete data

K-means

- As the most popular algorithm in Prototype-based clustering, k-means divides data into clusters based on the following steps
  1. randomly pick k centroids from the sample points as initial cluster centers
  2. assign each sample to the nearest centroid $\mu^{(j)}, \quad j \in 1, ..., k$
  3. move the centroids to the center of the samples that were assigned to it
  4. repeat steps 2 and 3 until the cluster assignments do not change or a user-defined tolerance or a maximum number of iterations is reached

Measuring the distance

- The distance between two points $\mathbf{x}$ and $\mathbf{y}$ in $m-$dimensional space is commonly measured by the squared Euclidean distance

$$d(\mathbf{x}, \mathbf{y})^2 = \sum_{j=1}^{m} \left(x_j - y_j\right)^2 = \|\mathbf{x} - \mathbf{y}\|_2^2.$$

The objective function

- **Q:** Based on the squared Euclidean distance, what could be the objective function for k-means?

The objective function

- **Q:** Based on the squared Euclidean distance, what could be the objective function for k-means?
- **A:** The within-cluster Sum of Squared Errors (SSE) across all clusters

$$J(w) = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)} \left\| \mathbf{x}^{(i)} - \mu^{(j)} \right\|_2^2$$

where:

- $\mu^{(j)}$ is the centroid for cluster j
- $w^{(i,j)} = 1$ if the sample $\mathbf{x}^{(i)}$ is in cluster j
- $w^{(i,j)} = 0$ otherwise

The problem

- **Q:** Can you name some of the problems of k-means?

The problem

- **Q:** Can you name some of the problems of k-means?
- **A:** Here are some:
  - the initial centroids are placed randomly
  - the number of clusters, k, must be specified
  - the clusters cannot overlap and are not hierarchical
  - there is at least one point in each cluster

