



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  $D = \{ \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \}, \quad \mathbf{x}_i \in \mathbb{R}^n$
- Clustering: Find a partition  $C = \{C_1, C_2, \dots, C_K\}$  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:

$$SSE(C) = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_{i}} ||\boldsymbol{x} - \boldsymbol{\mu}_{k}||^{2} = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_{i}} \sum_{j=1}^{n} (x^{j} - \mu_{k}^{j})^{2}$$

- $\boldsymbol{\mu}_k$  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  $C^*$  s.t.

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

### Pseudocode of K-means

- Input: Dataset D, Number of clusters K
- Output: Clustering  ${\cal C}$
- 1. Randomly initialize *K* centroids:  $\mu_1, \mu_2, \ldots, \mu_K$

2. repeat

3. 
$$C_k \leftarrow \emptyset$$
 for all  $k \in \{1, 2, \dots, K\}$ 

4. for each  $x \in D$  do // cluster assignment

5. 
$$k^* \leftarrow \operatorname{argmin}_{k \in \{1,2,\dots,K\}} \| \boldsymbol{x} - \boldsymbol{\mu}_k \|$$

$$6. \qquad C_{k^*} \leftarrow C_{k^*} \cup \{\boldsymbol{x}\}$$

- 7. for each  $k \in \{1, 2, ..., K\}$  do // centroid update
- 8.  $\boldsymbol{\mu}_k \leftarrow (1/|C_k|) \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}$
- 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(npK)
  - 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  $\mathbf{x} \in D$  and  $\boldsymbol{\mu}_1 \leftarrow \mathbf{x}$
- 2. for each  $k = \{2, 3, ..., K\}$  do
- 3. for each  $\boldsymbol{x} \in D$  do  $D(\boldsymbol{x}) \leftarrow \min_{i \in \{1,2,\dots,k\}} \|\boldsymbol{x} \boldsymbol{\mu}_i\|^2$
- 4. for each  $\mathbf{x} \in D$  do  $p(\mathbf{x}) \leftarrow D(\mathbf{x}) / \sum_{\mathbf{s} \in D} D(\mathbf{s})$
- 5. Select  $\boldsymbol{\mu}_k$  from *D* using the probability distribution  $p(\boldsymbol{x})$  for each  $\boldsymbol{x} \in D$
- 6. Perform *K*-means using  $\mu_1, \mu_2, \ldots, \mu_K$  as the initial cluster centers

# **EM Clustering**

- In K-means, each point either belongs to a cluster or not
   → hard clustering
- How about obtaining the probability of cluster membership?
   → 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; θ) over observed variables X and hidden (latent) variables Y, with parameters θ
   Goal: Maximize the likelihood of p(X; θ)
- 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  $\theta$ **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  $\theta^{(t+1)}$

$$\theta^{(t+1)} = \operatorname{argmax}_{\theta^{(t+1)}} Q(\theta^{(t+1)}, \theta^{(t)})$$

$$- Q(\theta^{(t+1)}, \theta^{(t)}) = \sum_{Y} p(Y \mid X; \theta^{(t)}) \log p(X, Y; \theta^{(t+1)})$$

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

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$
  
-  $x \in \mathbb{R}$ : mean,  $\sigma^2 \in R$ : variance

Probability density function of multivariate normal distribution

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

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

### **Gaussian Mixture Model**

- The Gaussian mixture model over *K* clusters:
  - $f(\boldsymbol{x}) = \sum_{k=1}^{K} f(\boldsymbol{x} \mid \boldsymbol{\mu}_{k}, \boldsymbol{\Sigma}_{k}) P(\boldsymbol{C}_{k})$ 
    - $P(C_k)$  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  $\boldsymbol{\theta} = \{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, P(C_1), \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, P(C_2), \dots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K, P(C_K)\}$
- Given a dataset *D*, the objective is to maximize the log-likelihood:  $\max_{\theta} L_D(\theta) = \max_{\theta} \sum_{i=1}^{N} \log f(\mathbf{x}_i)$ 13/31

