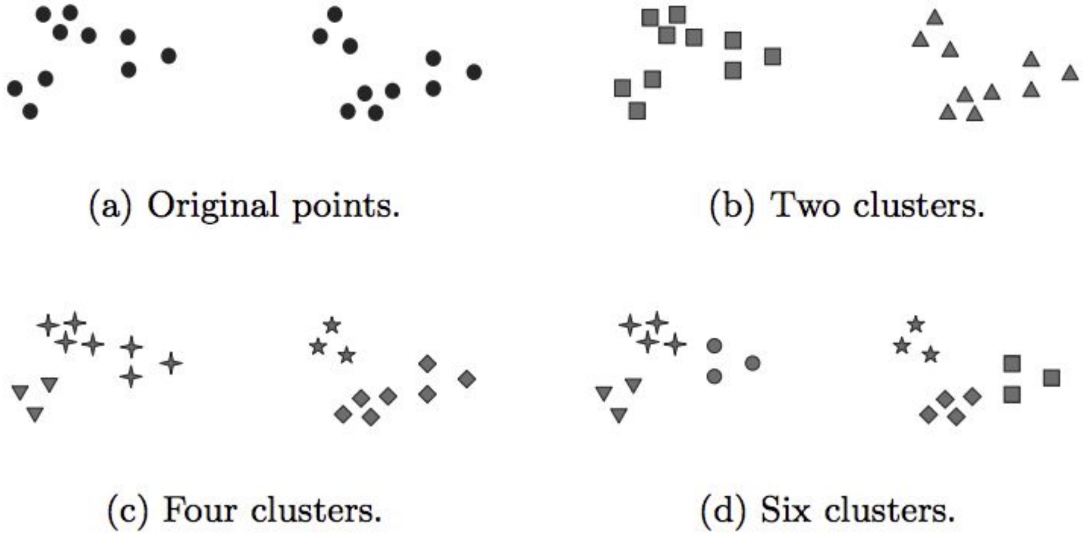
Data Mining Algorithms: Clustering

Department of Computer Science and Engineering
Kathmandu University

Data clustering

- Also called **clustering**, **cluster analysis, segmentation analysis, taxonomy analysis, or unsupervised classification**:
- It aims at creating groups of objects, or clusters, such that objects within a cluster are very similar and different from the objects in other clusters.
- Clustering techniques apply when there is no class to be predicted but the instances are to be divided into natural groups.
- For large datasets, designing a classification scheme by hand is unfeasible.
- In that context, data clustering is the process which aims at automatically discovering a classification scheme in order to organize the objects of a large database.

Data clustering


(a) Original points.

(b) Two clusters.

(c) Four clusters.

(d) Six clusters.

Source: Tan et al., 2016

The definition of a cluster is imprecise.

The best definition depends on the nature of data and the desired results.

Applications of clustering

Market research: To segment the market and determine target markets.

Biology: To find groups of genes that have similar functions.

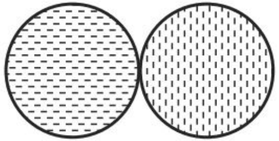
Information retrieval: To cluster the results provided by a search engine in order to organize the web pages according to topics. User can browse the retrieved items in a more efficient way.

Image processing: To segment a gray-scale or a color image in order to detect objects represented in the image and/or to compress the image.

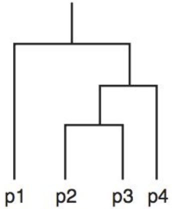
Online social network analysis (such as Facebook, LinkedIn,...): graph data clustering is used in order to detect communities among people.

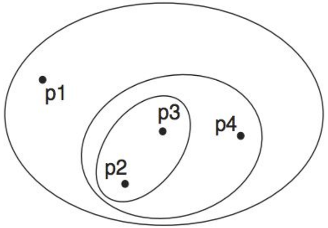
Different types of classification scheme

- A flat partition: A set of clusters or segments

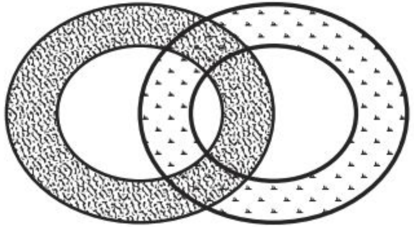


- A hierarchical tree or taxonomy: a set of nested partitions





- Hard or soft memberships to clusters
  - **Hard**: each object belongs to only one cluster
  - **Soft (or fuzzy)**: each object belongs to each cluster to a certain degree



Clustering methods

- Partitional clustering
  - Divides the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.
- Hierarchical clustering
  - A set of nested clusters that are organized as a tree.
- Overlapping clustering
  - Data objects can simultaneously belong to more than one cluster.

Partitional Clustering

Partitional methods

- Given a set of n objects, **construct k partitions of the data**, where each partition represents a cluster and k $\leq$ n
- Partitional methods are commonly **distance-based** and find **mutually exclusive clusters** of spherical shape
- The general criterion of a good partitioning (or **objective function**) is that objects in the same cluster are "close" or related to each other (i.e. **high intra-cluster similarity**), whereas objects in different clusters are "far apart" or very different (i.e. **low inter-cluster similarity**)
- Examples: k-means, k-medoids algorithms

K-Means

- **Given**: a data set D
- **Goal**: distribute the objects in D into k disjoint clusters, C_1, C_2, … , C_k , i.e., $C_i \subset$ D and $C_i \cap C_j = \varnothing$ for ($1 \leq i, j \leq k$)
- Each of the k clusters are represented by the mean of the objects (called the **centroid**) within it.
- Basic algorithm
    a. Select k points as initial cluster centroids.
    b. Repeat
        i. Assign each point to its nearest centroid.
        ii. For each cluster, recalculate its centroid based on which instances it contains
    c. Until convergence (i.e., centroids do not change)