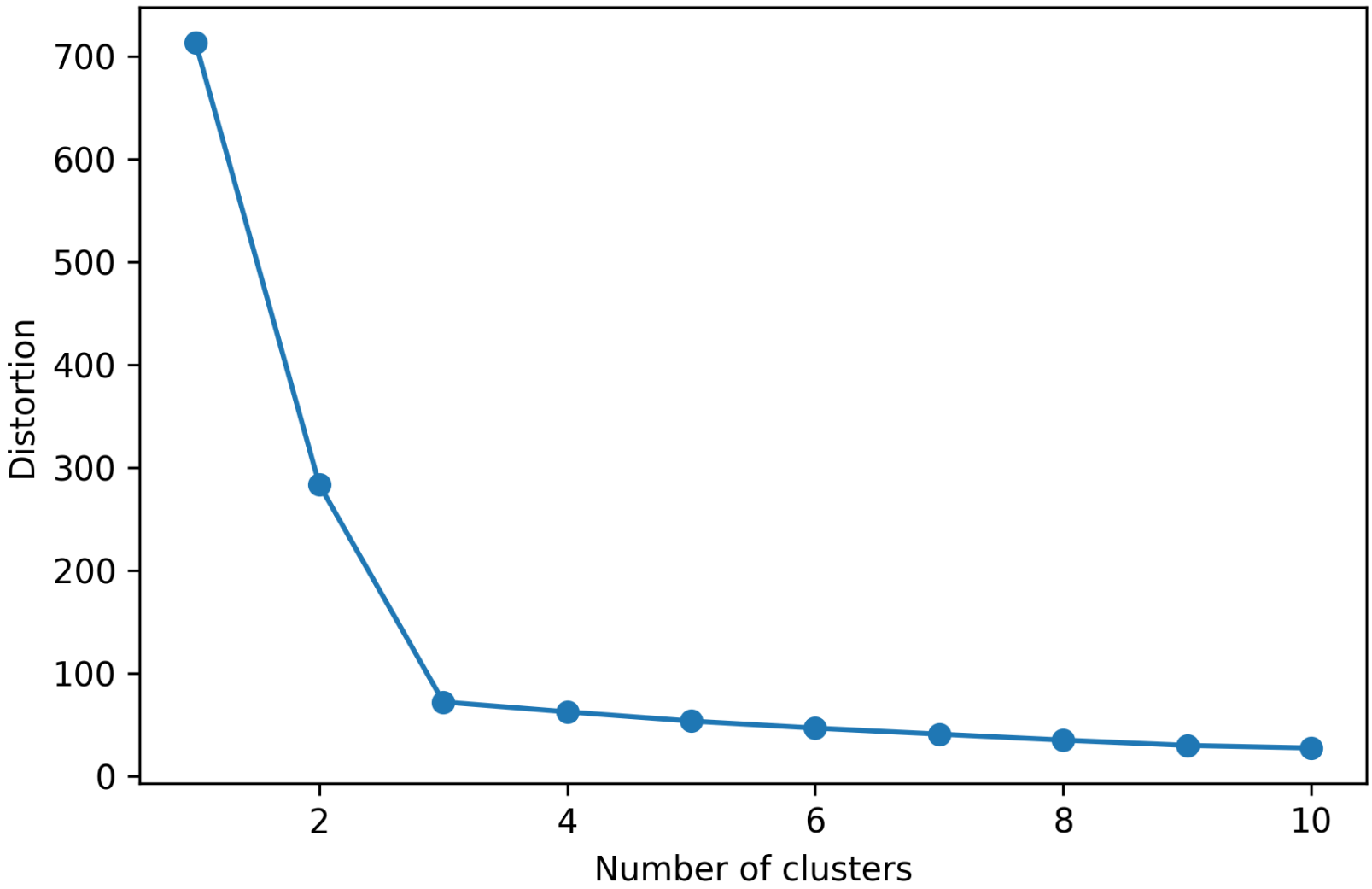
K-means++

- K-means++ was proposed to better place the initial centroids based on the following steps
  1. initialize an empty set M to store the k centroids being selected
  2. randomly choose the first centroid $\mu^{(j)}$ from the input samples and assign it to M
  3. for each sample $\mathbf{x}^{(i)}$ that is not in M, find the minimum distance $d\big(x^{(i)}, M\big)^2$ to any of the centroids in M
  4. to randomly select the next centroid $\mu^{(p)}$, use a weighted probability distribution equal to
  $$\frac{d(\mu^{(p)}, M)^2}{\sum_i d(x^{(i)}, M)^2}$$
  5. repeat steps 2 and 3 until k centroids are chosen
  6. proceed with the classic *k*-means algorithm

The elbow method

- The elbow method aims to find a good estimation of the number of clusters, k, based on the within-cluster SSE (distortion)
- The idea is that, when k increases SSE decreases (since the samples are closer to the corresponding centroids)
- Thus a good estimation of k is the one where the distortion begins to increase most rapidly
- This is illustrated in Figure 1 (see next page)

Figure 1


700
600
500
400
300
200
100
0
Distortion
2
4
6
8
10
Number of clusters

The silhouette plots

- Another metric to measure the performance of a clustering is silhouette analysis
- The idea is to show how tight the samples in the clusters are
- The silhouette coefficient of a single sample can be calculated as:
  1. calculate the cluster cohesion $a^{(i)}$ as the average distance between a sample $\mathbf{x}^{(i)}$ and all other points in the same cluster.
  2. calculate the cluster separation $b^{(i)}$ from the next closest cluster as the average distance between the sample $\mathbf{x}^{(i)}$ and all samples in the nearest cluster.
  3. calculate the silhouette $s^{(i)}$ as the difference between cluster cohesion and separation divided by the greater of the two:

$$s^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max\{b^{(i)}, a^{(i)}\}}.$$

- Examples are illustrated in Figures 2 and 3 (see next two pages)

