Clustering

Ruwen Qin
Stony Brook University

October 14, 2025

Data used in this module:

- protein.csv
- MNIST dataset from sklearn.datasets

Python notebook used in this module

- Clustering.ipynb

1 Introduction

We encounter a large amount of information and store or represent it as data. A common task is to divide these data into a set of groups. Clustering is an unsupervised learning method that separates a finite unlabeled dataset into a finite and discrete set of clusters. In this learning module, we study the fundamentals of clustering methods. This lecture note is mainly developed based on references [1, 5].

1.1 Definitions of Clustering

Clustering can be described as:

1. Instances, in the same cluster, must be similar as much as possible;
2. Instances, in the different clusters, must be different as much as possible;
3. Measurement for similarity and dissimilarity must be clear and have the practical meaning.

1.2 Mathematical Descriptions of Clustering

Given a dataset that consists of M instances on an N dimensional feature space: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$, where $\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,N}]^{\mathrm{T}}$ is the feature vector of instance i in the feature space, for $i = 1, \ldots, M$.

Hard Partitional Clustering seeks to partition the dataset $\mathbf{X}$ into $K(\leq M)$ clusters: $C = \{C_1, \ldots, C_K\}$, which satisfy:

- $C_k \neq \emptyset$, for $C_k \in C$
- $C_k \cap C_l = \emptyset$, for C_k, $C_l \in C$, and $k \neq l$
- $\cup_{k=1}^{K} C_k = \mathbf{X}$

That is, we partition $\mathbf{X}$ into K non-empty mutually exclusive and collective inclusive clusters.

Hierarchical clustering attempts to construct a nested tree structure for data partitioning of $\mathbf{X}$. $H = \{H_1, \ldots, H_Q\}$ are levels of the hierarchical tree with the subscript indicating the order of data partition. Clusters $C = \{C_1, \ldots, C_K\}$ such that $C_i \in H_m$, $C_j \in H_l$, and $m > l$ implies that $C_i \in C_j$ or $C_i \cap C_j = \emptyset$ for all i, $j \neq i \in \{1, \ldots, K\}$, and m, $l \in \{1, \ldots, Q\}$.

1.3 Process of Clustering

Figure 1 summarizes the process of clustering [5]

1. Feature Extraction and Selection: Extract and select the most representative features from the original dataset;
2. Clustering Algorithm Design or Selection: This step usually is combined with the selection of a corresponding proximity measure and the construction of a criterion function according to the characteristics of the problem;
3. Clusters Validation: Evaluate the clustering result and judge the validity of algorithm;
4. Result Interpretation: This step is about to give a practical explanation of the clustering results, and provide users with meaningful insights from the original data so that they gain the knowledge to solve the problems encountered.

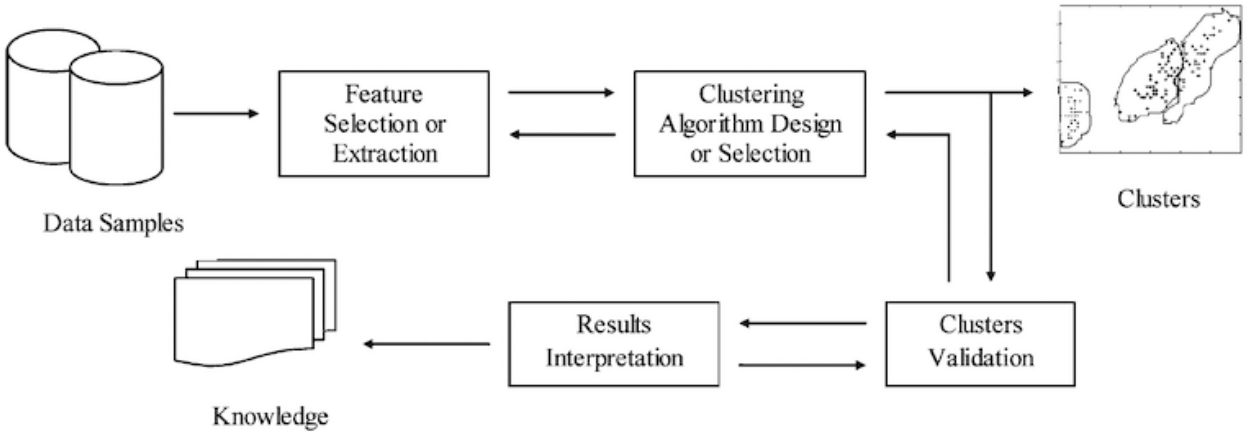

Figure 1: Clustering process [5]

2 Similarity and Dissimilarity Measures Between Feature Vectors

Every instance in the dataset is represented by its feature vector in the feature space. The similarity or dissimilarity measure between any two feature vectors is the foundation for clustering algorithms. Chapter 6 in [1] has a thorough discussion of similarity and dissimilarity measures. We selected several to present in this learning modules.

2.1 Similarity and Dissimilarity Functions

If $\mathbf{x}_i$ and $\mathbf{x}_j$ are two instances in their N-dimensional feature space. The features can be quantitative or qualitative, continuous or integers, nominal or ordinal, which determine the corresponding measure mechanisms.

A *distance or dissimilarity function* on the dataset $\mathbf{X}$ is defined to satisfy the following conditions:

- Symmetry: $D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_j, \mathbf{x}_i)$;
- Positivity: $D(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ for all $\mathbf{x}_i$ and $\mathbf{x}_j$.

If

- Triangle Inequality: $D(\mathbf{x}_i, \mathbf{x}_j) \leq D(\mathbf{x}_i, \mathbf{x}_k) + D(\mathbf{x}_k, \mathbf{x}_j)$ for all $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$;
- Reflexivity: $D(\mathbf{x}_i, \mathbf{x}_j) = 0$ iff $\mathbf{x}_i = \mathbf{x}_j$

also hold, the distance function is a metric.

A similarity function is defined to satisfy the following conditions:

- Symmetry: $S(\mathbf{x}_i, \mathbf{x}_j) = S(\mathbf{x}_j, \mathbf{x}_i)$;
- Positivity: $0 \leq S(\mathbf{x}_i, \mathbf{x}_j) \leq 1$ for all $\mathbf{x}_i$ and $\mathbf{x}_j$.

If

- Inequality: $S(\mathbf{x}_i, \mathbf{x}_j)S(\mathbf{x}_j, \mathbf{x}_k) \leq [S(\mathbf{x}_i, \mathbf{x}_j) + S(\mathbf{x}_j, \mathbf{x}_k)]S(\mathbf{x}_i, \mathbf{x}_k)$;
- Reflexivity: $S(\mathbf{x}_i, \mathbf{x}_j) = 1$ iff $\mathbf{x}_i = \mathbf{x}_j$

also hold, the similarity function is a similarity metric.

We define an $M \times M$ symmetric matrix, called *proximity matrix*, whose (i, j)th element is the similarity or dissimilarity measure for the ith and jth feature vectors in $\mathbf{X}$.

Distance functions are used to measure continuous features, while similarity measures are more important for qualitative variables.

2.2 Distance Measures

If feature values are all numerical, the distance between two vectors measures their dissimilarity.

Minkowski Distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_p. \tag{1}$$

It is the Manhattan distance when $p = 1$, the Euclidean distance when $p = 2$, and the maximum distance if $p = \infty$.

Mahalanobis Distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}_i - \mathbf{x}_j)} \tag{2}$$

where Σ is the covariance matrix of the dataset.

Average Distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \left(\frac{1}{N}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right)^{1/2} \tag{3}$$

which is a modification of Euclidean distance.

Cosine Distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{< \mathbf{x}_i, \mathbf{x}_j >}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2} \tag{4}$$

where $< \cdot, \cdot >$ is the inner product of two vectors.

