December 12, 2025

Inter-University Research Institute Corporation /
Research Organization of Information and Systems
National Institute of Informatics

Clustering

Data Mining 07 (データマイニング)

Mahito Sugiyama (杉山麿人)

Today's Outline

- Clustering methods will be introduced
- K-means, EM algorithm, DBSCAN, hierarchical clustering
- Evaluation of clustering results

Clustering

- **Goal:** Partition objects into several groups, where those in the same group are similar with each other
  - A typical problem in unsupervised learning
- Given a dataset $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}, \quad \boldsymbol{x}_i \in \mathbb{R}^d$
- Clustering: Find a partition $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$ of D s.t.

$$\bigcup_{i \in \{1,2,\dots,K\}} C_i = D \text{ and } C_i \cap C_j = \emptyset$$

  - Each $C_i \subseteq D$ is called a cluster

K-means

- *K*-means is one of the most heavily used algorithm
- The sum of squared errors scoring function:

$$\mathrm{SSE}(\mathcal{C}) = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2 = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} \sum_{j=1}^{d} (x^j - \mu_k^j)^2$$

  - $\boldsymbol{\mu}_k$ is the mean vector of a cluster C_k
  - Dissimilarity is measured by the squared Euclidean distance
- *K*-means tries to find the optimal clustering $\mathcal{C}^*$ s.t.

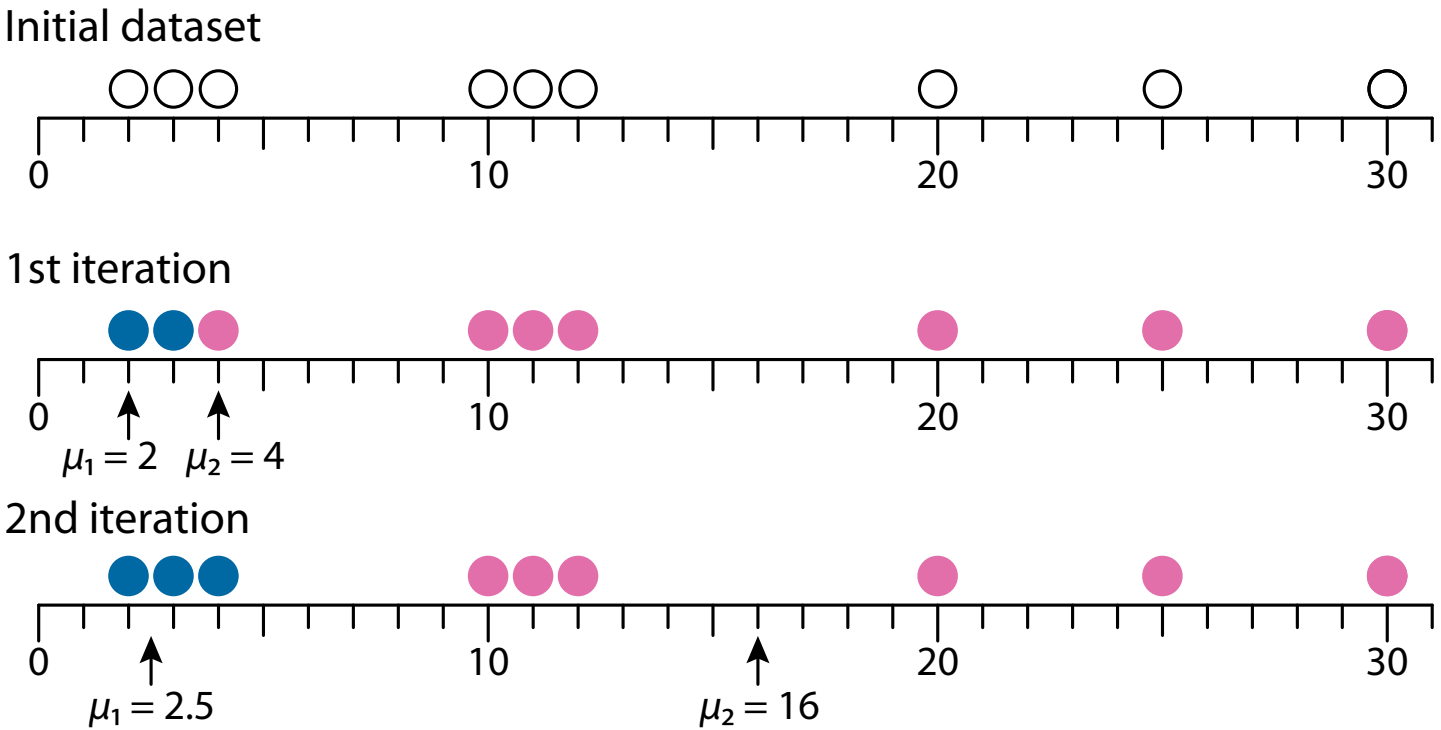
$$\mathcal{C}^* = \underset{\mathcal{C}}{\operatorname{argmin}}\, \mathrm{SSE}(\mathcal{C})$$

Pseudocode of *K*-means

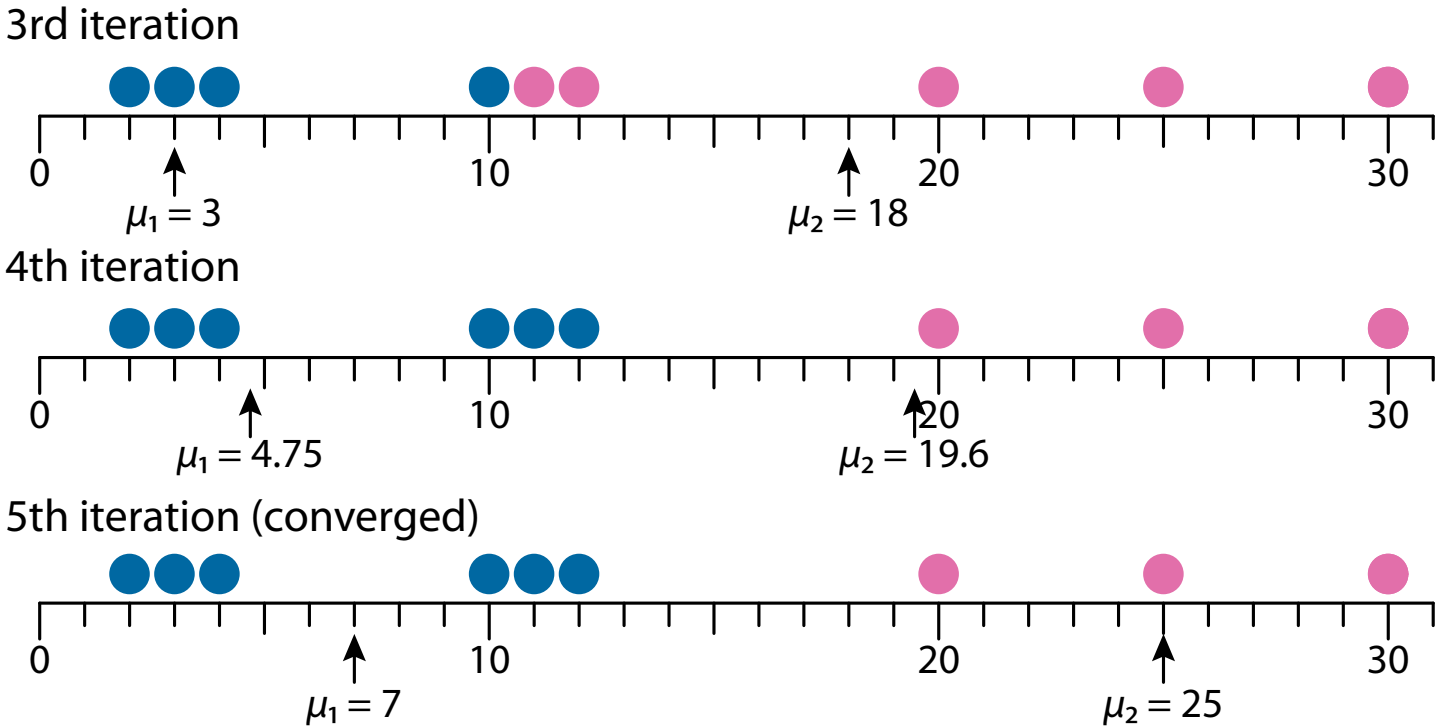
- **Input:** Dataset D, Number of clusters K
- **Output:** Clustering $\mathcal{C}$