K-Means

Assigning points to the closest centroid

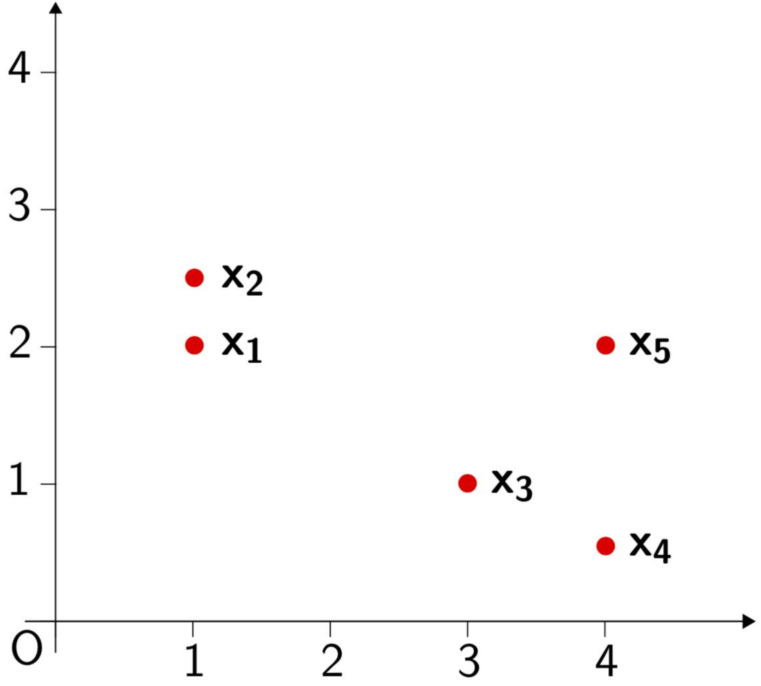
- We need a proximity measure that quantifies the notion of “closest” for the specific data under consideration.
  - Euclidean distance is commonly used for data points in Euclidean space.
  - Cosine similarity is more appropriate for documents.
  - Other measures: Manhattan distance, Jaccard measure etc.

K-Means: Example

We consider 5 data points in $\mathbb{R}^2$:

- $\mathbf{x_1} = (1, 2)$
- $\mathbf{x_2} = (1, 2.5)$
- $\mathbf{x_3} = (3, 1)$
- $\mathbf{x_4} = (4, 0.5)$
- $\mathbf{x_5} = (4, 2)$

We consider the euclidean distance between data points.



K-Means: Example

See class notes.

K-Means

Choosing initial centroids (seeds)

- The results of the K-means method depend strongly on the initial centroids.
- The goal of the clustering is typically expressed by an objective function. The quality of a clustering is measured by the objective function.
  - For data points in Euclidean space, the goal is to minimize the sum of the squared error (SSE).

    $$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2$$

  - Given two different sets of clusters that are produced by two different runs of K-means, we prefer the one with the smallest SSE.

K-Means

Choosing initial centroids (seeds)

- Different ways to generate the initial k centroids:
    - Pick first k points in D
    - Pick k points randomly from D
    - Pick top k widely-separated points etc.
- Randomly selected initial centroids may be poor.
    - Run K-means algorithm multiple times. In each run, choose a different set of randomly chosen initial centroids.
    - Then select the set of clusters with the minimum SSE.

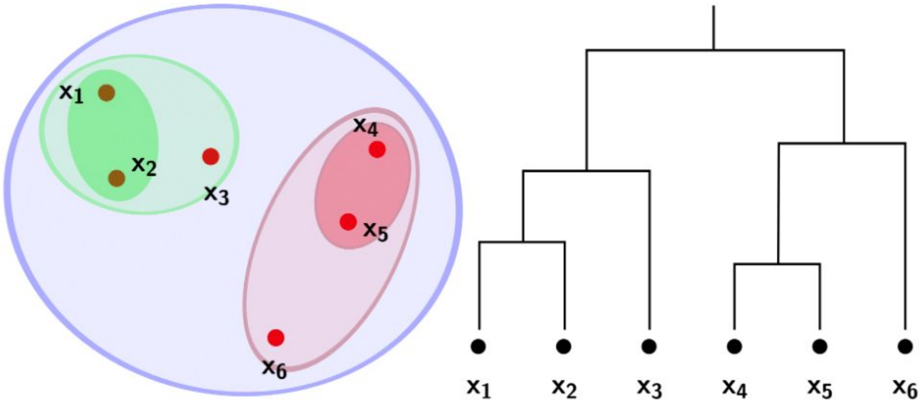
Limitations of k-Means

- Makes hard assignments of observations to clusters.
- Sensitive to initial seeds and outliers
- Does not consider cluster shapes

Hierarchical Clustering

Hierarchical Clustering

- Produces a nested series of clusters.
- Allows clusters to be found at different levels of granularity.
- A hierarchical clustering is often displayed graphically using a tree-like diagram, called **dendogram**.