Chord Distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \left(2 - 2\frac{< \mathbf{x}_i, \mathbf{x}_j >}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}\right)^{1/2} \tag{5}$$

More distances for quantitative feature vectors are introduced in Table 6.1 in [1].

2.3 Similarity Measures for Binary Feature Vectors

Let's first consider the scenario that all features are binary. $N_{0,0}$ and $N_{1,1}$ represent the number of simultaneous absence or presence of features in two feature vectors, and $N_{0,1}$ and $N_{1,0}$ count the features present only in one instance but not the other.

The first type of similarity measures treats the situation that the feature is simultaneously present in both instances (1,1) and absent in both instances (0,0) equally (called symmetric coefficients, thus defines the similarity between two binary features as):

$$S_{i,j} = S(\mathbf{x}_i, \mathbf{x}_j) = \frac{N_{0,0} + N_{1,1}}{N_{0,0} + N_{1,1} + w(N_{1,0} + N_{0,1})}. \tag{6}$$

- if $w = 1$, it is simple matching coefficient
- if $w = 2$, it is Rogers and Tanimoto measure
- if $w = 1/2$, it is Gower and Legendre measure

The second type of similarity measures only cares the simultaneous presence of the feature in both instance (called asymmetric coefficients) and defines the similarity between two binary features as:

$$S_{i,j} = S(\mathbf{x}_i, \mathbf{x}_j) = \frac{N_{1,1}}{N_{1,1} + w(N_{1,0} + N_{0,1})}. \tag{7}$$

- if $w = 1$, it is Jaccard coefficient
- if $w = 2$, it is Sokal and Sneath measure
- if $w = 1/2$, it is Gower and Legendre measure

Pearson Correlation Coefficient:

$$S_{i,j} = S(\mathbf{x}_i, \mathbf{x}_j) = \frac{(N_{0,0}N_{1,1}) - (N_{0,1}N_{1,0})}{\sqrt{(N_{1,1} + N_{0,1})(N_{1,1} + N_{1,0})(N_{0,1} + N_{0,0})(N_{1,0} + N_{0,0})}} \tag{8}$$

Dice:

$$S_{i,j} = S(\mathbf{x}_i, \mathbf{x}_j) = \frac{2N_{1,1}}{2N_{1,1} + N_{1,0} + N_{0,1}} \tag{9}$$

Other similarity measures for binary vectors are summarized in Tables 6.3 and 6.4 in [1].

Consider an example where $\mathbf{x}_i = [1, 0, 0, 1, 1, 0]^{\mathrm{T}}$ and $\mathbf{x}_j = [1, 0, 1, 1, 0, 1]^{\mathrm{T}}$, can you calculate those similarity measures?

2.4 Similarity Measures for Nominal Feature Vectors

If feature values are all categorical with more than two states, we can define:

$$s_{i,j,l} = \begin{cases} 1, & \text{if } x_{i,l} = x_{j,l} \\ 0, & \text{if } x_{i,l} \neq x_{j,l} \end{cases} \tag{10}$$

for $l = 1, \ldots, N$ to indicate the feature-wise match between the two instances.

The similarity of two feature vectors can be measured by the following metrics:

Jaccard Similarity:

$$S_{i,j} = \frac{\sum_{l=1}^{N} s_{i,j,l}}{2N - \sum_{l=1}^{N} s_{i,j,l}}. \tag{11}$$

Hamming Similarity:

$$S_{i,j} = \frac{\sum_{l=1}^{N} s_{i,j,l}}{N}. \tag{12}$$

Consider an example where

$$\mathbf{x}_i = [NY, A, L, U, Y]^{\mathrm{T}}$$
$$\mathbf{x}_j = [NY, B, L, R, N]^{\mathrm{T}}$$

What are the Jaccard similarity and Hamming Similarity for these two instances, respectively?

2.5 Similarity Measures for Ordinal Feature Vectors

Distance-based Measures:

Ordinal features order multiple states according to some standard and can be compared by using continuous dissimilarity measures in Section 2.2.

Kendall Rank Correlation Coefficient [3]:

Let $\mathbb{O} = \{O_1, \ldots, O_l\}$ be the ordered categories of the features for $\mathbf{X}$. An N-dimensional feature vector can be decomposed into $\frac{1}{2}N(N-1)$ ordered pairs:

$$\mathcal{P}_i = \{(x_{i,1}, x_{i,2}), \ldots, (x_{i,1}, x_{i,N}), \ldots, (x_{i,N-1}, x_{i,N})\},$$

and let's define $p_{i,k}$ as the kth element of the set $\mathcal{P}_i$. For $\mathcal{P}_i$ and $\mathcal{P}_j$, $p_{i,k}$ and $p_{j,k}$ are a concordant pair if their rankings agree, and are a discordant pair if their rankings do not agree. We define $\Delta_{i,j}$ as the

number of discordant pairs, which measures a distance between sets called the symmetric difference distance. Accordingly, the similarity between the two instances is:

$$S_{i,j} = 1 - \frac{2\Delta_{i,j}}{N(N-1)}. \tag{13}$$

Consider an example with three ordered categories:

$$\mathbb{O} = \{G(ood), N(eural), P(oor)\}$$

and

$$\mathbf{x}_i = [G, P, N, P]^{\mathrm{T}}$$
$$\mathbf{x}_j = [G, P, N, N]^{\mathrm{T}}$$

What is the Kendall Ranking Correlation Coefficient that measures the similarity between these two ordered categorical instances?

Spearman's Rank Correlation Coefficient: It is the Pearson correlation coefficient between the ordered feature vectors:

$$S_{i,j} = 1 - \frac{6\|R(\mathbf{x}_i) - R(\mathbf{x}_j)\|_2^2}{N(N^2-1)} \tag{14}$$

where $R(\mathbf{x})$ maps $\mathbf{x}$ to the orders of elements.

Consider the same example above. The mapping function is: *Good*: 3, $Neural$: 2, *Poor*: 1.

2.6 Similarity Measures for Mixed Feature Vectors

For instances characterized by mixed types of features, we can map all features into the interval [0,1] in order to use distance measures such as Euclidean distance. We can also transfer them into binary features and then use the similarity measures for binary features. But both transformation methods could lead to information loss.

Gower [2] proposed a method:

$$S_{i,j} = \frac{\sum_{l=1}^{N} S'_{i,j,l}\delta_{i,j,l}}{\sum_{l=1}^{N} \delta_{i,j,l}} \tag{15}$$

where

$$\delta_{i,j,l} = \begin{cases} 1, & \text{if } x_{i,l} \text{ and } x_{j,l} \text{ can be compared;} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

for all features, $l = 1, \ldots, N$, and $S'_{i,j,l}$ is the similarity measure between $x_{i,l}$ and $x_{j,l}$ if they can be compared. Specifically,

- If $x_{i,l}$ and $x_{j,l}$ are quantitative,

$$S'_{i,j,l} = 1 - \frac{|x_{i,l} - x_{j,l}|}{R_l}, \tag{17}$$

  where R_l is the range of feature l.
- If $x_{i,l}$ and $x_{j,l}$ are binary, $S'_{i,j,l}$ is one if the feature is present in both instances, and zero otherwise.
- If $x_{i,l}$ and $x_{j,l}$ are nominal, $S'_{i,j,l}$ is one if $x_{i,l} = x_{j,l}$ and zero otherwise.

Consider an example where:

$$\mathbf{x}_i = (NY, Yes, 45, N/A)$$
$$\mathbf{x}_j = (MA, No, 30, good)$$

and the range for the third variable is 70.

2.7 Distance Measure for Mixed Feature Vectors

The distance between two instances with mixed types of features is calculated as [2]:

$$d_{i,j} = \frac{\left(\sum_{l=1}^{N} {d'_{i,j,l}}^2 \delta_{i,j,l}\right)^{1/2}}{\sum_{l=1}^{N} \delta_{i,j,l}} \tag{18}$$

where $\delta_{i,j,l}$ is defined in (16).