1. Randomly initialize K centroids: $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
2. **repeat**
3. $\quad C_k \leftarrow \emptyset$ for all $k \in \{1, 2, \ldots, K\}$
4. $\quad$ **for each** $\boldsymbol{x} \in D$ **do** // cluster assignment
5. $\quad\quad k^* \leftarrow \operatorname{argmin}_{k \in \{1,2,\ldots,K\}} \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2$
6. $\quad\quad C_{k^*} \leftarrow C_{k^*} \cup \{\boldsymbol{x}\}$
7. $\quad$ **for each** $k \in \{1, 2, \ldots, K\}$ **do** // centroid update
8. $\quad\quad \boldsymbol{\mu}_k \leftarrow (1/|C_k|) \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}$
9. **until** cluster assignment does not change

K-means on 1-Dimensional Data



K-means on 1-Dimensional Data



Notes on *K*-means

- *K*-means is a classic algorithm (proposed in 1967!), while is still the state-of-the-art
  - It is fast; its time complexity is $O(ndK)$
  - Easy to use; there is only one parameter K
- Drawbacks
  - Its result may be a local optimum, not global
  - Its result depends on initialization
  - It cannot detect non-spherical clusters

K-means++

- K-means++ is an algorithm for selecting initial clustering
  - This can alleviate the problem of finding worse clustering than optimal

1. Randomly select a data point $\boldsymbol{x} \in D$ and $\boldsymbol{\mu}_1 \leftarrow \boldsymbol{x}$
2. **for each** $k = \{2, 3, \dots, K\}$ **do**
3. **for each** $\boldsymbol{x} \in D$ **do** $D(\boldsymbol{x}) \leftarrow \min_{i \in \{1,2,\dots,k-1\}} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2$
4. **for each** $\boldsymbol{x} \in D$ **do** $p(\boldsymbol{x}) \leftarrow D(\boldsymbol{x}) / \sum_{\boldsymbol{s} \in D} D(\boldsymbol{s})$
5. Select $\boldsymbol{\mu}_k$ from D using $p(\boldsymbol{x})$ as probability distribution
6. Perform K-means using $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$ as the initial cluster centers

EM Clustering

- In K-means, each point either belongs to a cluster or not
  $\rightarrow$ hard clustering
- How about obtaining the probability of cluster membership?
  $\rightarrow$ soft clustering
- The EM (Expectation-Maximization) clustering with a mixture of Gaussian distributions is the representative method
  - It is sometimes called soft K-means

The General EM Algorithm (1/2)

- **Input:** A joint distribution $p(X, Y; \theta)$ over observed variables X and hidden (latent) variables Y, with parameters θ
  Goal: Maximize the likelihood of $p(X; \theta)$
- This is difficult as the marginal distribution

  $$\log p(X; \theta) = \log \left(\sum_{Y} p(X, Y; \theta) \right)$$

  should be optimized

The General EM Algorithm (2/2)

- **Input:** A joint distribution $p(X, Y; \theta)$ over observed variables X and hidden (latent) variables Y, with parameters θ
  Goal: Maximize the likelihood of $p(X; \theta)$ (may be local optimum)

1. Set an initial parameter $\theta^{(t)}$ with $t = 0$
2. **Expectation step (E-step):** Evaluate $p(Y \mid X; \theta^{(t)})$
3. **Maximization step (M-step):** Evaluate $\theta^{(t+1)}$ such that
   $\theta^{(t+1)} = \operatorname{argmax}_{\theta^{(t+1)}} Q(\theta^{(t+1)}, \theta^{(t)})$
   - $Q(\theta^{(t+1)}, \theta^{(t)}) = \sum_Y p(Y \mid X; \theta^{(t)}) \log p(X, Y; \theta^{(t+1)})$
4. $\theta^{(t+1)} \leftarrow \theta^{(t)}$, $t \leftarrow t + 1$ and repeat until convergence

Multivariate Normal Distribution

- Probability density function of 1D normal distribution

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

  - $\mu \in \mathbb{R}$: mean, $\sigma^2 \in \mathbb{R}$: variance

- Probability density function of multivariate normal distribution

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)$$

  - $\boldsymbol{\mu} \in \mathbb{R}^n$: the cluster mean vector
  - $\Sigma \in \mathbb{R}^{n \times n}$: the covariance matrix

Clustering by Gaussian Mixture Model

- The Gaussian mixture model over K clusters:

  $$f(\boldsymbol{x}) = \sum_{k=1}^{K} f(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k) P(C_k)$$

  - $P(C_k)$ is the mixture parameter satisfying $\sum_{k=1}^{K} P(C_i) = 1$, corresponding to the latent variable

- We denote the set of all parameters by $\boldsymbol{\theta}$ such that

  $$\boldsymbol{\theta} = \{\boldsymbol{\mu}_1, \Sigma_1, P(C_1), \boldsymbol{\mu}_2, \Sigma_2, P(C_2), \ldots, \boldsymbol{\mu}_K, \Sigma_K, P(C_K)\}$$

- Given a dataset D, the objective of clustering is the following:

  $$\max_{\boldsymbol{\theta}} L_D(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log f(\boldsymbol{x}_i)$$

Optimization by EM Algorithm

- Given the current $\boldsymbol{\theta}$, the E-step:

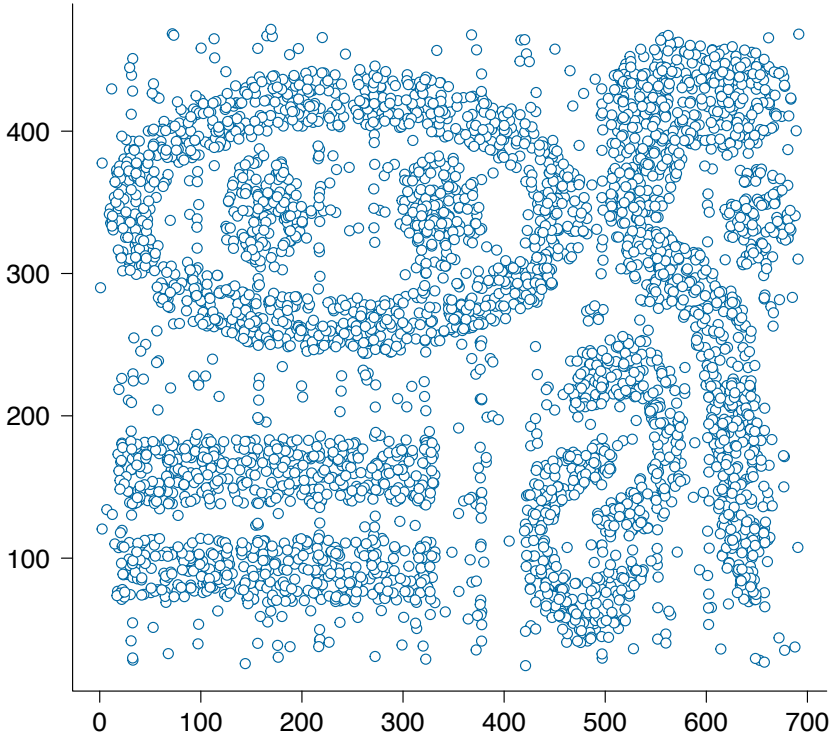
  $$w_{ik} = P(C_k \mid \boldsymbol{x}_i) = \frac{P(C_k \text{ and } \boldsymbol{x}_i)}{P(\boldsymbol{x}_i)} = \frac{f(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma_k) P(C_k)}{f(\boldsymbol{x}_i)}$$

  for each data point $\boldsymbol{x}_i$ and each cluster C_k

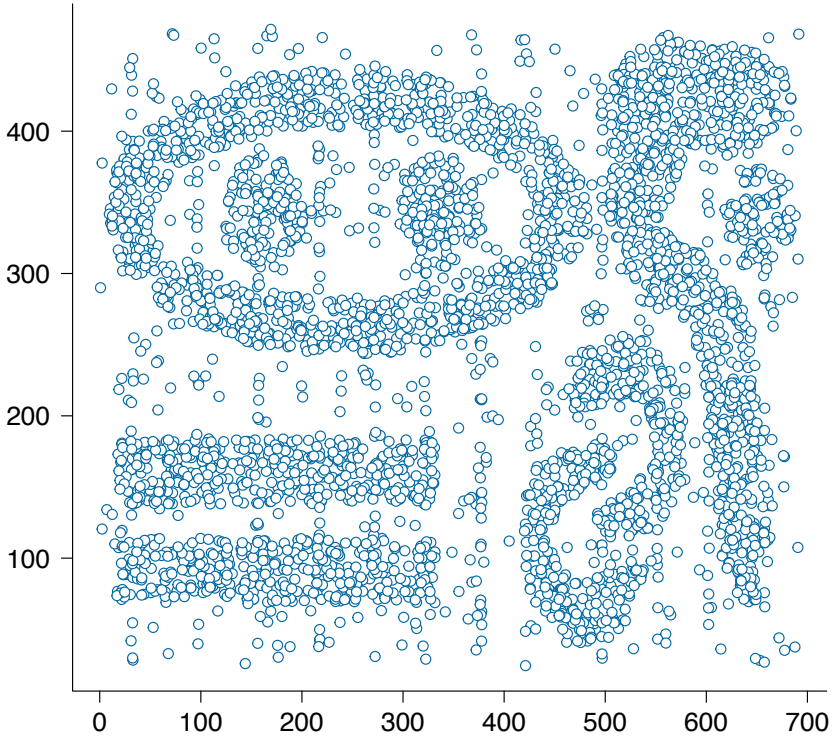
- The M-step:

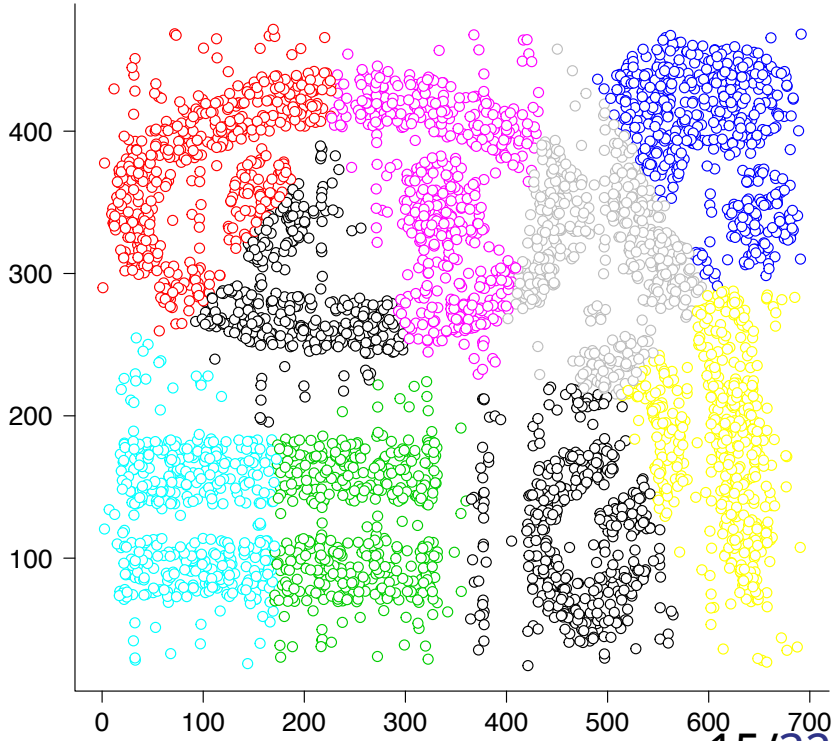
  $$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} w_{ik} \boldsymbol{x}_i}{\sum_{i=1}^{n} w_{ik}}, \quad \Sigma_k = \frac{\sum_{i=1}^{n} w_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2}{\sum_{i=1}^{n} w_{ik}}, \quad P(C_k) = \frac{\sum_{i=1}^{n} w_{ik}}{n}$$