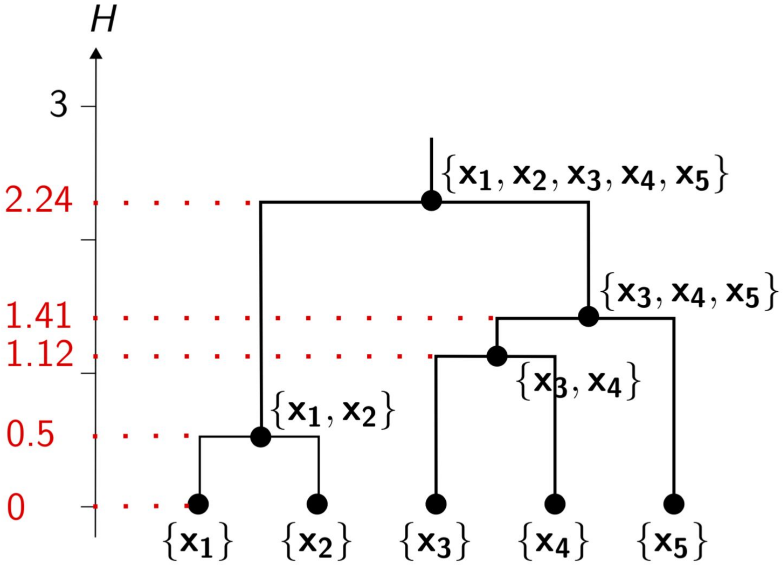


Dendogram

A dendogram is a binary tree in which each internal node is associated with a height H satisfying the condition:

H(n) $\leq$ H(n') $\Leftrightarrow$ n $\subset$ n';

Where
n and n' are two nodes of the binary tree
H(n) is the distance at which the cluster associated with n is created

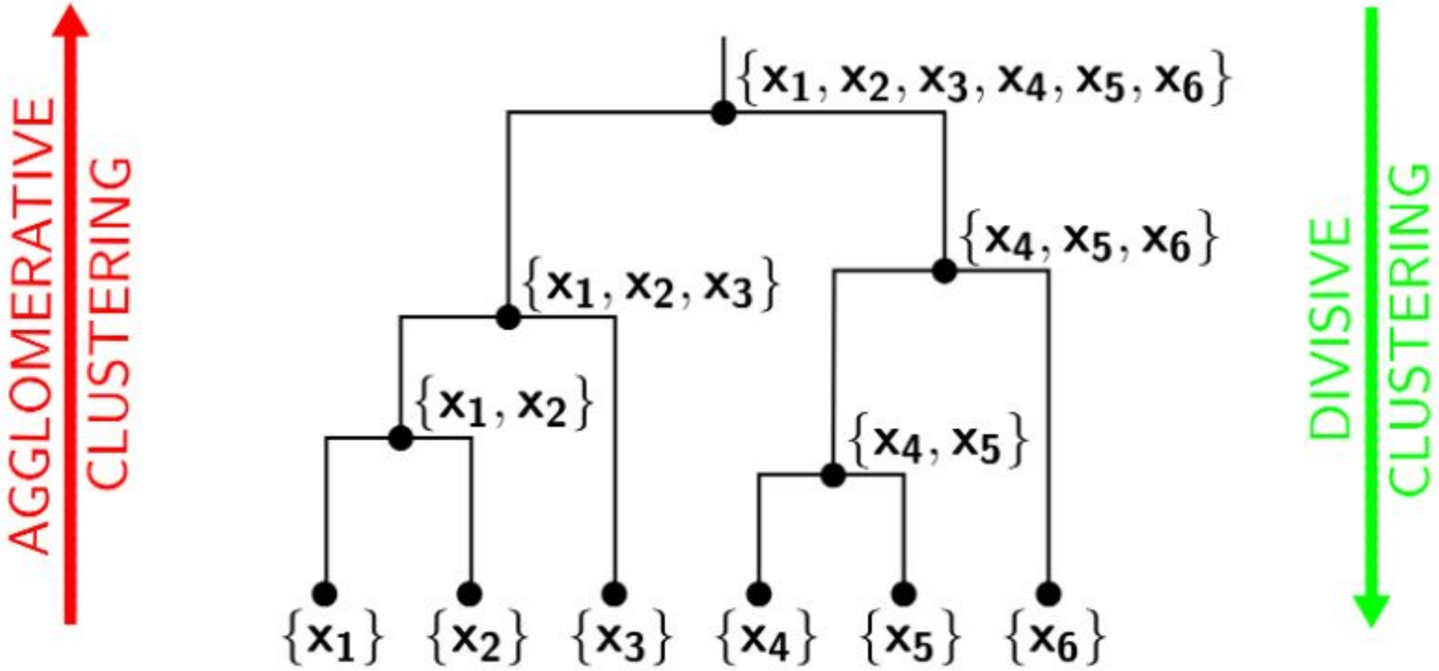


Hierarchical Clustering

Approaches:

1. Agglomerative (bottom-up)
   a. Start with each object forming a separate group
   b. Then successively merge the objects or groups close to one another, until all groups are merged into one, or a termination condition holds
2. Divisive (top-down)
   a. Start with all the objects in the same cluster
   b. Successively split a cluster into smaller clusters, until eventually each object is in one cluster, or a termination condition holds

Hierarchical methods



Agglomerative hierarchical clustering (AHC)

Basic AHC algorithm:

1. Initialize the tree representation with n leaves
2. While not all data points are grouped together
   a. Merge the two closest clusters according to some distance measure
   b. Add a parent node in the tree representation accordingly

The critical point for AHC algorithms is the **distance measure between clusters**.