- If $x_{i,l}$ and $x_{j,l}$ are ordinal or numerical

$$d'_{i,j,l} = \frac{|x_{i,l} - x_{j,l}|}{R_l}, \tag{19}$$

  where R_l is the range of feature l.
- If $x_{i,l}$ and $x_{j,l}$ are binary, $d'_{i,j,l}$ is zero if the feature is present in both instances or absent in both, and zero otherwise.
- If $x_{i,l}$ and $x_{j,l}$ are nominal, $d'_{i,j,l}$ is zero if $x_{i,l} = x_{j,l}$ and one otherwise.
- If $x_{i,l}$ and $x_{j,l}$ are numerical, it is preferred that we choose a normalized distance measure, for example,

$$d'_{i,j,l} = \frac{|{x^*_{i,l}}^2 - {x^*_{j,l}}^2|}{\sigma_l}, \tag{20}$$

  where $x^* = (x - \mu_l)/\sigma_l$ is normalized feature value, and μ_l and σ_l are the mean and standard deviation of feature l, respectively.

Consider the same example in Section 2.6, but let's measure the distance rather than similarity. For the numerical feature, the mean μ is 30, and the standard deviation σ is 10.

In this learning module, we skip the discussion of generalized Minkowiski Distance, which is also useful. Readers are suggested to study Section 6.5.3 in [1].

3 Similarity and Dissimilarity Measures between Clusters

Many clustering algorithms are hierarchical. That is, clustering is a sequence of nested partitions. In an agglomerative hierarchical algorithm, the two most similar groups are merged to form a large cluster at each step, and this processing is continued until the desired number of clusters is obtained. In a divisive hierarchical algorithm, the process is reversed by starting with all data points in one cluster and subdividing into smaller clusters. In either case, we need to compute the distance between an instance and a cluster and the distance between two clusters [1].

3.1 Distance Measures Between Clusters

Let C_i and C_j be two clusters of numerical instances, and $C_i \cap C_j = \emptyset$. Let $d(\cdot, \cdot)$ be a distance function between two instances, defined in Section 2.2.

Mean-based Distance

$$D(C_i, C_j) = d(\mu(C_i), \mu(C_j)) \tag{21}$$

where $\mu(C_i)$ and $\mu(C_j)$ are the means of the two clusters, respectively. That is,

$$\mu(C) = \sum_{x_l \in C} \mathbf{x}_l / |C|, \tag{22}$$

where $|C|$ means the size of the cluster.

The Nearest Neighbor Distance

$$D(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y}). \tag{23}$$

The Farthest Neighbor Distance

$$D(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y}). \tag{24}$$

The Average Neighbor Distance

$$D(C_i, C_j) = \frac{1}{|C_i|\,|C_j|} \sum_{x_l \in C_i} \sum_{y_m \in C_j} d(\mathbf{x}_l, \mathbf{y}_m) \tag{25}$$

3.2 Lance-Williams Formula

In an agglomerative hierarchical algorithm, we need to compute the distances between old clusters and a new cluster formed by two clusters. Lance-Williams formula is a recurrence formula that gives the distance a cluster C_k and a cluster C formed by merging two clusters C_i and C_j (i.e., $C = C_i \cup C_j$):

$$\begin{aligned} &D(C_k, C_i \cup C_j) \\ =&\alpha_i D(C_k, C_i) + \alpha_j D(C_k, C_j) \\ +&\beta D(C_i, C_j) \\ +&\gamma |D(C_k, C_i) - D(C_k, C_j)| \end{aligned} \tag{26}$$

By a suitable choice of the parameters α_i, α_j, β, and γ, various inter-cluster distances can be calculated, as listed in Table 1.

Table 1: Some commonly used values for the parameters in the Lance-Williams's formula

Algorithm	α_i	α_j	β	γ
Single-link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{-1}{2}$
Complete-link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group average	$\frac{\lvert C_i\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	$\frac{\lvert C_j\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	0	0
Weighted group average	$\frac{1}{2}$	$\frac{1}{2}$	0	0
Ward's method	$\frac{\lvert C_i\rvert+\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	$\frac{\lvert C_j\rvert+\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	$\frac{-\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	0
Centroid	$\frac{\lvert C_i\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	$\frac{\lvert C_j\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	$\frac{-\lvert C_i\rvert\lvert C_j\rvert}{(\lvert C_i\rvert+\lvert C_j\rvert)^2}$	0
Median	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{-1}{4}$	0

4 Classification of Clustering Algorithms

Xu and Tian [4] proposed a classification of traditional clustering methods (Table 5 in [4]) and one for modern clustering algorithms (Table 14 in [4]). The authors also compare the time complexity of different algorithms by class. In the remainder of the lecture notes, we select a few algorithm to study.

- Agglomerative Hierarchical Methods
- k-mean
- Gaussian mixture model
- ...

5 Agglomerative Hierarchical Methods

A hierarchical algorithm divides a dataset into a sequence of nested partitions [1]. Agglomerative hierarchical clustering starts with every single object in a single cluster. Then it repeats merging the closest pair of clusters according to some similarity criteria until all of the data are in one cluster.

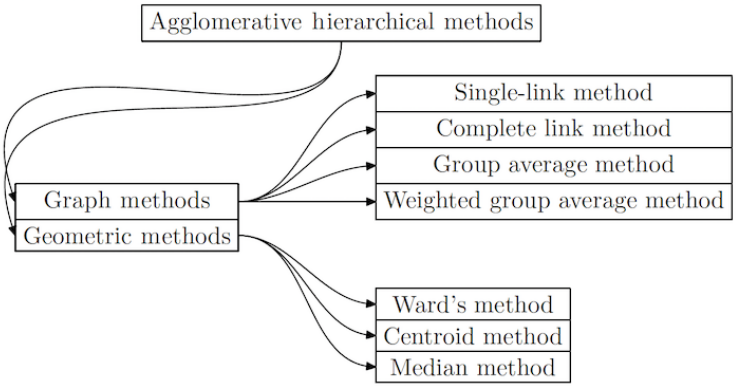

Figure 2: Some commonly used hierarchical methods (Figure 7.8 in [1])

Figure 3 is an example that we use to help explain agglomerative hierarchical methods.

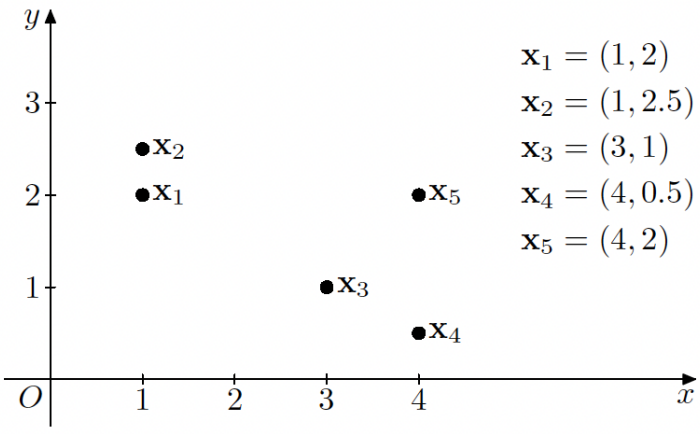

Figure 3: An example of five instances with 2-dimensional features (Figure 7.9 in [1])

5.1 Dendrogram

Dendrogram is a commonly used representation of hierarchical clusterings. Dendrogram is a hierarchically nested tree diagram on a dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$. It has M leaves that each is a cluster of a single instance, each internal node is associated with a height satisfying the condition:

$$h(A) \leq h(B) \Leftrightarrow A \subseteq B \tag{27}$$

for all subsets of data points A and B if $A \cap B \neq \emptyset$.