Clusters that *K*-means cannot find



Clusters that *K*-means cannot find





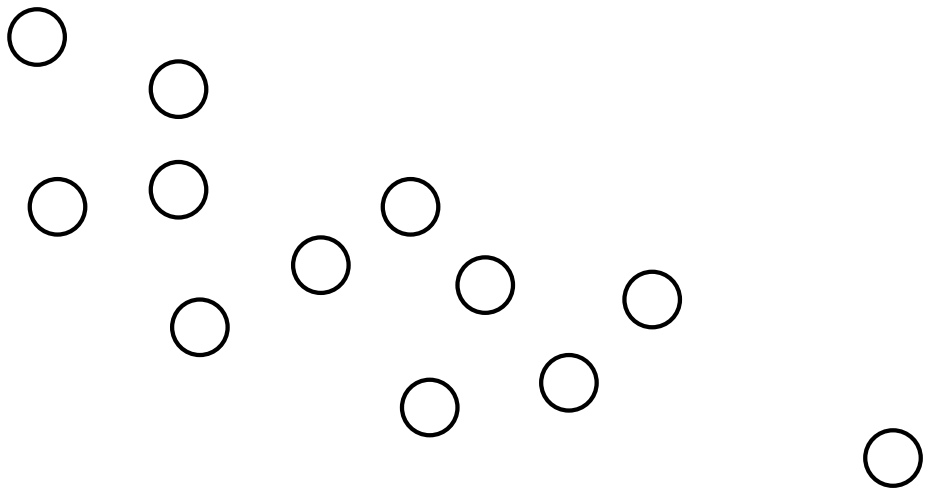
DBSCAN

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm
- ϵ-neighborhood: A ball of radius ϵ around a point $\boldsymbol{x} \in \mathbb{R}^d$,

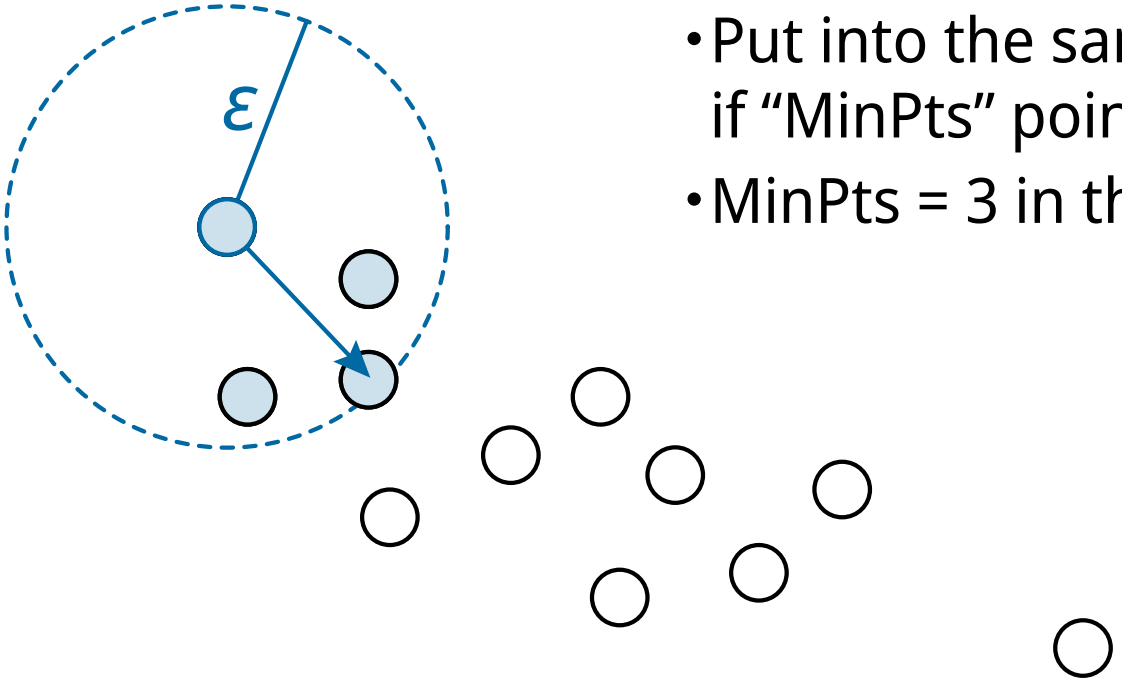
  $N_\epsilon(\boldsymbol{x}) = B(\boldsymbol{x}, \epsilon) = \{\, \boldsymbol{y} \in D \mid \mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}) \leq \epsilon \,\}$

  - $\boldsymbol{x}$ is a core point if $|N_\epsilon(\boldsymbol{x})| \geq \mathrm{MinPts}$
  - $\boldsymbol{x}$ is directly density reachable from $\boldsymbol{y}$ if $\boldsymbol{x} \in N_\epsilon(\boldsymbol{y})$
- $\boldsymbol{x}$ is density reachable from $\boldsymbol{y}$ if there is a chain of points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_l$ s.t. $\boldsymbol{x}_1 = \boldsymbol{y}$, $\boldsymbol{x}_l = \boldsymbol{x}$, and $\boldsymbol{x}_{i+1}$ is directly density reachable from $\boldsymbol{x}_i$
  - $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same cluster if $\boldsymbol{y}$ is density reachable from $\boldsymbol{x}$