Figure 2

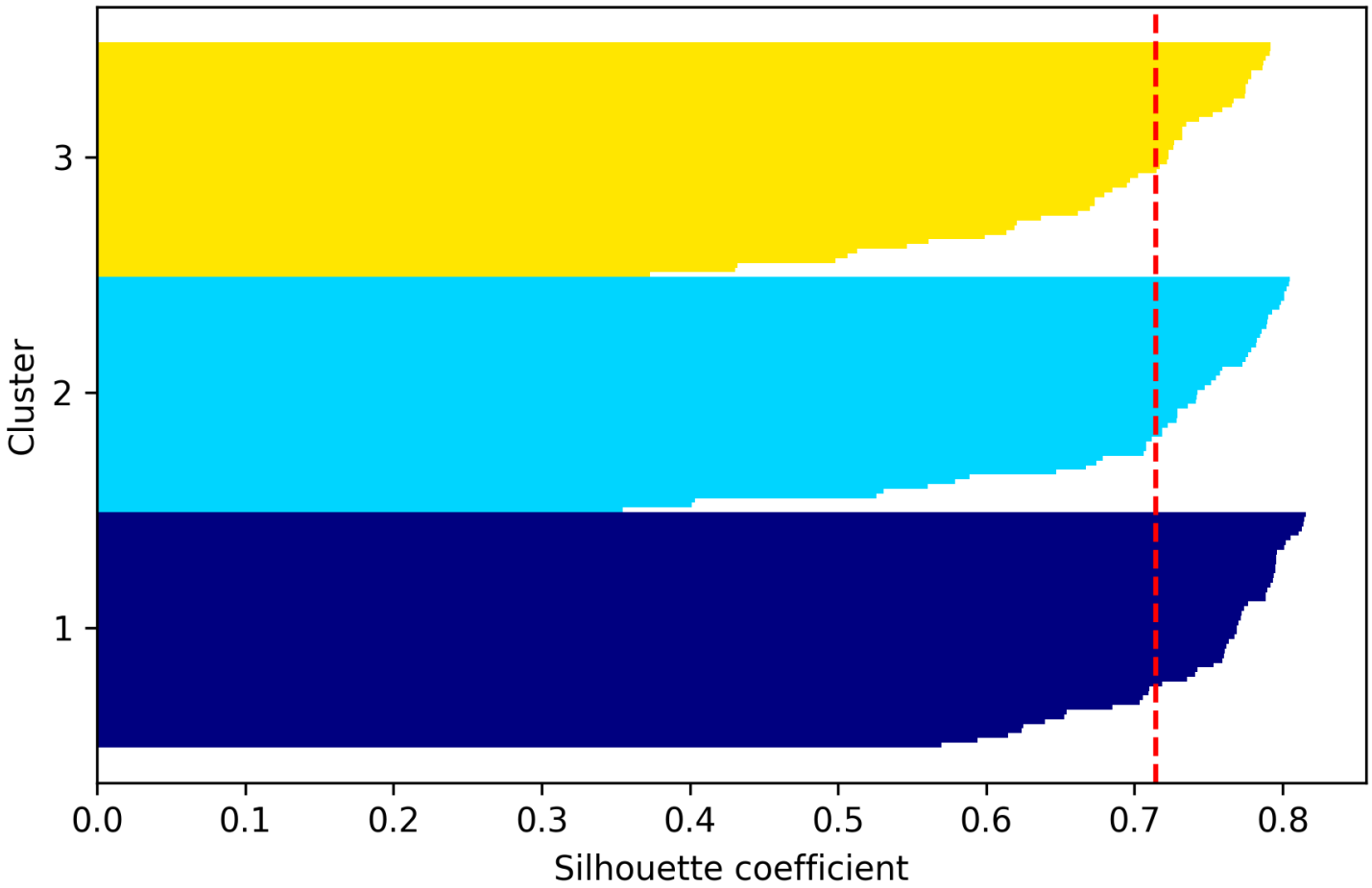