The heights in the dendrogram satisfy the following ultrametric conditions

$$h(\mathbf{x}_i, \mathbf{x}_j) \leq \max\{h(\mathbf{x}_i, \mathbf{x}_k), h(\mathbf{x}_j, \mathbf{x}_k)\} \quad \forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in D. \tag{28}$$

That is, $\mathbf{x}_i$ and $\mathbf{x}_j$ are the closest pair in the dataset, and the height is the distance between them.

Figure 4 illustrates an example of dendrogram with five instances.

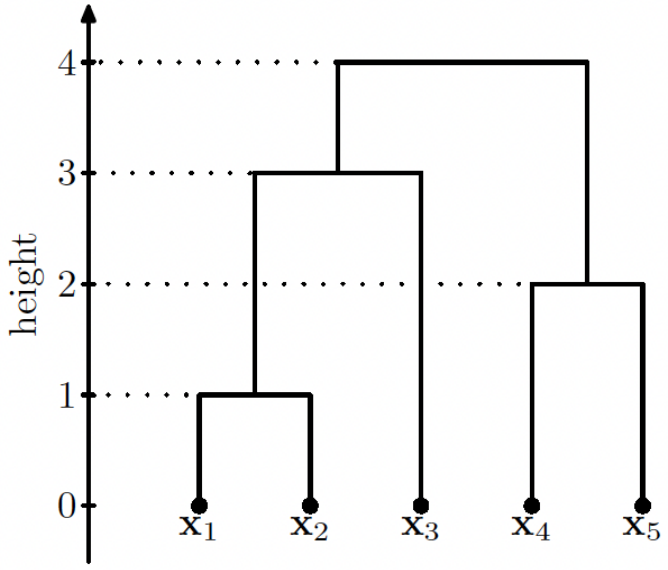

Figure 4: An example of dendrogram with five instances (Figure 7.3 in [1])

5.2 Single-link Method

The single-link method employs the nearest neighbor distance to measure the dissimilarity between two clusters. Let C_i, C_j, and C_k be three clusters of data points. Then the distance between C_k and $C_i \cup C_j$ can be obtained from the Lance-Williams formula

$$\begin{aligned} D(C_k, C_i \cup C_j) &= \frac{1}{2}D(C_k, C_i) + \frac{1}{2}D(C_k, C_j) - \frac{1}{2}|D(C_k, C_i) - D(C_k, C_j)| \\ &= \min\{D(C_k, C_i), D(C_k, C_j)\} \end{aligned} \tag{29}$$

which indicates that the inter-cluster distance is the nearest neighbor distance:

$$D(C, C') = \min_{\mathbf{x}\in C, \mathbf{y}\in C'} d(\mathbf{x}, \mathbf{y}). \tag{30}$$

Figure 5 is dendrogram produced by applying the single-link method to the example in Figure 3. Please show the dissimilarity matrix of each step and how you obtain this dendrogram.

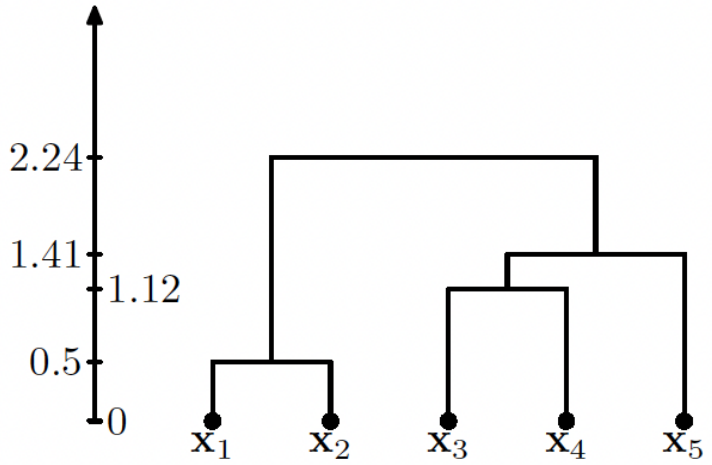

Figure 5: The dendrogram produced by applying the single-link method (Figure 7.10 in [1])

5.3 Complete Link Method

The complete link method uses the farthest neighbor distance to measure the dissimilarity between two clusters.

$$\begin{aligned} D(C_k, C_i \cup C_j) &= \frac{1}{2}D(C_k, C_i) + \frac{1}{2}D(C_k, C_j) + \frac{1}{2}|D(C_k, C_i) - D(C_k, C_j)| \\ &= \max\{D(C_k, C_i), D(C_k, C_j)\}, \end{aligned} \tag{31}$$

which indicates that the inter-cluster distance is the farthest neighbor distance:

$$D(C, C') = \max_{\mathbf{x}\in C, \mathbf{y}\in C'} d(\mathbf{x}, \mathbf{y}). \tag{32}$$

Figure 6 is dendrogram produced by applying the complete link method to the example in Figure 3. Please show the dissimilarity matrix of each step and how you obtain this dendrogram.

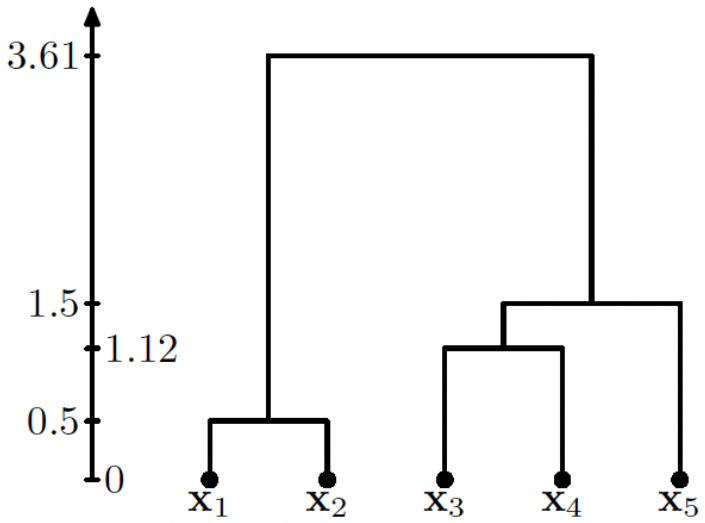

Figure 6: The dendrogram produced by applying the complete link method (Figure 7.11 in [1])

5.4 Group Average Method

The group average method measures the distance between two clusters using the average of the distances between all possible pairs of instances that are made up of one instance from each cluster:

$$D(C_k, C_i \cup C_j) = \frac{|C_i|}{|C_i| + |C_j|} D(C_k, C_i) + \frac{|C_j|}{|C_i| + |C_j|} D(C_k, C_j) \tag{33}$$

where C_i, C_j, and C_k are three clusters in one level of clustering.

The inter-cluster distance is

$$D(C, C') = \frac{1}{|C||C'|} \Sigma(C, C') = \frac{1}{|C||C'|} \sum_{\mathbf{x}_l \in C, \mathbf{y}_m \in C'} d(\mathbf{x}_l, \mathbf{y}_m) \tag{34}$$

Therefore,

$$\begin{aligned} & D(C_k, C_i \cup C_j) \\ =& \frac{|C_i|}{|C_i| + |C_j|} D(C_k, C_i) + \frac{|C_j|}{|C_i| + |C_j|} D(C_k, C_j) \\ =& \frac{|C_i|}{|C_i| + |C_j|} \frac{1}{|C_i||C_k|} \Sigma(C_k, C_i) + \frac{|C_j|}{|C_i| + |C_j|} \frac{1}{|C_j||C_k|} \Sigma(C_k, C_j) \\ =& \frac{1}{(|C_i| + |C_j|)(|C_k|)} \Sigma(C_k, C_i \cup C_j) \end{aligned} \tag{35}$$

Figure 7 is dendrogram produced by applying the group average method to the example in Figure 3. Please show the dissimilarity matrix of each step and how you obtain this dendrogram.