Cluster Construction in DBSCAN

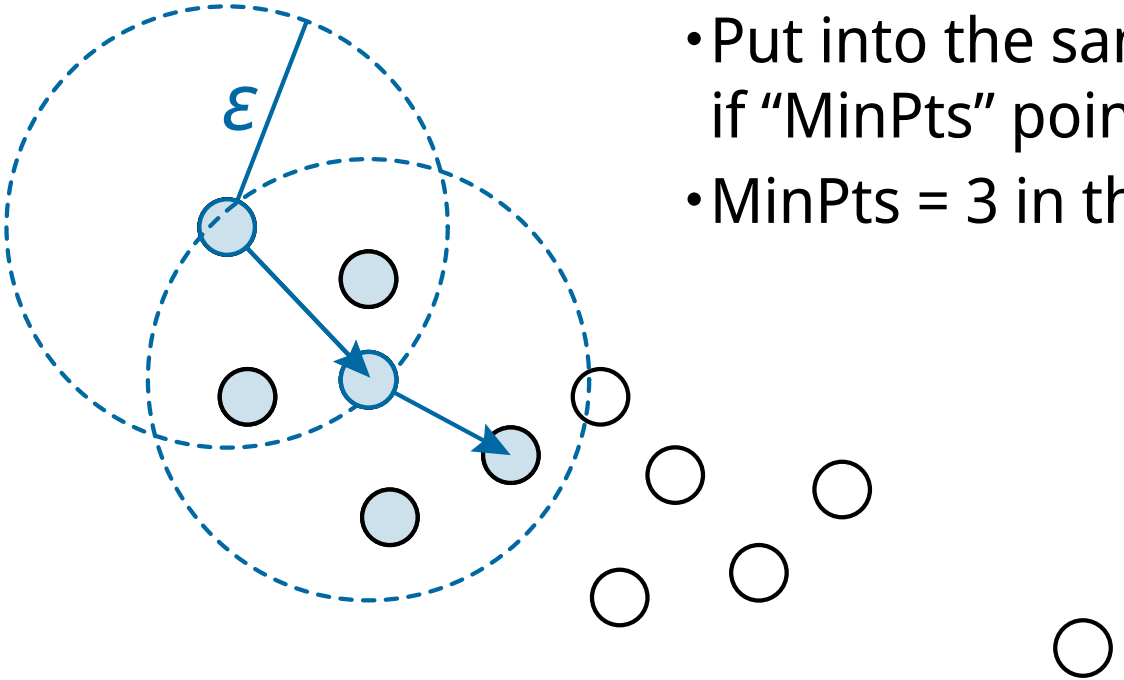


Cluster Construction in DBSCAN



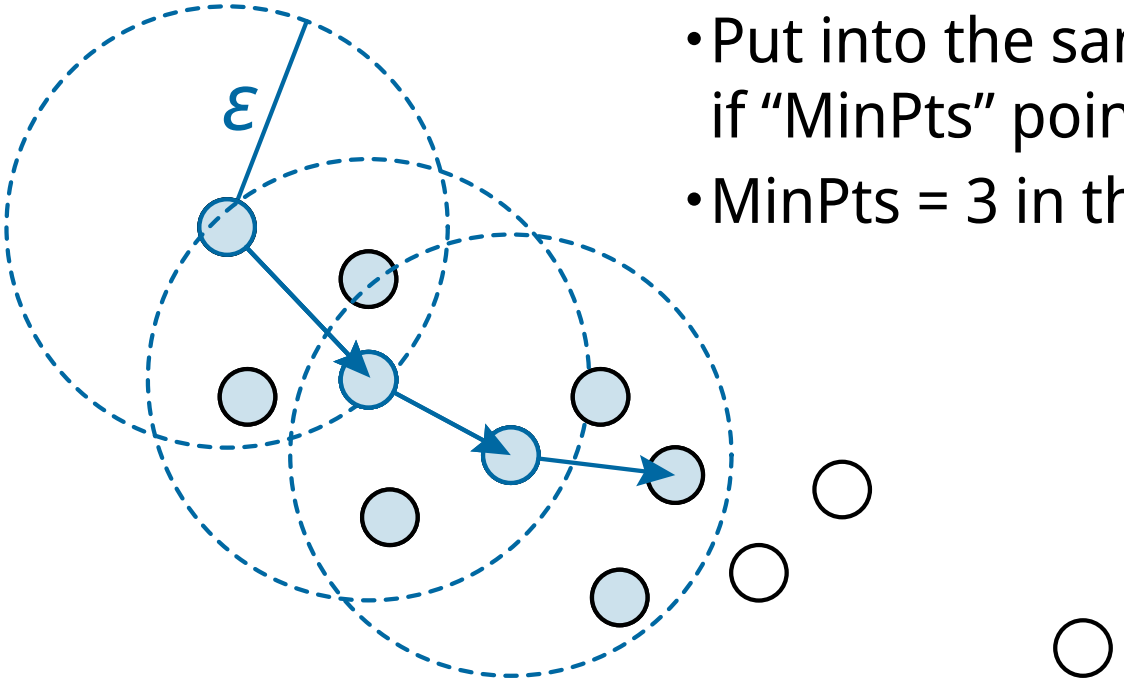
- Put into the same cluster
  if “MinPts” points are in the circle
- MinPts = 3 in this example

Cluster Construction in DBSCAN



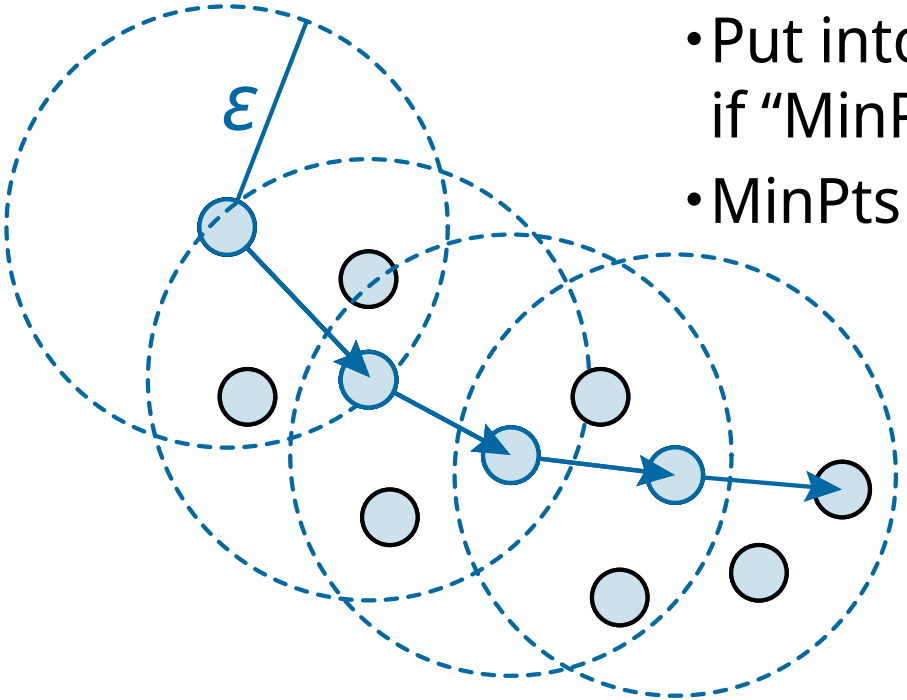
- Put into the same cluster
  if “MinPts” points are in the circle
- MinPts = 3 in this example

Cluster Construction in DBSCAN



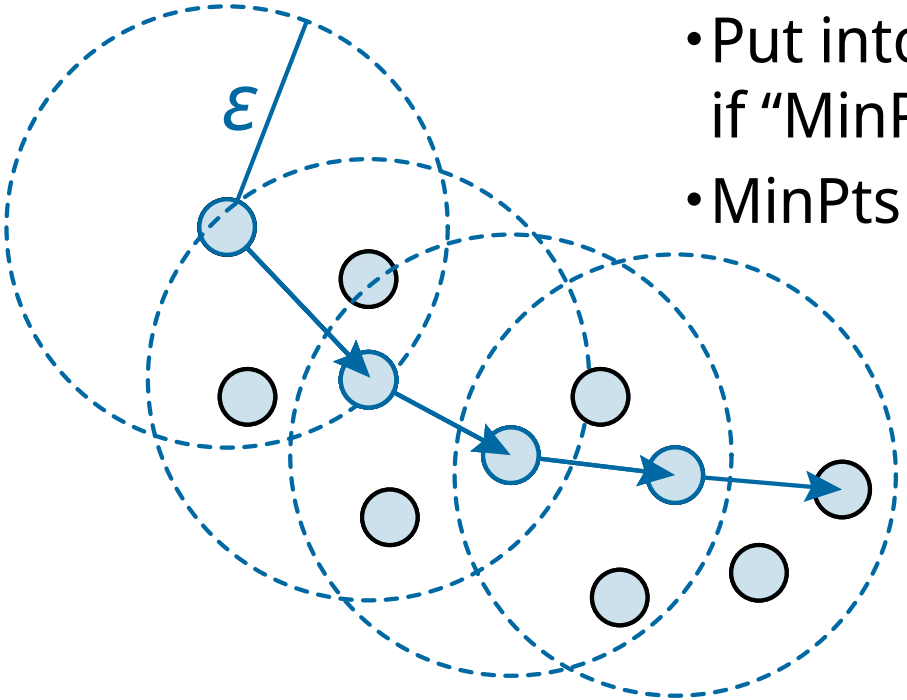
- Put into the same cluster
  if “MinPts” points are in the circle
- MinPts = 3 in this example