Distance between clusters

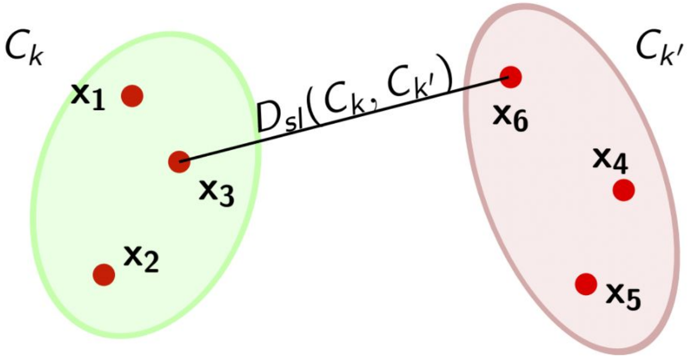
We can distinguish AHC algorithms according to the type of **cluster proximity measures** used

- Single link method
- Complete link method
- Group average method (UPGMA)
- Centroid method
- Ward’s method

Cluster proximity measures: Single link method

The distance between two clusters is the minimum distance between a point in the first cluster and a point in the second cluster

$$D_{sl}(C_k, C_{k'}) = \min_{\mathbf{x} \in C_k, \mathbf{y} \in C_{k'}} \{D(\mathbf{x}, \mathbf{y})\}$$

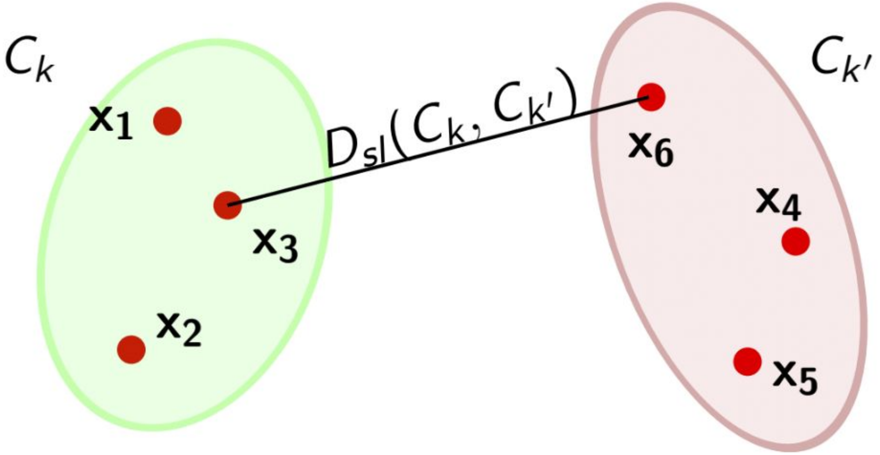


Single link method

In graph terms, it is the shortest edge between two nodes in different subsets of nodes

Also known as the **nearest neighbor method**, since it employs the nearest neighbor to measure the dissimilarity between two clusters

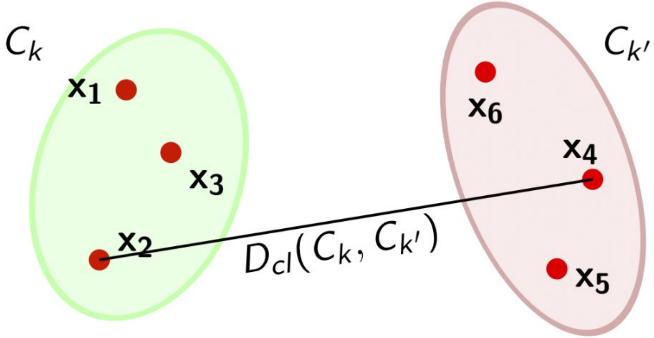
$$D_{sl}(C_k, C_{k'}) = \min_{\mathbf{x} \in C_k, \mathbf{y} \in C_{k'}} \{D(\mathbf{x}, \mathbf{y})\}$$



Cluster proximity measures: Complete link method

It uses the farthest neighbor to measure the dissimilarity between two clusters

$$D_{cl}(C_k, C_{k'}) = \max_{\mathbf{x}\in C_k, \mathbf{y}\in C_{k'}} \{D(\mathbf{x}, \mathbf{y})\}$$

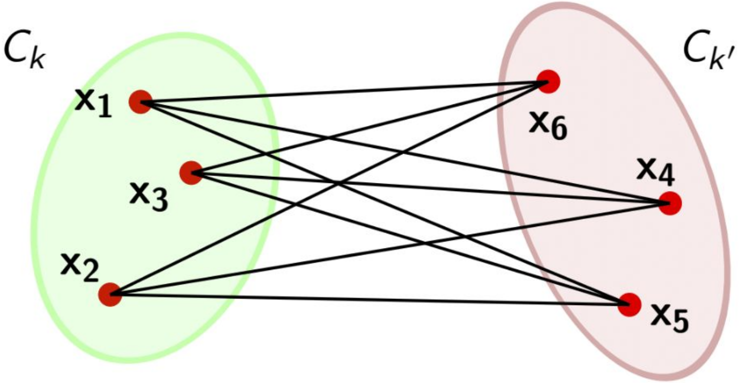


Cluster proximity measures: Group average method

Aka UPGMA (Unweighted Pair Group Method using Arithmetic mean)

Defines cluster proximity to be the average pairwise proximities (average length of edges) for all pairs of points from different clusters

$$D_{upgma}(C_k, C_{k'}) = \frac{1}{|C_k||C_{k'}|} \sum_{\mathbf{x}\in C_k, \mathbf{y}\in C_{k'}} D(\mathbf{x}, \mathbf{y})$$



Cluster proximity measures (Contd.)