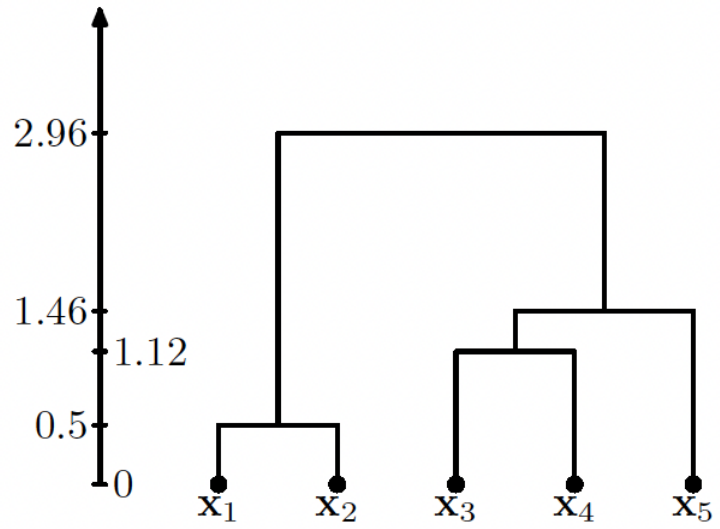

Figure 7: The dendrogram produced by applying the group average method (Figure 7.12 in [1])

5.5 Weighted Group Average Method

The group average method measures the distance between two clusters:

$$D(C_k, C_i \cup C_j) = \frac{1}{2} D(C_k, C_i) + \frac{1}{2} D(C_k, C_j) \tag{36}$$

Figure 8 is dendrogram produced by applying the group average method to the example in Figure 3. Please show the dissimilarity matrix of each step and how you obtain this dendrogram.

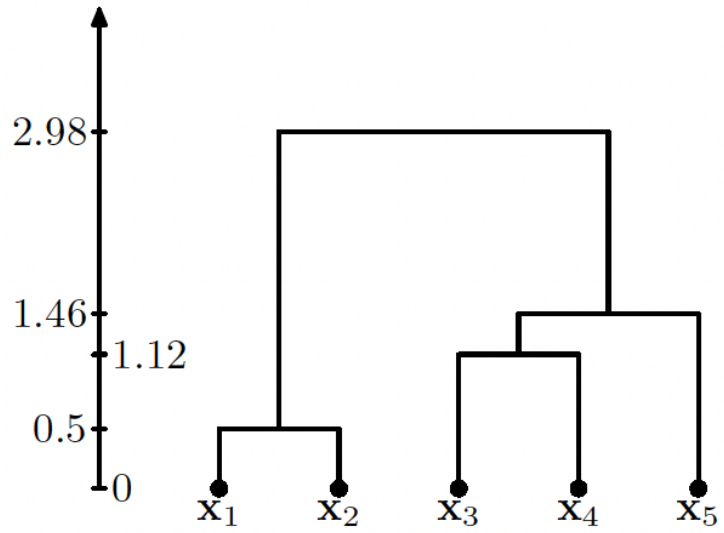

Figure 8: The dendrogram produced by applying the weighted group average method (Figure 7.13 in [1])

5.6 Centroid Method

According to the Lance-Williams formula:

$$\begin{aligned} & D(C_k, C_i \cup C_j) \\ =& \frac{|C_i|}{|C_i| + |C_j|} D(C_k, C_i) + \frac{|C_j|}{|C_i| + |C_j|} D(C_k, C_j) - \frac{|C_i||C_j|}{(|C_i| + |C_j|)^2} D(C_i, C_j) \end{aligned} \tag{37}$$

where C_i, C_j, and C_k are three clusters in one level of clustering.

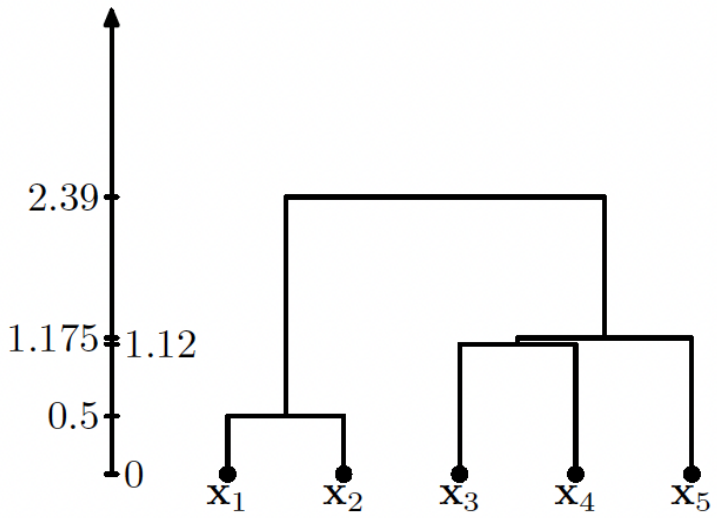

Figure 9: The dendrogram produced by applying the centroid method (Figure 7.14 in [1])

5.7 Median Method

$$
\begin{aligned}
&D(C_k, C_i \cup C_j) \\
=&\frac{1}{2}D(C_k, C_i) + \frac{1}{2}D(C_k, C_j) - \frac{1}{4}D(C_i, C_j)
\end{aligned}
\tag{38}
$$

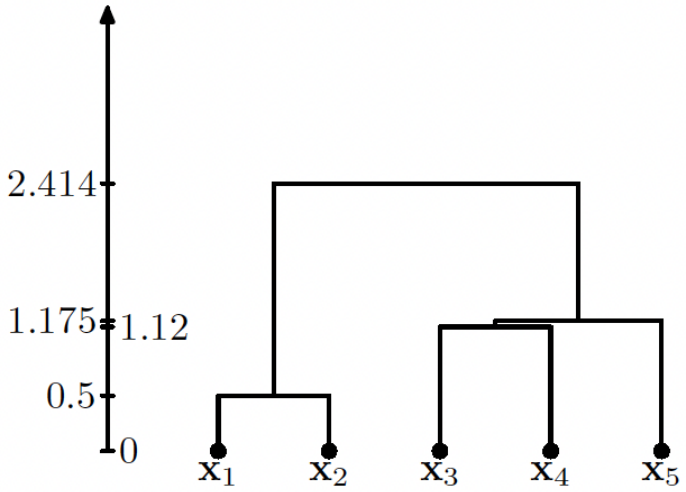

Figure 10: The dendrogram produced by applying the median method (Figure 7.15 in [1])

5.8 Ward's Method

The Ward's method seeks to minimize the *loss of information* associate with each merging. The information loss is measured as an error sum of squares (ESS). Therefore, Ward's method is often referred to as the "minimum variance" method.

Given a group of data points C, the ESS associated with C is given by

$$
\begin{aligned}
\mathrm{ESS}(C) &= \sum_{\mathbf{x}\in C} (\mathbf{X} - \boldsymbol{\mu}(C))(\mathbf{X} - \boldsymbol{\mu}(C))^{\mathrm{T}} \\
&= \sum_{\mathbf{x}\in C} (x)\mathbf{x}^{\mathrm{T}} - |C|\boldsymbol{\mu}(C)\boldsymbol{\mu}(C)^{\mathrm{T}}
\end{aligned}
\tag{39}
$$

If there are K groups $C_1, \ldots, C_K$ in one level of the clustering, the information loss is

$$
\mathrm{ESS} = \sum_{k=1}^{K} \mathrm{ESS}(C_k) \tag{40}
$$

which is the total within-group ESS.