Cluster Construction in DBSCAN



- Put into the same cluster if “MinPts” points are in the circle
- MinPts = 3 in this example

Cluster Construction in DBSCAN



- Put into the same cluster if “MinPts” points are in the circle
- MinPts = 3 in this example

Noise if it is not reachable from any other points

Pseudocode of DBSCAN (1/2)

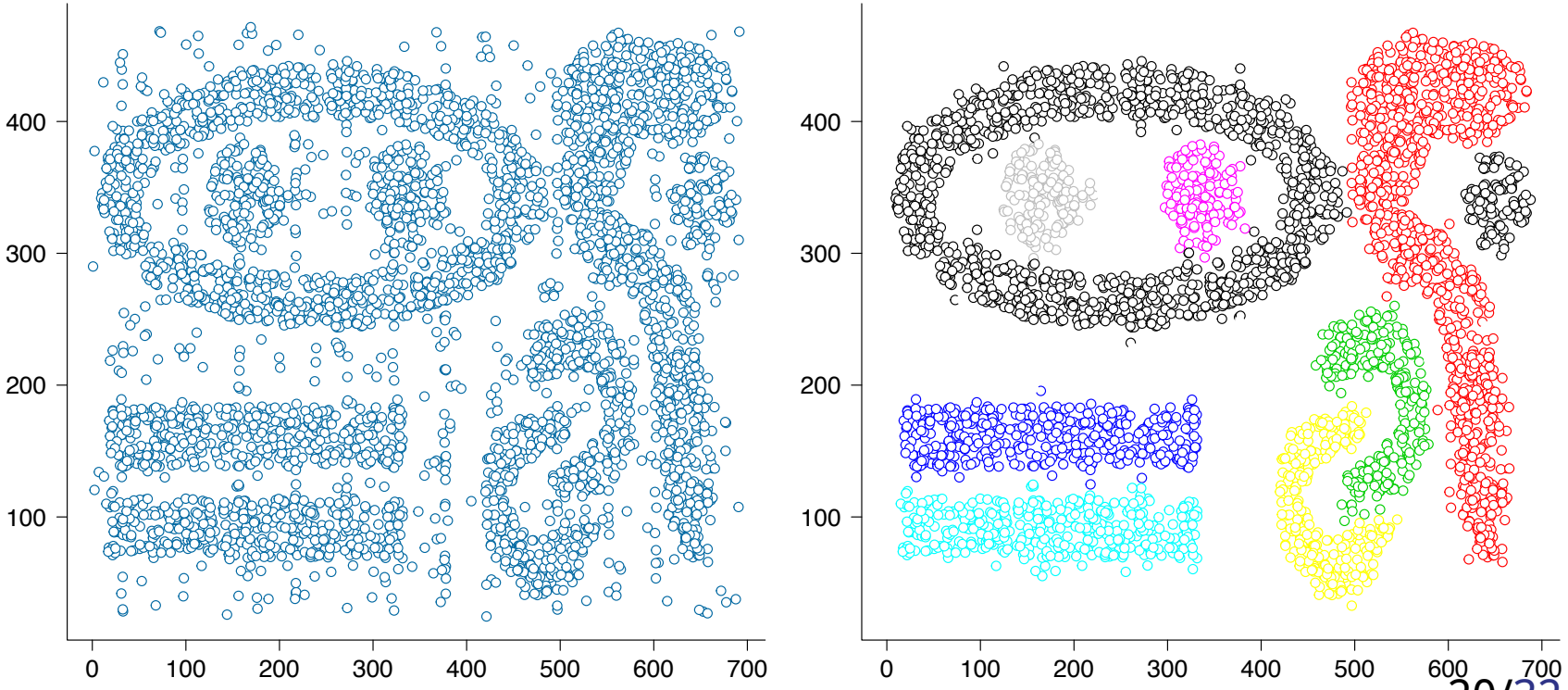
1. $D_{\text{core}} \leftarrow \emptyset$; $k \leftarrow 0$
2. **for each** $\boldsymbol{x} \in D$ **do** // find core points
3. **if** $|N_\epsilon(\boldsymbol{x})| \geq \text{MinPts}$ **then** $D_{\text{core}} \leftarrow D_{\text{core}} \cup \{\boldsymbol{x}\}$
4. **for each** $\boldsymbol{x} \in D_{\text{core}}$ **do**
5. $k \leftarrow k + 1$; DensityConnected(x, k)
6. $\mathcal{C} \leftarrow \{C_1, \ldots, C_k\}$, where $C_i \leftarrow \{\, \boldsymbol{x} \in D \mid \text{id}(\boldsymbol{x}) = i \,\}$
7. $D_{\text{Noise}} \leftarrow \{\, \boldsymbol{x} \in D \mid \text{id}(\boldsymbol{x}) \text{ is not assigned} \,\}$
8. **return** $\mathcal{C}$, D_{Noise}

Pseudocode of DBSCAN (2/2)

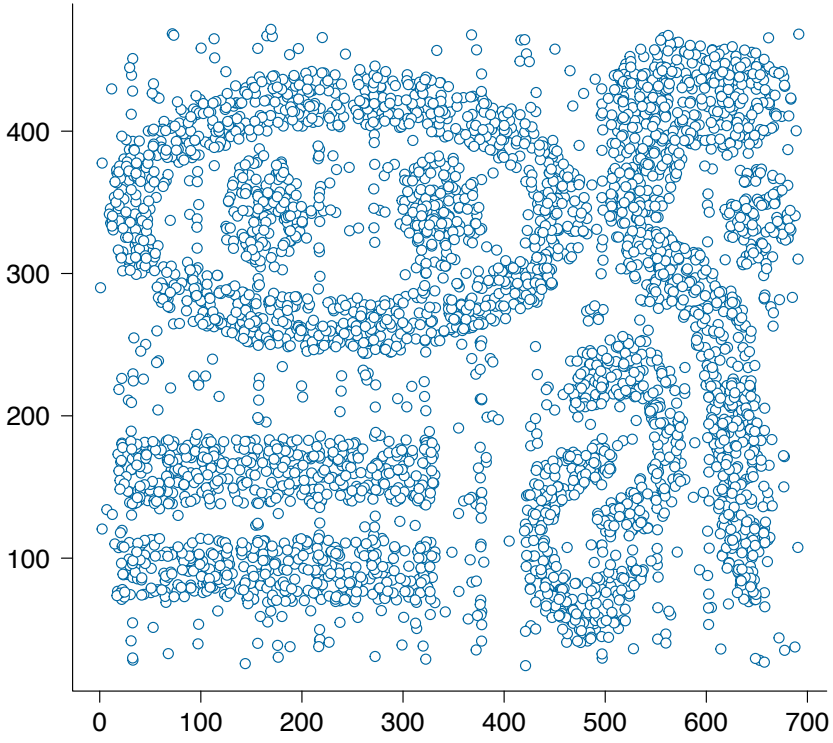
DensityConnected($\boldsymbol{x}, k$)