Centroid method

- Distance between two clusters is the distance between their centers

Ward’s method

- Cluster proximity is measured in terms of the increase in the sum of squared error (SSE) that results from merging the two clusters
- Ward’s method tries to minimize the sum of the squared distances of points from their cluster centers

Agglomerative hierarchical clustering (AHC)

Steps:

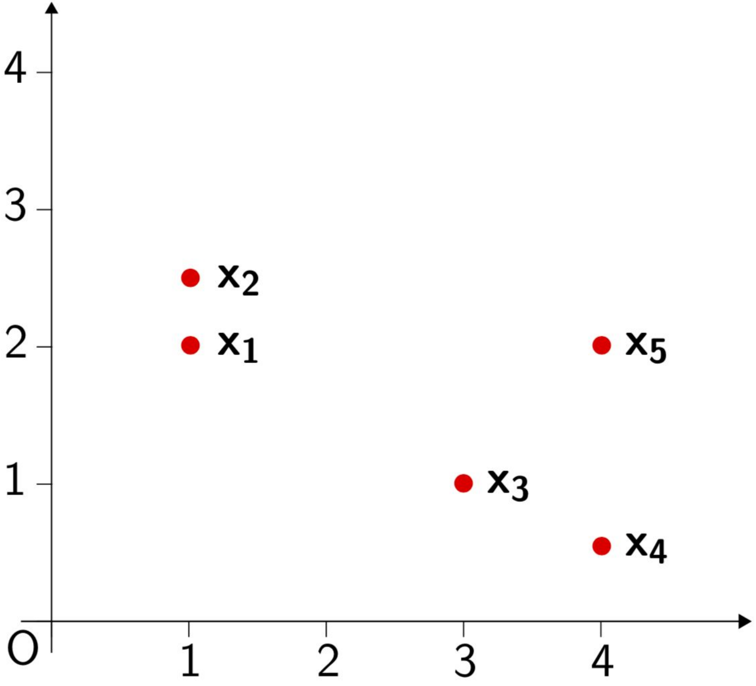
1. Assign each object to a separate cluster
2. Evaluate all pairwise **distances between clusters**
3. Construct a distance matrix using the distance values
4. Look for the pair of clusters with the shortest distance
5. Remove the pair from the matrix and merge them
6. Evaluate all distances from this new cluster to all other clusters, and update the matrix
7. Repeat until the distance matrix is reduced to a single element

AHC: Example (using single link method)

We consider 5 data points in $\mathbb{R}^2$:

- $\mathbf{x_1} = (1, 2)$
- $\mathbf{x_2} = (1, 2.5)$
- $\mathbf{x_3} = (3, 1)$
- $\mathbf{x_4} = (4, 0.5)$
- $\mathbf{x_5} = (4, 2)$

We consider the euclidean distance between data points.



AHC: Example (using single link method)

$$dend(h) = \{\ \{\mathbf{x_1}\}, \{\mathbf{x_2}\}, \{\mathbf{x_3}\}, \{\mathbf{x_4}\}, \{\mathbf{x_5}\} \quad \text{if } 0 \leq h$$

The starting distance matrix **D** is the euclidean distance matrix between points :

$$\mathbf{D} = \mathbf{D}_{\mathbf{eucl}} = \mathbf{D}_{\mathbf{sl}} = \begin{array}{cccccc} & \mathbf{x_1} & \mathbf{x_2} & \mathbf{x_3} & \mathbf{x_4} & \mathbf{x_5} \\ \mathbf{x_1} & 0 & 0.5 & 2.24 & 3.35 & 3 \\ \mathbf{x_2} & 0.5 & 0 & 2.5 & 3.61 & 3.04 \\ \mathbf{x_3} & 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ \mathbf{x_4} & 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ \mathbf{x_5} & 3 & 3.04 & 1.41 & 1.5 & 0 \end{array}$$

AHC: Example (using single link method)