# **EM Clustering**

• Given the current **\theta**, the E-step:

$$w_{ik} = P(C_k \mid \mathbf{x}_i) = \frac{P(C_k \text{ and } \mathbf{x}_i)}{P(\mathbf{x}_i)} = \frac{f(\mathbf{x}_i; \mathbf{\mu}_k, \Sigma_k)P(C_k)}{f(\mathbf{x}_i)}$$
for each data point  $\mathbf{x}_i$  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
- $\varepsilon$ -neighborhood: A ball of radius  $\varepsilon$  around a point  $\mathbf{x} \in \mathbb{R}^n$ ,  $N_{\varepsilon}(\mathbf{x}) = B(\mathbf{x}, \varepsilon) = \{ \mathbf{y} \in D \mid d(\mathbf{x}, \mathbf{y}) \le \varepsilon \}$ 
  - **x** is a core point if  $|N_{\varepsilon}(\mathbf{x})| \ge \text{MinPts}$
  - **x** is directly density reachable from **y** if  $\mathbf{x} \in N_{\varepsilon}(\mathbf{y})$
- x is density reachable from y if there is a chain of points x<sub>1</sub>, x<sub>2</sub>,..., x<sub>i</sub> s.t. x<sub>1</sub> = y, x<sub>i</sub> = x, and x<sub>i+1</sub> is directly density reachable from x<sub>i</sub>
  - x and y are in the same cluster if y is density reachable from x





17/31



17/31





17/31

# Pseudocode of DBSCAN (1/2)

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

8. return C,  $D_{Noise}$ 

### Pseudocode of DBSCAN (2/2)

DensityConnected(**x**, k)

- 1. for each  $y \in N_{\varepsilon}(x)$  do
- 2.  $id(y) \leftarrow k$
- 3. if  $y \in D_{core}$  then DensityConnected(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
  - Runtime is slower than *K*-means, the time complexity is  $O(n^2d)$  (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

# **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. 
$$C \leftarrow \{C_i = \{\boldsymbol{x}_i\} \mid \boldsymbol{x}_i \in D\}$$

2. repeat

- 3.  $(i, j) \leftarrow \operatorname{argmin}_{i,j} \delta(C_i, C_j)$
- $4. \quad C_{ij} \leftarrow C_i \cup C_j$
- 5.  $C \leftarrow (C \setminus \{C_i, C_j\}) \cup \{C_{ij}\}$

6. **until** |C| = 1

### **Distance between Clusters**

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

### 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(C) in K-means or log-likelihood in EM
- Silhouette index: for  $\mathbf{x}_i \in C_j$ ,

$$s(i) = \frac{1}{N} \sum_{i=1}^{N} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$
  
$$a(i) = \frac{1}{|C_j| - 1} \sum_{\mathbf{y} \in C_j, \ \mathbf{y} \neq \mathbf{x}} ||\mathbf{y} - \mathbf{x}_i||^2, \quad b(i) = \min_{k \in \{1, \dots, K\}, k \neq j} \frac{1}{|C_k|} \sum_{\mathbf{y} \in C_k} ||\mathbf{y} - \mathbf{x}_i||^2$$

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

### **External Criteria**

- Accuracy is not appropriate!
- Variation of Information: For two partitions  $C = \{C_1, \dots, C_K\}$  and  $T = \{T_1, \dots, T_M\}$  of D with |D| = N,  $VI(C, T) = -\sum_{i,j} r_{ij} \left( \log \frac{r_{ij}}{|C_i|/N} + \log \frac{r_{ij}}{|T_j|/N} \right)$   $r_{ij} = \frac{|C_i \cap T_j|}{N}$ 
  - $0 \leq VI(\mathcal{C}, \mathcal{T}) \leq \min\{\log N, 2 \log(\max K, M)\}, 0 \text{ being the best} \}$
- Adjusted Rand index is also often used

# Summary

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