If the squared Euclidean distance is used to compare the dissimilarity matrix, then the dissimilarity matrix can be updated by the Lance-Williams formula during the process of clustering:

$$
\begin{aligned}
&D(C_k, C_i \cup C_j) \\
=&\frac{|C_k| + |C_i|}{|C_i| + |C_j| + |C_k|} D(C_k, C_i) + \frac{|C_k| + |C_j|}{|C_i| + |C_j| + |C_k|} D(C_k, C_j) \\
&- \frac{|C_k|}{|C_i| + |C_j| + |C_k|} D(C_i, C_j).
\end{aligned}
\tag{41}
$$

During the clustering process, the increase of ESS by merging andy two clusters C_i and C_j is calculated as:

$$\Delta\text{ESS}_{i,j} = \frac{|C_i||C_j|}{|C_i| + |C_j|}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}. \tag{42}$$

If the two clusters are single-instance cluster,

$$\Delta\text{ESS}_{i,j} = \frac{1}{2}d_{i,j}. \tag{43}$$

The merge of cluster C_k into $C_t = C_i \cup C_j$ leads the an increase in ESS given by

$$\begin{aligned}\Delta\text{ESS}_{kt} &= \frac{|C_k||C_t|}{|C_k| + |C_t|}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_t)(\boldsymbol{\mu}_k - \boldsymbol{\mu}_t)^{\mathrm{T}} \\ &= \frac{|C_k||C_i|}{|C_k| + |C_t|}\Delta\text{ESS}_{k,i} + \frac{|C_k||C_j|}{|C_k| + |C_t|}\Delta\text{ESS}_{k,j} - \frac{|C_k|}{|C_k| + |C_t|}\Delta\text{ESS}_{i,j}.\end{aligned} \tag{44}$$

In the following, let's practice the clustering of the five instances. The clustering process is illustrated by the dendrogram below.

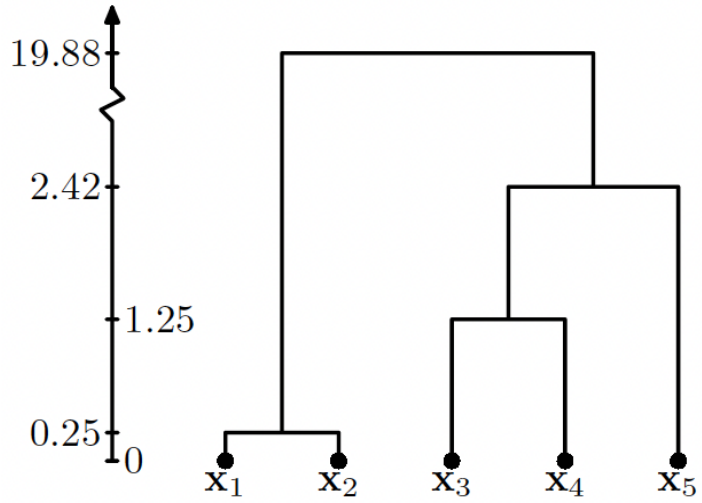

Figure 11: The dendrogram produced by applying the Ward's method (Figure 7.16 in [1])

6 Center-based Clustering Algorithms

Center-based algorithms are efficient for clustering large databases and high-dimensional databases. Center-based algorithms usually have their own objective functions, which define how good a clustering solution is. The goal of a center based algorithm is to minimize its objective function.

6.1 The K-mean Algorithm

k-mean is a partitional clustering method. It was designed for clustering numerical data in which each cluster has a center called the mean. The number of clusters k is fixed in this algorithm.

Let $\mathcal{D}$ be a dataset with M instances. Each instance is characterized by an N-dimensional feature vector. Let $C_1, \ldots, C_K$ be the disjointed K clusters and $\boldsymbol{\mu}(C_j)$ is the centroid of cluster C_j in the feature space. $d(\mathbf{x}, \boldsymbol{\mu}(C_j))$ is the distance from an instance $\mathbf{x}$ to the centroid of cluster j.

The Conventional K-mean Algorithm

In the initialization phase, the algorithm randomly assigns the instances into K clusters. In the iteration phase, the algorithm computes the distance between each instance and each cluster and assigns the instance to the nearest cluster.

Figure 12: K-mean algorithm -1 [1]

Require: Data set D, Number of Clusters k, Dimensions d:
{C_i is the ith cluster}
{1. Initialization Phase}
1: (C_1, C_2,…,C_k) = Initial partition of D.
{2. Iteration Phase}
2: **repeat**
3: d_{ij} = distance between case i and cluster j;
4: n_i = arg min$_{1 \le j \le k}$ d_{ij};
5: Assign case i to cluster n_i;
6: Recompute the cluster means of any changed clusters above;
7: **until** no further changes of cluster membership occur in a complete iteration
8: Output results.

Optimization-based Algorithm

We can formulate the k-means algorithm as an optimization problem:

$$\min_{W,Q} P(W, Q) = \sum_{l=1}^{K} \sum_{i=1}^{M} w_{i,l} d(\mathbf{x}_i, \boldsymbol{q}_j) \tag{45}$$

where $Q = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K\}$ are selected centers for the clusters. W is an $M \times K$ matrix satisfying the following conditions:

1. $w_{i,l} \in \{0, 1\}$ for $i = 1, \ldots, M$ and $l = 1, \ldots, K$.
2. $\sum_{l=1}^{K} w_{i,l} = 1$.

That is, an instance is assigned to the closest cluster but not others.

This optimization problem can be decomposed into two subproblems:

- Subproblem P1: Fix $Q = \widehat{Q}$ and solve the reduced problem $\min_W P(W, \widehat{Q})$:

$$w_{i,l} = \begin{cases} 1, & \text{if } d(\mathbf{x}_i, \boldsymbol{q}_l) = \min_{1 \le j \le K} d(\mathbf{x}_i, \boldsymbol{q}_j) \\ 0, & \text{otherwise} \end{cases} \tag{46}$$

  for $i = 1, \ldots, M$ and $l = 1, \ldots, K$.

- Subproblem P2: Fix $W = \widehat{W}$ and solve the reduced problem $\min_Q P(\widehat{W}, Q)$.

$$\boldsymbol{q}_l = \frac{\sum_{i=1}^{M} w_{i,l}\mathbf{x}_i}{\sum_{i=1}^{M} w_{i,l}} \tag{47}$$

The algorithm is presented below in 13.

Figure 13: K-mean algorithm -2 [1]

Require: Data set D, Number of Clusters k, Dimensions d:
1: Choose an initial Q^0 and solve $P(W, Q^0)$ to obtain W^0;
2: Let T be the number of iterations;
3: **for** $t = 0$ to T **do**
4: Let $\hat{W} \Leftarrow W^t$ and solve $P(\hat{W}, Q)$ to obtain Q^{t+1};
5: **if** $P(\hat{W}, Q^t) = P(\hat{W}, Q^{t+1})$ **then**
6: Output $\hat{W}$, Q^t;
7: Break;
8: **end if**
9: Let $\hat{Q} \Leftarrow Q^{t+1}$ and solve $P(W^t, \hat{Q})$ to obtain W^{t+1};
10: **if** $P(W^t, \hat{Q}) = P(W^{t+1}, \hat{Q})$ **then**
11: Output W^t, $\hat{Q}$;
12: Break;
13: **end if**
14: **end for**
15: Output W^{T+1}, Q^{T+1}.

Properties of the k-means Algorithm

The k-means algorithm has some important properties:

- It is efficient in clustering large data sets, since its computational complexity is linearly proportional to the size of the data sets.
- It often terminates at a local optimum.
- The clusters have convex shapes, such as a ball in three-dimensional space.
- It works on numerical data.
- The performance is dependent on the initialization of the centers.

6.2 The k-modes Algorithm

k-modes is an extension of the k-mean algorithms to handle categorical datasets. The centers of clusters are the mode of the clusters. The distance from an instance to the mode of a cluster is measured by a dissimilarity measure.