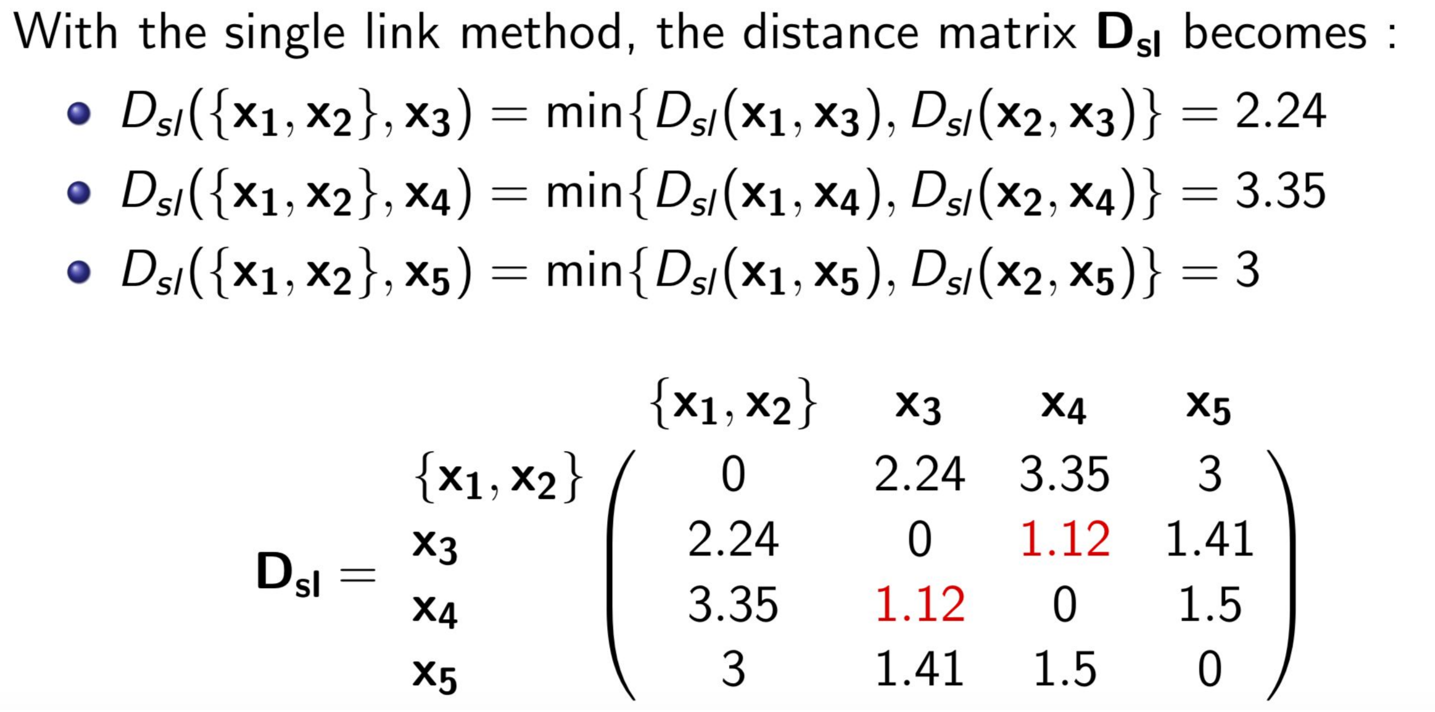
Merge $\mathbf{x_1}$ and $\mathbf{x_2}$

$$dend(h) = \begin{cases} \{\mathbf{x_1}\}, \{\mathbf{x_2}\}, \{\mathbf{x_3}\}, \{\mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x_1}, \mathbf{x_2}\}, \{\mathbf{x_3}\}, \{\mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 0.5 \leq h \end{cases}$$

With the single link method, the distance matrix $\mathbf{D_{sl}}$ becomes :

- $D_{sl}(\{\mathbf{x_1}, \mathbf{x_2}\}, \mathbf{x_3}) = \min\{D_{sl}(\mathbf{x_1}, \mathbf{x_3}), D_{sl}(\mathbf{x_2}, \mathbf{x_3})\} = 2.24$
- $D_{sl}(\{\mathbf{x_1}, \mathbf{x_2}\}, \mathbf{x_4}) = \min\{D_{sl}(\mathbf{x_1}, \mathbf{x_4}), D_{sl}(\mathbf{x_2}, \mathbf{x_4})\} = 3.35$
- $D_{sl}(\{\mathbf{x_1}, \mathbf{x_2}\}, \mathbf{x_5}) = \min\{D_{sl}(\mathbf{x_1}, \mathbf{x_5}), D_{sl}(\mathbf{x_2}, \mathbf{x_5})\} = 3$

AHC: Example (using single link method)



With the single link method, the distance matrix $\mathbf{D_{sl}}$ becomes :

- $D_{sl}(\{\mathbf{x_1}, \mathbf{x_2}\}, \mathbf{x_3}) = \min\{D_{sl}(\mathbf{x_1}, \mathbf{x_3}), D_{sl}(\mathbf{x_2}, \mathbf{x_3})\} = 2.24$
- $D_{sl}(\{\mathbf{x_1}, \mathbf{x_2}\}, \mathbf{x_4}) = \min\{D_{sl}(\mathbf{x_1}, \mathbf{x_4}), D_{sl}(\mathbf{x_2}, \mathbf{x_4})\} = 3.35$
- $D_{sl}(\{\mathbf{x_1}, \mathbf{x_2}\}, \mathbf{x_5}) = \min\{D_{sl}(\mathbf{x_1}, \mathbf{x_5}), D_{sl}(\mathbf{x_2}, \mathbf{x_5})\} = 3$

$$\mathbf{D_{sl}} = \begin{array}{c|cccc} & \{\mathbf{x_1}, \mathbf{x_2}\} & \mathbf{x_3} & \mathbf{x_4} & \mathbf{x_5} \\ \hline \{\mathbf{x_1}, \mathbf{x_2}\} & 0 & 2.24 & 3.35 & 3 \\ \mathbf{x_3} & 2.24 & 0 & 1.12 & 1.41 \\ \mathbf{x_4} & 3.35 & 1.12 & 0 & 1.5 \\ \mathbf{x_5} & 3 & 1.41 & 1.5 & 0 \end{array}$$

AHC: Example (using single link method)

Merge $\mathbf{x_3}$ and $\mathbf{x_4}$

$$dend(h) = \begin{cases} \{\mathbf{x_1}\}, \{\mathbf{x_2}\}, \{\mathbf{x_3}\}, \{\mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x_1}, \mathbf{x_2}\}, \{\mathbf{x_3}\}, \{\mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 0.5 \leq h < 1.12 \\ \{\mathbf{x_1}, \mathbf{x_2}\}, \{\mathbf{x_3}, \mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 1.12 \leq h \end{cases}$$

AHC: Example (using single link method)

With the single link method, the distance matrix $\mathbf{D_{sl}}$ becomes :

- $D_{sl}(\{\mathbf{x_3},\mathbf{x_4}\},\{\mathbf{x_1},\mathbf{x_2}\}) = \min\{D_{sl}(\mathbf{x_3},\mathbf{x_1}), D_{sl}(\mathbf{x_3},\mathbf{x_2}), D_{sl}(\mathbf{x_4},\mathbf{x_1}), D_{sl}(\mathbf{x_4},\mathbf{x_2})\} = \min\{D_{sl}(\mathbf{x_3},\{\mathbf{x_1},\mathbf{x_2}\}), D_{sl}(\mathbf{x_4},\{\mathbf{x_1},\mathbf{x_2}\})\} = 2.24$
- $D_{sl}(\{\mathbf{x_3},\mathbf{x_4}\},\mathbf{x_5}) = \min\{D_{sl}(\mathbf{x_3},\mathbf{x_5}), D_{sl}(\mathbf{x_4},\mathbf{x_5})\} = 1.41$

$$\mathbf{D_{sl}} = \begin{array}{c|ccc} & \{\mathbf{x_1},\mathbf{x_2}\} & \{\mathbf{x_3},\mathbf{x_4}\} & \mathbf{x_5} \\ \hline \{\mathbf{x_1},\mathbf{x_2}\} & 0 & 2.24 & 3 \\ \{\mathbf{x_3},\mathbf{x_4}\} & 2.24 & 0 & 1.41 \\ \mathbf{x_5} & 3 & 1.41 & 0 \end{array}$$