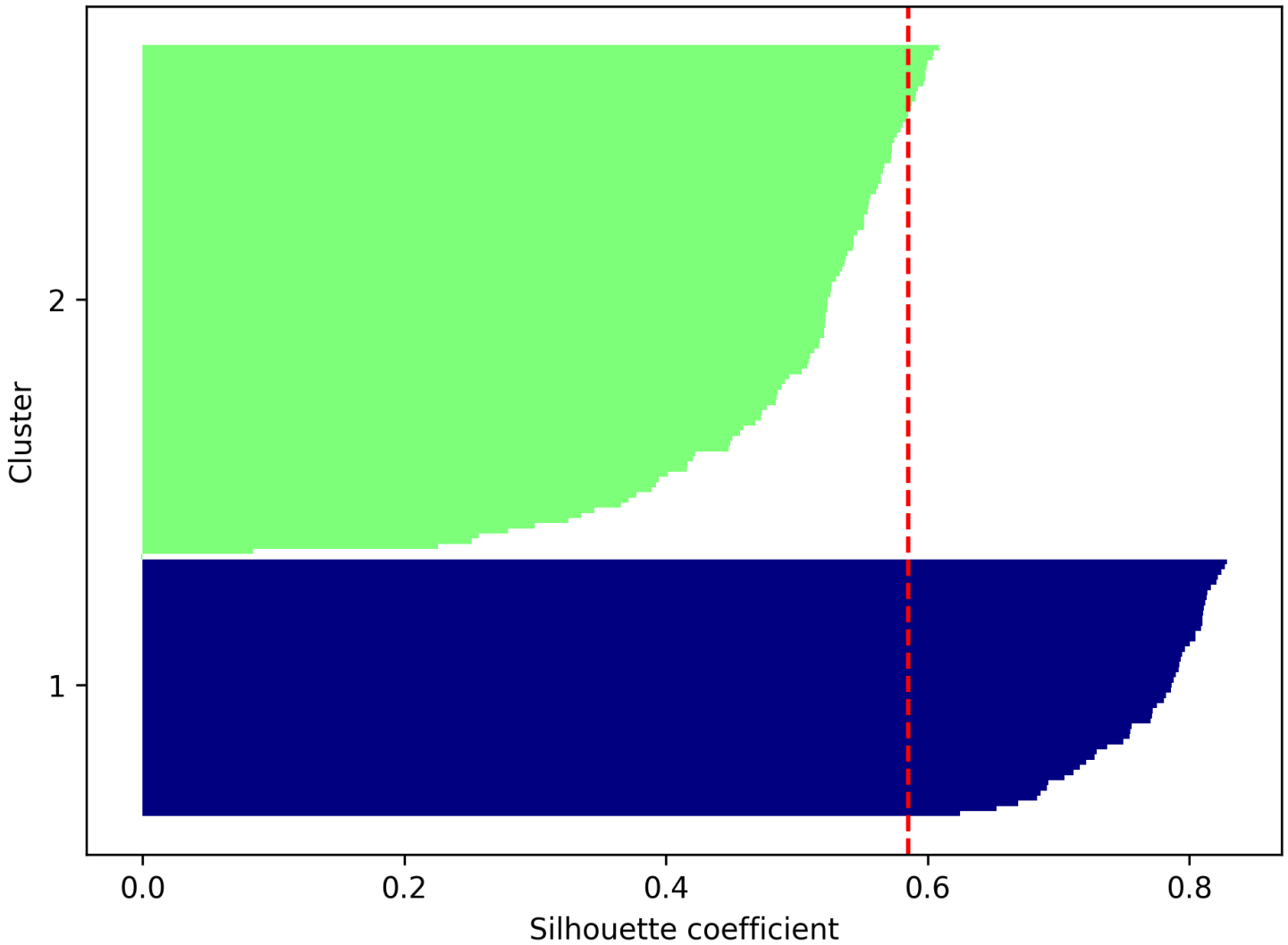