7 Gaussian Mixture Models

In a Gaussian mixture model (GMM), the data $D = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$ in an N-dimensional feature space are assumed to come from a distribution with a density function being a mixture of K Gaussian distributions:

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k \Phi(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \tag{48}$$

where p_k's are the mixing proportions. That is, $0 \leq p_k \leq 1$ and $\sum_{k=1}^{K} p_k = 1$. $\Phi(\mathbf{x}|\boldsymbol{\mu}, \Sigma)$ is the density of Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ:

$$\Phi(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right] \tag{49}$$

The mixture approach aims to maximize the log likelihood function

$$L(\boldsymbol{\theta}|\mathbf{x}_1,\ldots,\mathbf{x}_M) = \sum_{i=1}^{M} \ln \sum_{k=1}^{K} p_k \Phi(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k), \tag{50}$$

where $\boldsymbol{\theta}$ is the collection of $[\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K]$,$[\Sigma_1,\ldots,\Sigma_K]$, and $[p_1,\ldots,p_K]$. That is, given a chosen number of clusters K and for each cluster k, we estimate

- the mean vector $\boldsymbol{\mu}_k$,
- the covariance matrix Σ_k, and
- the mixing proportion p_k

to maximize the log likelihood function.

There are various algorithms for solving GMM, such as Expectation-Maximization (EM) Algorithm, Variational Inference (VI), Gibbs Sampling, among others. In this learning module, we briefly introduce the EM algorithm.

The EM algorithm maximizes the log-likelihood in an iterative approach:

- **Initialization**: we can use a random assignment method or a K-mean method to initialize the K clusters. Thereby, we have the initial estimation of $\{\widehat{\boldsymbol{\mu}}_k, \widehat{\Sigma}_k, \widehat{p}_k\}_{k=1}^{K}$.
- **E-step (Expectation)**: Calculate the posterior probability that the instance $\mathbf{x}_i$ comes from the kth cluster, $\widehat{u}_{i,k}$, for any k, according to Bayes's rule:

$$\widehat{u}_{i,k} = \frac{\widehat{p}_k \Phi(\mathbf{x}_i|\widehat{\boldsymbol{\mu}}_k, \widehat{\Sigma}_k)}{\sum_{l=1}^{K} \widehat{p}_l \Phi(\mathbf{x}_i|\widehat{\boldsymbol{\mu}}_l, \widehat{\Sigma}_l)}. \tag{51}$$

- **M-step (Maximization)**: Based on the posterior probabilities, the clustering is updated:

$$C_k = \{\mathbf{x}_i : \widehat{u}_{i,k} = \arg\max_l \widehat{u}_{i,l}\}, \quad \forall k = 1,\ldots,K. \tag{52}$$

  That is, instance $\mathbf{x}_i$ is allocated to the cluster that it attains the highest posterior probability.

  After that, we update the estimations of model parameters:

$$\widehat{p}_k = \frac{1}{M} \sum_{i=1}^{M} \widehat{u}_{i,k}, \tag{53}$$

$$\widehat{\boldsymbol{\mu}}_k = \frac{1}{M} \sum_{i=1}^{M} \frac{\widehat{u}_{i,k}}{\widehat{p}_k} \mathbf{x}_{i,k}, \tag{54}$$

$$\widehat{\Sigma}_k = \frac{1}{M} \sum_{i=1}^{M} \frac{\widehat{u}_{i,k}}{\widehat{p}_k} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_k)^{\mathrm{T}}. \tag{55}$$

- **Convergence Check**: If the change in the log likelihood function is less than a pre-defined threshold, we stop. Otherwise, we repeat E-step and M-step until convergence.

The modeling approach above indicates GMMs do probabilistic soft assignment, unlike K-mean methods perform hard assignment, in clustering.

Similar to K-mean method, the number of clusters are pre-assumed. The choice of number of clusters is also a decision that needs to be made.

8 Performance Evaluation

8.1 Inertia

Inertia is the sum of squared distances of samples to their closest cluster center, weighted by the sample weights if provided. It is a measure of how **compact** the clusters are, with **lower inertia values indicating**

tighter and more cohesive clusters.

$$\text{Inertia} = \sum_{i=1}^{M}\sum_{k=1}^{K} w_{i,k}\|\mathbf{x}_i - \boldsymbol{\mu}(C_k)\|^2 \tag{56}$$

where

- M: the number of instances
- K: the number of clusters
- $\mathbf{x}_i$: an instance
- $\boldsymbol{\mu}(C_k)$: the mean of cluster k
- $w_{i,k}$: a binary variable indicating the assignment of instance i to cluster k. It takes the value 1 if instance i is assigned to cluster k and 0 otherwise. That is, $w_{i,k} \in \{0, 1\}$ and $\sum_{k=1}^{K} w_{i,k} = 1$, for $i = 1, \ldots, M$.

The smaller the Inertia value, the better the clustering results.

8.2 Rand Indices

8.2.1 Rand Index

The Rand Index (RI) is a measure that evaluates the **similarity between two clusterings**. It compares the agreement between the pairs of instances in the true clustering and the predicted clustering. RI is defined as the sum of the number of true positive instances(pairs of instances that are in the same cluster in both the true and predicted clusterings) and the number of true negative pairs (pairs of instances that are in different clusters in both the true and predicted clusterings), divided by the total number of pairs.

$\mathbf{X}$ is a dataset with M instances. $C = \cup_k C_k$ is the true clustering and $\widehat{C} = \cup_l \widehat{C}_l$ is the predicted clustering. Rand Index is calculated as

$$\text{RI} = \frac{a + b}{\binom{M}{2}} \tag{57}$$

where

- a is the number of pairs in the same clusters in both true and predicted clusterings

$$a = |P_{TP}|, \text{where } P_{TP} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i, \mathbf{x}_j \in C_k; \mathbf{x}_i, \mathbf{x}_j \in \widehat{C}_l\} \tag{58}$$

- b is the number of pairs in different clusters in both true and predicted clusterings,

$$b = |P_{TN}|, \text{where } P_{TN} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \in C_k, \mathbf{x}_j \in C_{k'}; \mathbf{x}_l \in \widehat{C}_l, \mathbf{x}_l \in \widehat{C}_{l'}\} \tag{59}$$

- $\binom{M}{2}$ is the total number of pairs of instances.

The Rand Index value ranges from 0 to 1. 1 means perfect agreement between the two clusterings. 0 means no agreement (or completely random clustering). Therfore, the larger the RI value, the higher the similarity between the two clusterings.

8.2.2 Adjusted Rand Index

The Rand index does not ensure to obtain a value close to 0 for a random labelling. The Adjusted Rand Index (ARI) adjusts RI to account for chance. It corrects for the expected similarity between two random clusterings.

Below is the contingency table for the clustering result.

The ARI is calculated by

$$\text{ARI} = \frac{\text{RI} - \text{E(RI)}}{\text{Max(RI)} - \text{E(RI)}}, \tag{60}$$

	$\widehat{C}_1$	$\dots$	$\widehat{C}_L$	total
C_1	$M_{1,1}$	$\dots$	$M_{1,L}$	M_1
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
C_K	$M_{K,1}$	$\dots$	$M_{K,L}$	M_K
total	$\widehat{M}_1$	$\dots$	$\widehat{M}_L$	M

where

$$\mathrm{RI} = \sum_{k,l} \binom{M_{k,l}}{2}, \tag{61}$$

$$\mathrm{E(RI)} = \frac{\sum_k \binom{M_k}{2} \sum_l \binom{\widehat{M}_l}{2}}{\binom{M}{2}}, \tag{62}$$

$$\mathrm{Max(RI)} = \frac{\sum_k \binom{M_k}{2} + \sum_l \binom{\widehat{M}_l}{2}}{2}. \tag{63}$$

The adjusted RI take ranges from -1 to 1., where 1 indicates perfect similarity, 0 indicates the similarity expected by chance, and negative values indicate less similarity than expected by chance.