AHC: Example (using single link method)

Merge $\{\mathbf{x_3}, \mathbf{x_4}\}$ and $\mathbf{x_5}$

$$
dend(h) = \begin{cases}
\{\mathbf{x_1}\}, \{\mathbf{x_2}\}, \{\mathbf{x_3}\}, \{\mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 0 \leq h < 0.5 \\
\{\mathbf{x_1}, \mathbf{x_2}\}, \{\mathbf{x_3}\}, \{\mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 0.5 \leq h < 1.12 \\
\{\mathbf{x_1}, \mathbf{x_2}\}, \{\mathbf{x_3}, \mathbf{x_4}\}, \{\mathbf{x_5}\} & \text{if } 1.12 \leq h < 1.41 \\
\{\mathbf{x_1}, \mathbf{x_2}\}, \{\mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5}\} & \text{if } 1.41 \leq h
\end{cases}
$$

AHC: Example (using single link method)

With the single link method, the distance matrix $\mathbf{D_{sl}}$ becomes :

- $D_{sl}(\{\mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5}\}, \{\mathbf{x_1}, \mathbf{x_2}\}) = \min\{D_{sl}(\{\mathbf{x_3}, \mathbf{x_4}\}, \{\mathbf{x_1}, \mathbf{x_2}\}), D_{sl}(\mathbf{x_5}, \{\mathbf{x_1}, \mathbf{x_2}\})\} = 2.24$

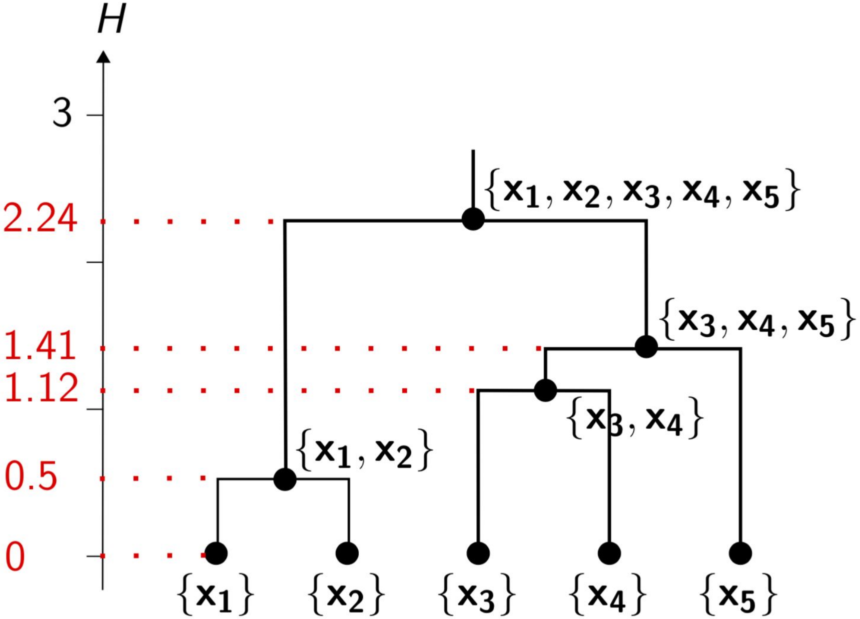
$$\mathbf{D_{sl}} = \begin{array}{c|cc} & \{\mathbf{x_1}, \mathbf{x_2}\} & \{\mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5}\} \\ \hline \{\mathbf{x_1}, \mathbf{x_2}\} & 0 & 2.24 \\ \{\mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5}\} & 2.24 & 0 \end{array}$$

AHC: Example (using single link method)

Merge $\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$ and $\{\mathbf{x}_1, \mathbf{x}_2\}$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \le h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \le h < 1.12 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 1.12 \le h < 1.41 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 1.41 \le h < 2.24 \\ \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} & \text{if } 2.24 \le h \end{cases}$$

AHC: Example (using single link method)



Source: Lecture slides of Julien Ah Pine, Université de Lyon 2, France, 2013

Divisive Hierarchical Clustering

Steps:

1. Start with the whole dataset as one cluster
2. Recursively divide the cluster into two sub-clusters
3. Repeat Step 2 until each cluster has only one object or some other stopping criterion has been met.

Step 2 needs to resolve the following two issues:

- Which cluster to split?
- How to split a cluster?

Which cluster to split?

There are a number of possibilities when selecting the next cluster to split:

- Split the cluster in some sequential order
- Split the cluster that has the largest number of objects
- Split the cluster that has the largest variation within it

How to split a cluster?

One simple approach for splitting a cluster is to split the cluster based on distances between the objects in the cluster.

1. Create a distance matrix by computing distances between all pairs of objects within the cluster.
2. Find the two most dissimilar objects (i.e., the objects that have the largest distance between them) and use them as seeds of a K-means method to create two new clusters.