Figure 3



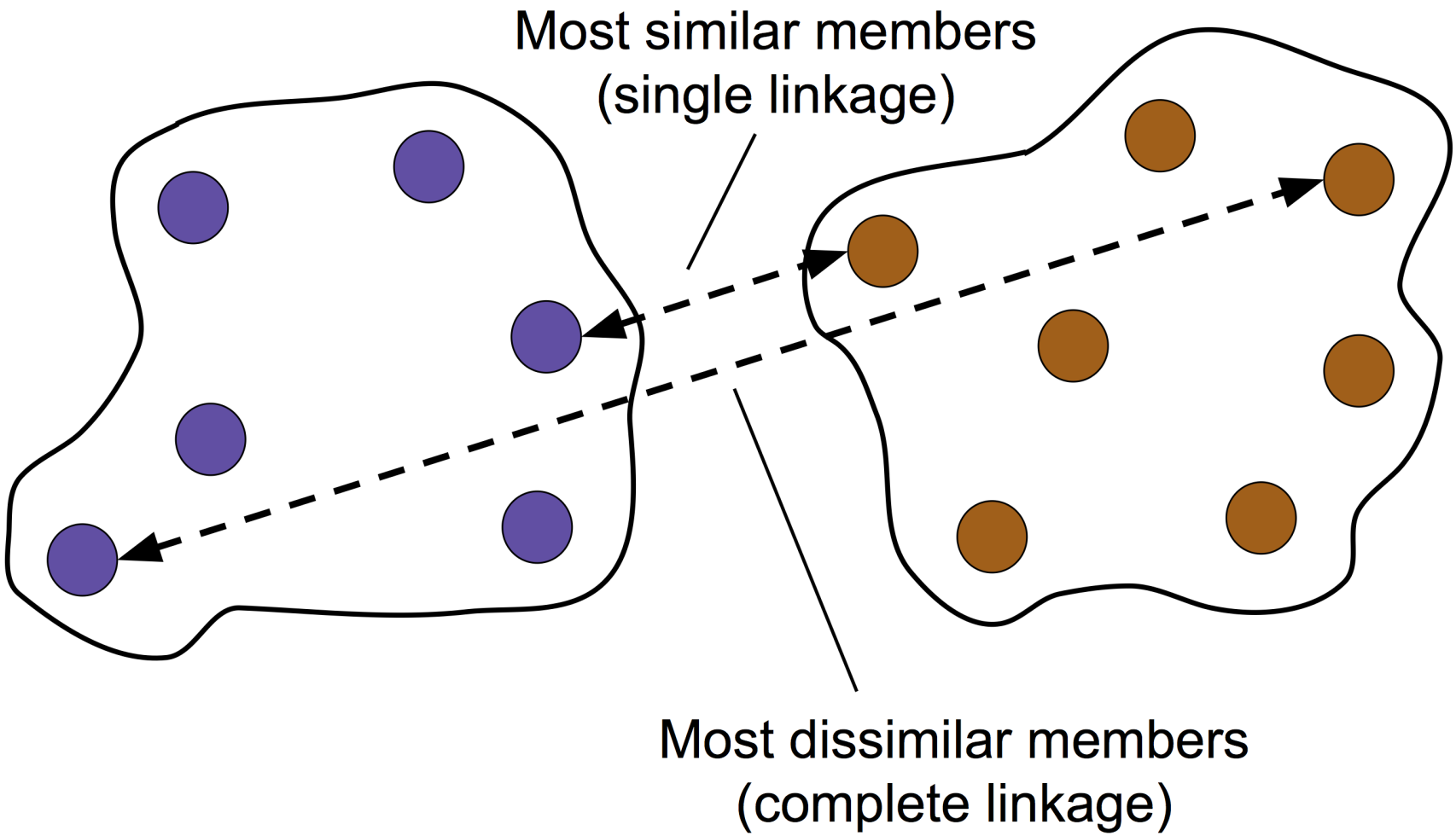
Hierarchical clustering

- Compared to prototype-based clustering, hierarchical clustering has two advantages
  - it does not require specifying the nubmer of clusters, k
  - it can plot dendrograms, which facilitate interpreting the results by creating meaningful taxonomies
- Hierarchical clustering can be divided into two categories
  - divisive (top-down): start with one cluster (of all samples) and iteratively split the clusters into smaller ones, until each cluster contains only one sample
  - agglomerative (bottom-up): the opposite

Agglomerative hierarchical clustering

- The idea is that, in each round we merge the two clusters that are the most similar
- There are two standard ways to measure the similarity between clusters
  - single linkage: the similarity is determined by the **shortest** distance between the two clusters
  - complete linkage: the similarity is determined by the **longest** distance between the two clusters
- The two measurements are shown in Figure 4 (see next page)

Figure 4


Most similar members
(single linkage)
Most dissimilar members
(complete linkage)

Hierarchical complete linkage clustering

- Hierarchical complete linkage clustering has the following steps
  1. compute the distance matrix of all samples
  2. represent each data point as a singleton cluster
  3. merge the two closest clusters based on the distance between the most dissimilar (distant) members
  4. update the similarity matrix
  5. repeat steps 2-4 until one single cluster remains