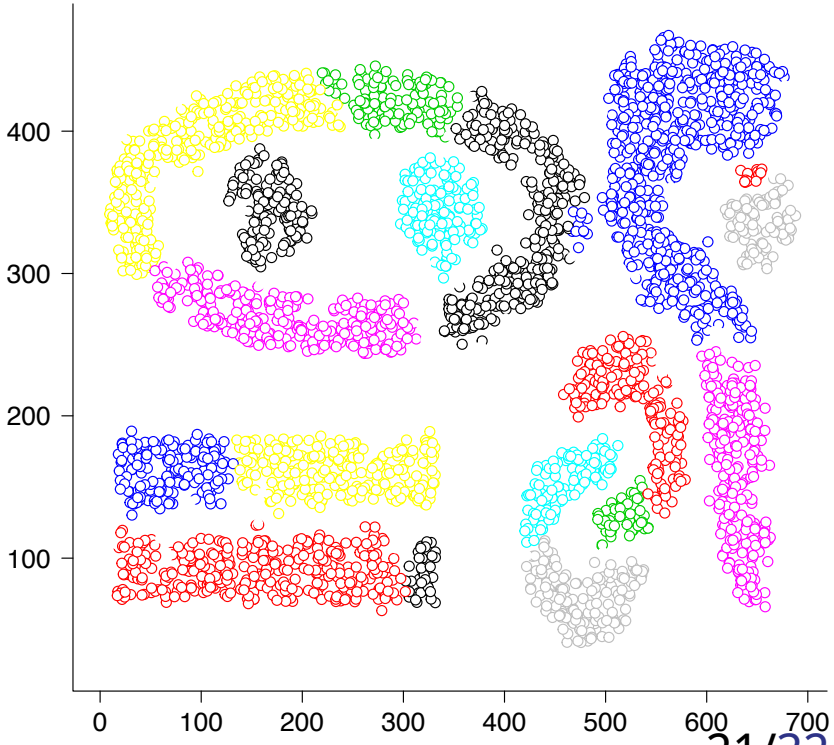
1. **for each** $\boldsymbol{y} \in N_\epsilon(\boldsymbol{x})$ **do**
2. $\quad \mathrm{id}(y) \leftarrow k$
3. $\quad$ **if** $\boldsymbol{y} \in D_{\mathrm{core}}$ **then** DensityConnected($\boldsymbol{y}, k$)

DBSCAN with $\varepsilon = 14$ and MinPts $= 10$

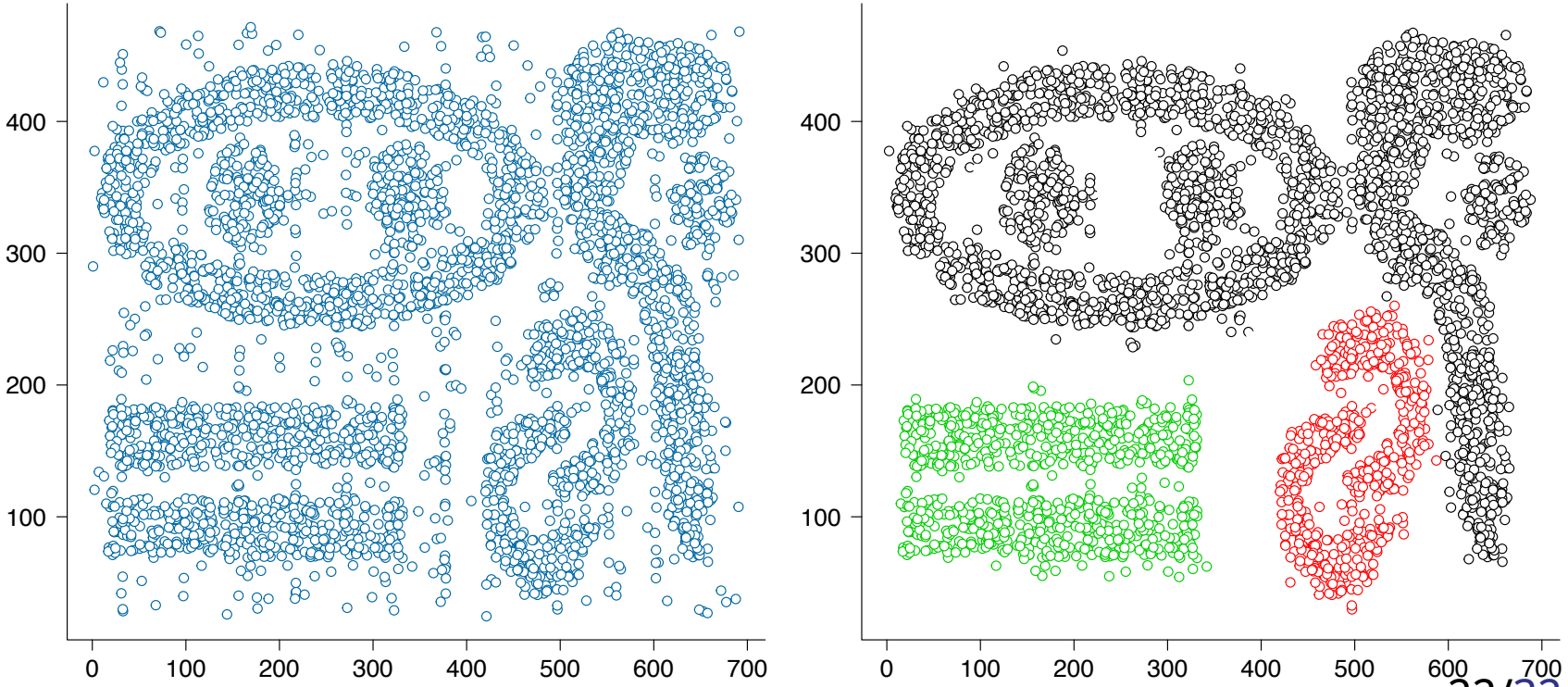


DBSCAN with $\varepsilon = 12$ and MinPts $= 10$





DBSCAN with $\varepsilon = 16$ and MinPts $= 10$



Notes on DBSCAN

- DBSCAN can find clusters of arbitrary shapes
  - The number K of clusters is not needed
- Drawbacks
  - One has to appropriately set ϵ and MinPts, which are often difficult
    - **HDBSCAN** alleviates this problem by hierarchical clustering
  - Runtime is slower than K-means, the time complexity is $O(n^2 d)$ (v.s. $O(ndk)$ in K-means)
    - We can speed-up using an index tree (e.g. k-d tree), but it is not efficient for high-dimensional data
    - **DBSCAN++** speed-ups DBSCAN by sampling

Hierarchical Clustering

- Hierarchical clustering makes a hierarchy of clusters
  - We can find clusters in a cluster
- Two approaches: divisive (top-down) and agglomerative (bottom-up)
  - Divisive: Start from the largest one cluster of the entire dataset and recursively divide clusters
  - Agglomerative: Start from the smallest clusters of single data points and recursively join similar clusters