8.3 Mutual Information

8.3.1 Normalized Mutual Information

Normalized Mutual Information (NMI) is a metric that measures the **similarity between two clusterings**, taking into account the entropy of the individual clusterings.

$$\mathrm{NMI} = \frac{\mathrm{MI}(C, \widehat{C})}{\sqrt{H(C)H(\widehat{C})}} \tag{64}$$

where

- $\mathrm{MI}(C, \widehat{C})$ is the mutual information between the true clustering C and the predicted clustering $\widehat{C}$:

$$\begin{aligned} \mathrm{MI}(C, \widehat{C}) &= \sum_{k=1}^{K} \sum_{l=1}^{L} P(C_k \cap \widehat{C}_l) \log \left(\frac{P(C_k \cap \widehat{C}_l)}{P(C_k) P(\widehat{C}_l)} \right) \\ &= \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{M_{k,l}}{M} \log \left(\frac{M_{k,l} M}{M_k \widehat{M}_l} \right). \end{aligned} \tag{65}$$

- $H(C)$ is the entropy of true clustering:

$$H(C) = -\sum_{k=1}^{K} P(C_k) \log P(C_k) = -\sum_{k=1}^{K} \frac{M_k}{M} \log \left(\frac{M_k}{M} \right). \tag{66}$$

- $H(\widehat{C})$ is the entropy of predicted clustering:

$$H(\widehat{C}) = -\sum_{l=1}^{L} P(\widehat{C}_l) \log P(\widehat{C}_l) = -\sum_{l=1}^{L} \frac{\widehat{M}_l}{M} \log \left(\frac{\widehat{M}_l}{M} \right). \tag{67}$$

The MI value ranges from 0 to ∞. The value depends on the sizes of clusters. MI can become arbitrarily large if we have lots of clusters.

8.3.2 Adjusted Mutual Information

Adjusted Mutual Information (AMI) is a variation of Mutual Information (MI) that has been adjusted to account for chance. It is used to measure the agreement between two clusterings, considering the entropy of each clustering.

$$\text{AMI} = \frac{\text{MI} - \text{E}[\text{MI}]}{\text{Max}[H(C), H(\widehat{C})] - \text{E}(\text{MI})} \tag{68}$$

where

- MI is the mutual information defined in eq. (65).
- E[MI] is the expected mutual information

$$\text{E}[\text{MI}] = \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{M_{k,l}=(C_k+\widehat{C}_l-M)^+}^{\min[C_k,\widehat{C}_l]} \frac{M_{k,l}}{M}\log\left(\frac{M_{k,l}M}{M_k,\widehat{M}_l}\right)\cdot \frac{M_k!\,\widehat{M}_l!(M-M_k)!(M-\widehat{M}_l)!}{M!M_{k,l}!(M_k-M_{k,l})!(\widehat{M}_l-M_{k,l})!(M-M_k-M_l+M_{k,l})!} \tag{69}$$

- $H(C)$ is the entropy of true clustering defined in eq. (66).
- $H(\widehat{C})$ is the entropy of predicted clustering defined in eq. (67).

The adjusted MI value ranges from 0 to 1. 1 means perfect agreement between the two clusterings, and 0 means no agreement.

8.4 Homogeneity, Completeness and V-measure

8.4.1 Homogeneity

Homogeneity is a metric that measures **the extent to which all clusters contain only data points that are members of a single class**. It is calculated as

$$\text{homo} = 1 - \frac{H(C|\widehat{C})}{H(C)} \tag{70}$$

where

- $H(C)$ is the entropy of true clustering defined in eq. (66).
- $H(C|\widehat{C})$ is the entropy of the conditional entropy of the true clustering given the predicted clustering,

$$H(C|\widehat{C}) = -\sum_{l=1}^{L} P(\widehat{C}_l)H(C|\widehat{C}_l) = -\sum_{l=1}^{L}\frac{\widehat{M}_l}{M}\sum_{k=1}^{K}\frac{M_{k,l}}{\widehat{M}_l}\log\left(\frac{M_{k,l}}{\widehat{M}_l}\right). \tag{71}$$

The homogeneity value ranges from 0 to 1. 1 means perfect homogeneity (i.e., each cluster contains only data from one class), and 0 means completely mixed clusters (i.e., each cluster mixes data points from all classes). The higher the homogeneity value, the better the clustering result.

8.4.2 Completeness

Completeness is a metric that measures **the extent to which all members of a given class are assigned to the same cluster**. It is calculated as

$$\text{cmpl} = 1 - \frac{H(\widehat{C}|C)}{H(\widehat{C})} \tag{72}$$

where

- $H(\widehat{C})$ is the entropy of the predicted clustering calculated in eq.(67).

- $H(\widehat{C}|C)$ is the entropy of the conditional entropy of the predicted clustering given the true clustering,

$$H(\widehat{C}|C) = -\sum_{k=1}^{K} P(C_k) H(\widehat{C}|C_k) = -\sum_{k=1}^{K} \frac{M_k}{M} \sum_{l=1}^{L} \frac{M_{k,l}}{M_K} \log\left(\frac{M_{k,l}}{M_K}\right). \tag{73}$$

The completeness value ranges from 0 to 1. 1 means perfect perfect completeness (i.e., each class is fully contained in one cluster). 0 means class members scattered across many clusters. Therefore, the larger the completeness value, the better the clustering result.

8.4.3 V-measure

V-measure is a metric that **combines both homogeneity and completeness** to provide a single measure of clustering quality.

$$\text{V-measure} = \frac{(1+\beta)\text{homo} \times \text{cmpl}}{\beta\,\text{homo} + \text{cmpl}}. \tag{74}$$

where $\beta = 1$ is a default value.

The V-measure value ranges from 0 to 1. 1 means perfect agreement between the two clusterings, and 0 means no agreement. Therefore, the larger the value, the better the clustering result.

8.5 Silhouette Score

The silhouette score measures **how similar an instance is to its own cluster compared to other clusters**.

$\mathbf{X}$ is the dataset with M instances assigned to $L(\geq 2)$ clusters $\widehat{C} = \cup_{l=1}^{L} \widehat{C}_l$. Silhouette score for an instance $\mathbf{x}_i \in C_l$ is calculated as

$$s_i = \frac{b_i - a_i}{\max[a_i, b_i]} \tag{75}$$

where

- a_i: the mean distance from $\mathbf{x}_i$ to other instances in its own cluster

$$a_i = \frac{1}{|C_l| - 1} \sum_{\mathbf{x}_j \in C_l, \mathbf{x}_j \neq \mathbf{x}_i} d(\mathbf{x}_i, \mathbf{x}_j). \tag{76}$$

- b_i: the minimum mean distance from $\mathbf{x}_i$ to other clusters

$$b_i = \min_{l' \neq l} \frac{1}{|C_{l'}|} \sum_{\mathbf{x}_j \in C_{l'}} d(\mathbf{x}_i, \mathbf{x}_j). \tag{77}$$

The silhouette score ranges from -1 to +1. A high value indicates that an instance is well matched to its own cluster and poorly matched to neighboring clusters:

- A score close to 1 indicates $\mathbf{x}_i$ is very compact within the cluster to which it belongs and far away from the other clusters.
- A score around 0 means the instance is on or very close to the decision boundary between two neighboring clusters.
- A score less than 0 indicates the instance might have been assigned to the wrong cluster.

The silhouette score for the entire dataset $\mathbf{X}$ is

$$\text{silhouette} = \frac{1}{M} \sum_{i=1}^{M} s_i, \tag{78}$$

which is useful for determining the optimal number of clusters in a dataset. It reaches its highest value when clusters are well-defined and compact, and it's lowest when the clusters overlap.

References

[1] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2020.

[2] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[3] Maurice George Kendall. Rank correlation methods. 1948.

[4] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193, 2015.

[5] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.