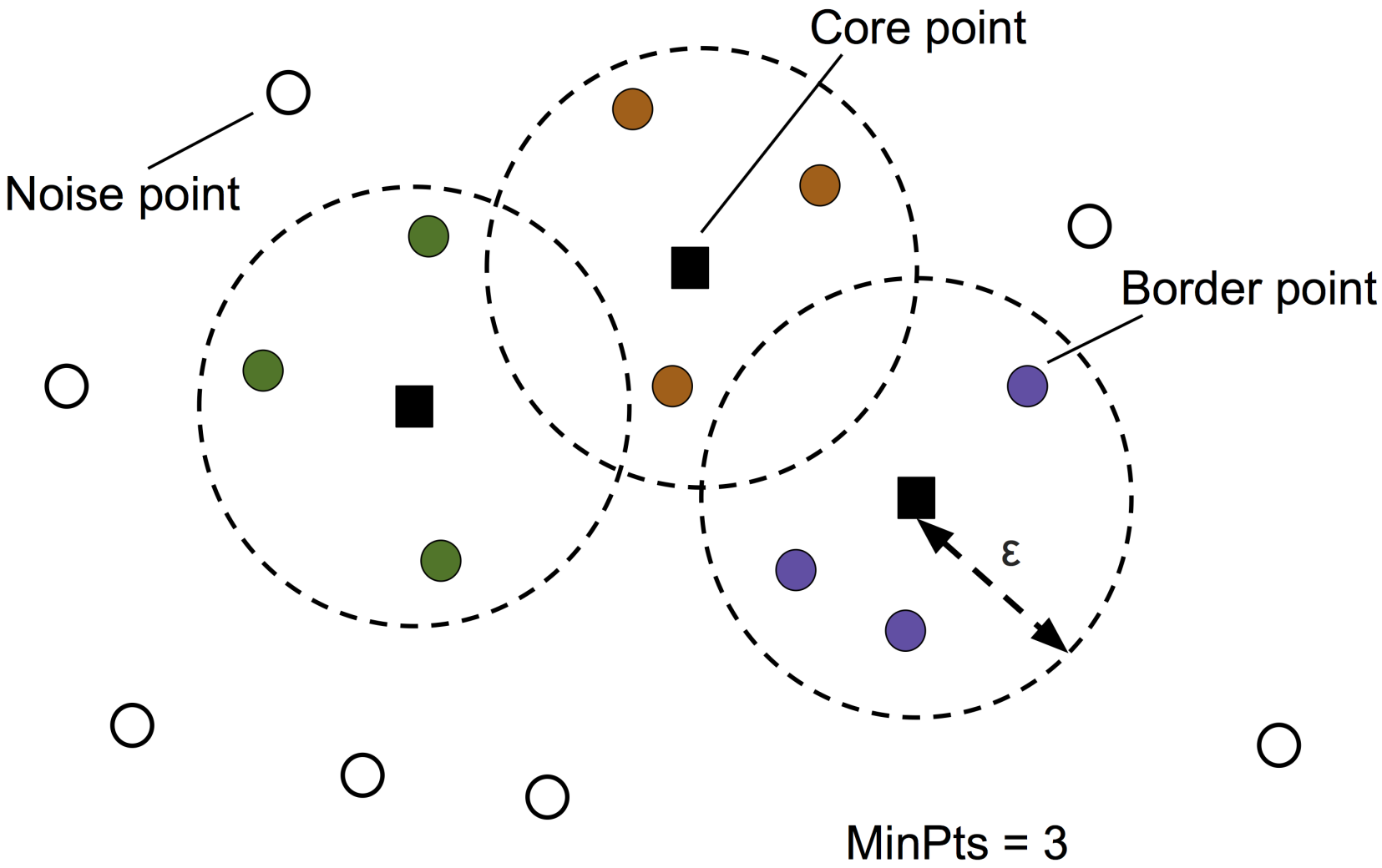
DBSCAN

- Density-based Spatial Clustering of Applications with Noise (DBSCAN)
  - does not assume spherical clusters as k-means
  - does not require a cut-off point needed in hierarchical clustering
  - assigns cluster labels based on dense regions of samples

Core point, border point, and noise point

- A point is considered as **core point** if at least a specified number (MinPts) of neighboring points fall within the specified radius ϵ.
- A **border point** is a point that has fewer neighbors than MinPts within ϵ, but lies within the ϵ radius of a core point.
- All other points that are neither core nor border points are considered as **noise points**.
- This is illustrated in Figure 5

Figure 5



The two steps

- DBSCAN has two steps:
  1. Form a separate cluster for each core point or a connected group of core points (core points are connected if they are no farther away than ϵ).
  2. Assign each border point to the cluster of its corresponding core point.
- Figures 6 to 8 illustrate half-moon like data and the difference in the performance between k-means, agglomerative clustering and DBSCAN