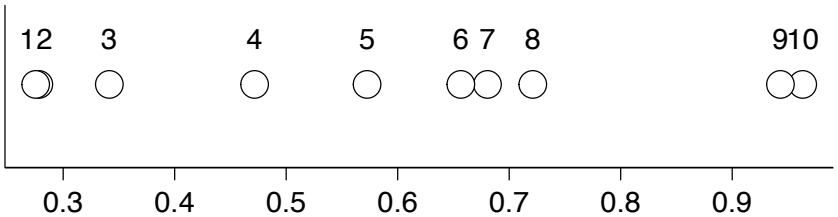
Agglomerative Hierarchical Clustering

1. $\mathcal{C} \leftarrow \{C_i = \{\boldsymbol{x}_i\} \mid \boldsymbol{x}_i \in D\}$
2. **repeat**
3. $\quad (i, j) \leftarrow \text{argmin}_{i,j}\, \text{dist}(C_i, C_j)$
4. $\quad C_{ij} \leftarrow C_i \cup C_j$
5. $\quad \mathcal{C} \leftarrow (\mathcal{C} \setminus \{C_i, C_j\}) \cup \{C_{ij}\}$
6. **until** $|C| = 1$

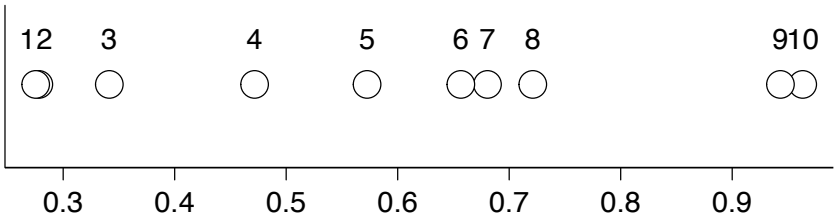
Distance between Clusters

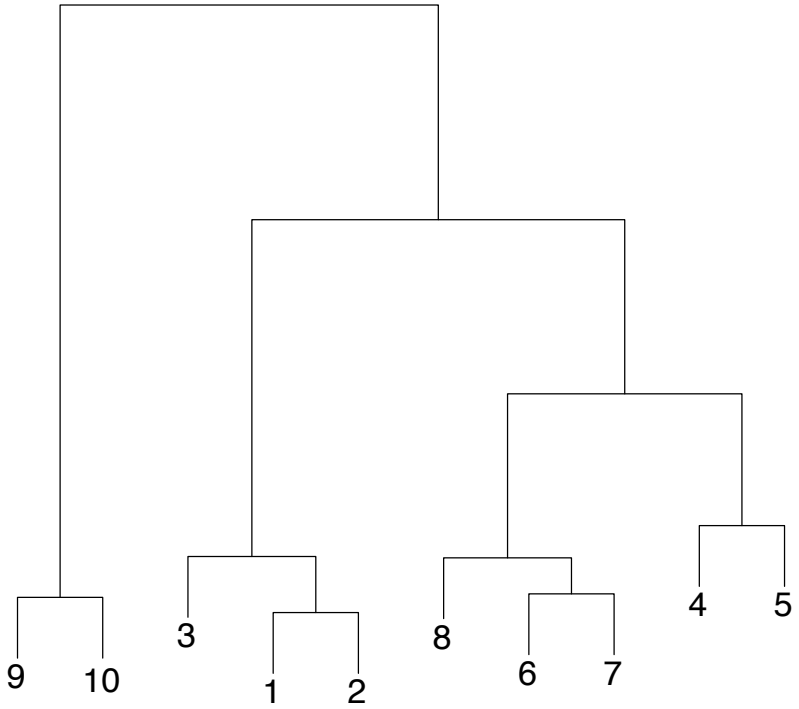
- There are a number of choices how to measure the distance between clusters
- Single link: $\delta(C_i, C_j) = \min\{\text{dist}(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j\}$
- Complete link: $\delta(C_i, C_j) = \max\{\text{dist}(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j\}$
- Group average: $\delta(C_i, C_j) = \sum_{\boldsymbol{x} \in C_i} \sum_{\boldsymbol{y} \in C_j} \text{dist}(\boldsymbol{x}, \boldsymbol{y}) / |C_i||C_j|$

Dendrogram



Dendrogram (agglomerative, complete)





Evaluation of Clusters

- How to evaluate the goodness of clusters?
- Internal and external criteria
  - Internal: Evaluate clusters without ground truth labels
  - External: Evaluate clusters using ground truth labels

Internal Criteria

- Just use SSE($\mathcal{C}$) in K-means or log-likelihood in EM
- Silhouette index: for $\boldsymbol{x}_i \in C_j$,

$$s(i) = \frac{1}{n} \sum_{i=1}^{N} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

$$a(i) = \frac{1}{|C_j| - 1} \sum_{\boldsymbol{y} \in C_j,\ \boldsymbol{y} \neq \boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{x}_i\|^2,$$

$$b(i) = \min_{k \in \{1,\ldots,K\}, k \neq j} \frac{1}{|C_k|} \sum_{\boldsymbol{y} \in C_k} \|\boldsymbol{y} - \boldsymbol{x}_i\|^2$$

  - $-1 \leq s(i) \leq 1$, higher is better

External Criteria

- Accuracy is not appropriate!
- Variation of Information: For two partitions $\mathcal{C} = \{C_1, \dots, C_K\}$ and $\mathcal{T} = \{T_1, \dots, T_M\}$ of D with $|D| = n$,

$$\mathrm{VI}(\mathcal{C}, \mathcal{T}) = -\sum_{i,j} r_{ij} \left(\log \frac{r_{ij}}{|C_i|/n} + \log \frac{r_{ij}}{|T_j|/n} \right)$$

$$r_{ij} = \frac{|C_i \cap T_j|}{n}$$

  - $0 \leq \mathrm{VI}(\mathcal{C}, \mathcal{T}) \leq \min\{\log n, 2 \log(\max K, M)\}$, 0 being the best
- Adjusted Rand index is also often used

Dendrogram Purity

- The standard external criterion to evaluate hierarchical clusters
- Given a dataset D, its hierarchical clusters $\mathcal{H}$, and a ground-truth partition $\mathcal{C} = \{C_1, \dots, C_K\}$
  - LCA($\boldsymbol{x}_i, \boldsymbol{x}_j$): the smallest cluster in $\mathcal{H}$ that includes both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
  - pur($F; G$) $= |F \cup G|/|F|$ for a pair of clusters $F, G \in D$
  - Let $P = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i, \boldsymbol{x}_j \in C_k\}$
- Dendrogram purity of $\mathcal{H}$ is

$$\mathrm{DP}(\mathcal{H}) = \frac{1}{|P|} \sum_{k=1}^{K} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j} \mathrm{pur}(\mathrm{LCA}(\boldsymbol{x}_i, \boldsymbol{x}_j); C_k)$$

Summary

- Popular clustering methods are introduced
  - K-means
  - EM algorithm
  - DBSCAN
  - Hierarchical clustering
- Clustering results can be evaluated internally or externally