Figure 6

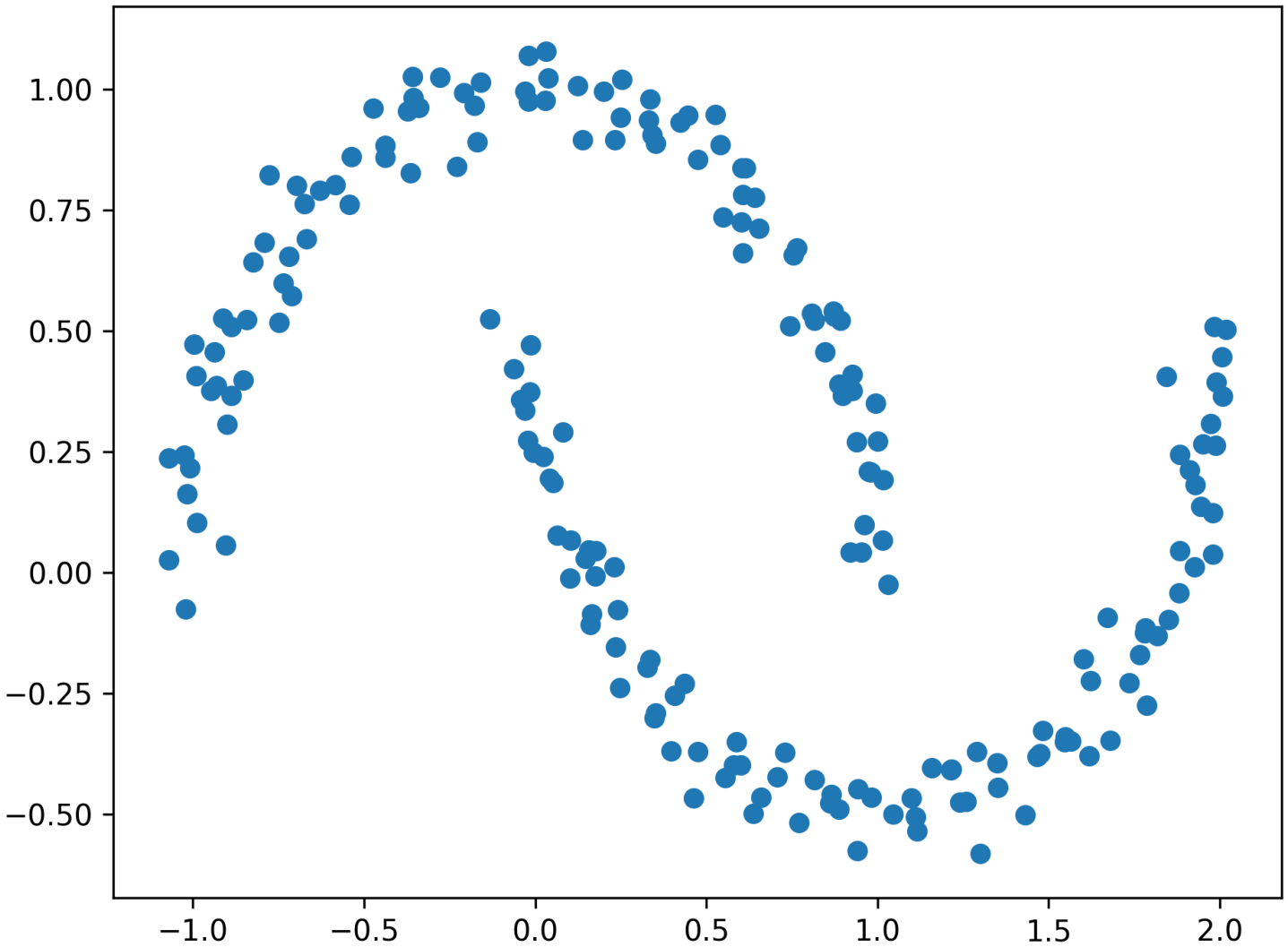


Figure 7

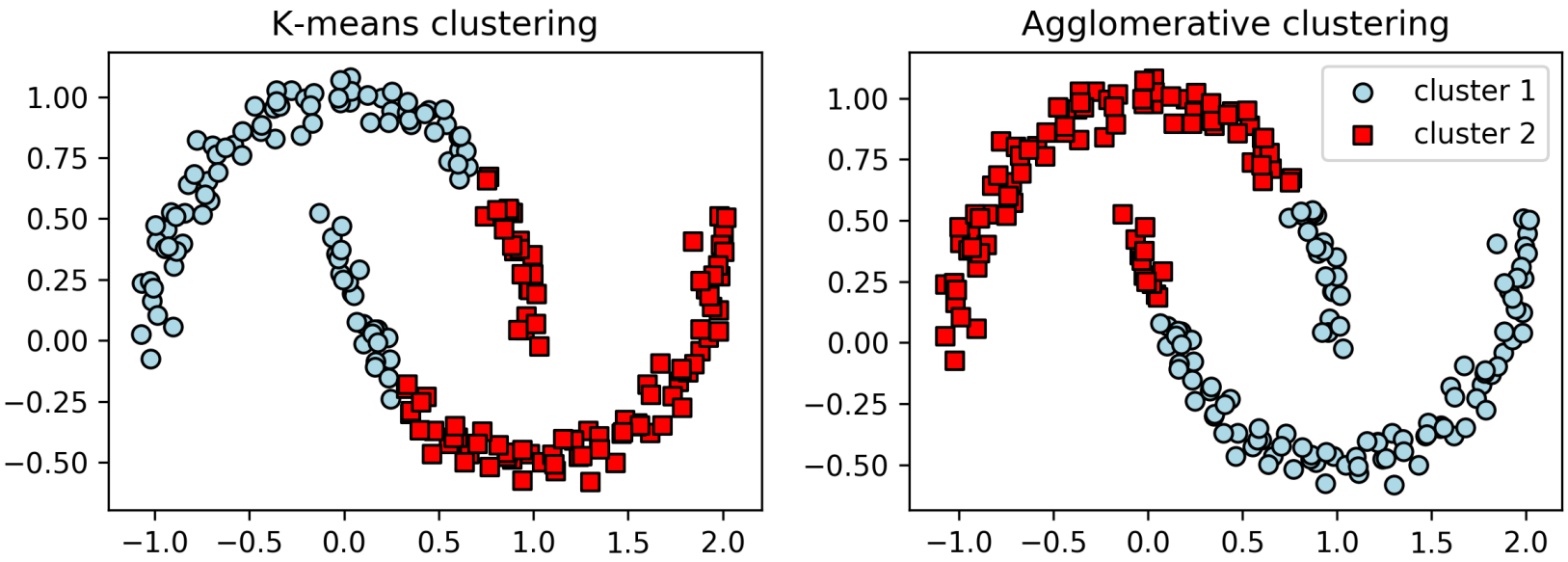
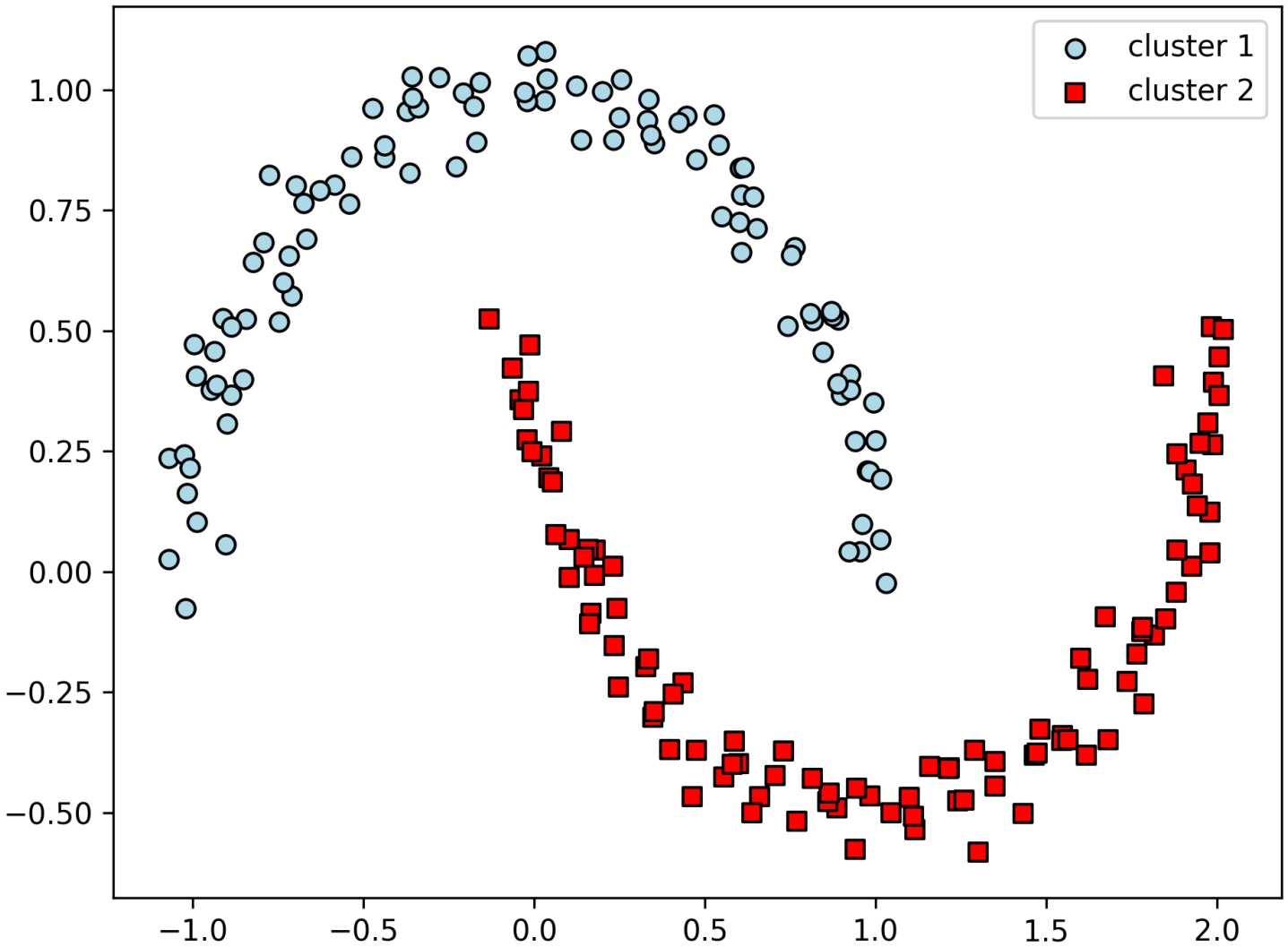

K-means clustering
Agglomerative clustering
cluster 1
cluster 2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Figure 8


cluster 1
cluster 2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0