Clustering: Determining the number of clusters

- Some clustering algorithms like k-means require the number of clusters at first.
- Determining the number of clusters is far from easy, often because the “right” number is ambiguous.
- There are many possible ways to estimate the number of clusters.
  a. Set the number of clusters to about √(n/2) for a data set of n points. In expectation, each cluster has √(2n) points.
  b. The elbow method
     - Run the clustering algorithm for k clusters, and calculate the sum of within-cluster variances, var(k).
     - Plot the curve of var with respect to k. The first (or most significant) turning point of the curve suggests the “right” number.
  c. Cross-validation

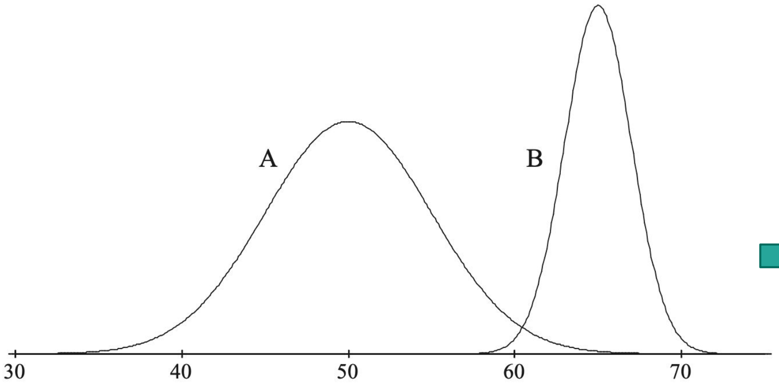
Probabilistic model-based methods

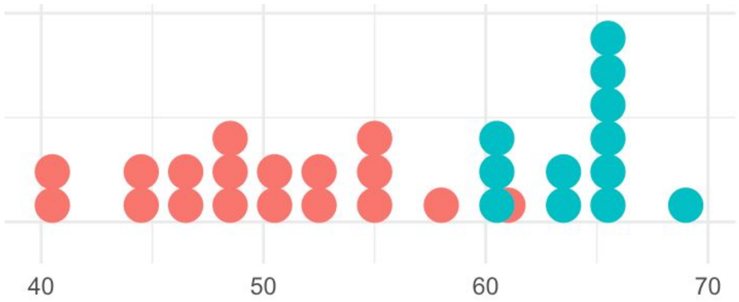
In this method, each object is assigned a **probability of belonging to a cluster**.

For example, while clustering of news articles, such models might result that the given news article has 40% chance of being in the cluster of science, 50% in history, and 10% in world news.

Probabilistic model-based methods

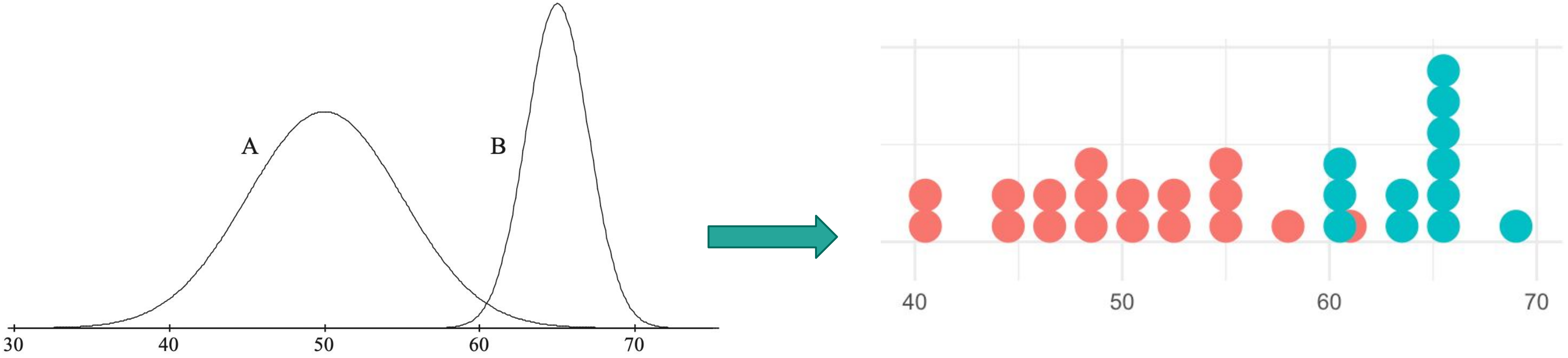
In probabilistic model-based clustering, the data is considered as coming from a mixture of k probability distributions, representing k clusters, that govern the attribute values for members of that cluster.





Probabilistic model-based methods

If we had data belonging to two known classes A and B, each having a normal distribution with means and standard deviations μ_A and σ_A for class A, and μ_B and σ_B for class B, we could define a model whereby samples are taken from these distributions, using cluster A with probability p_A and cluster B with probability p_B (where $p_A + p_B = 1$),

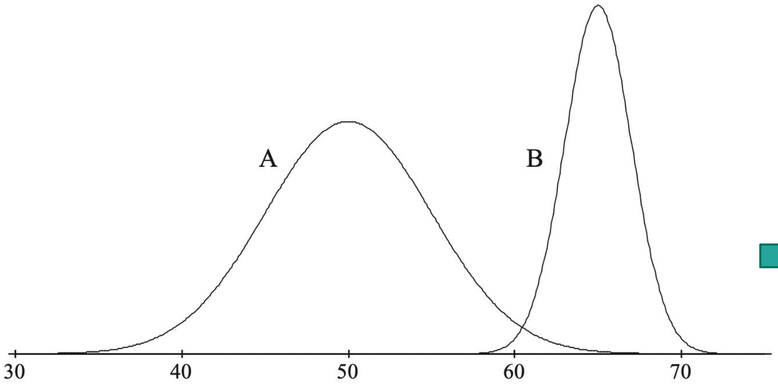


μ_A=50, σ_A=5, p_A=0.6 μ_A=65, σ_A=2, p_B= 0.4

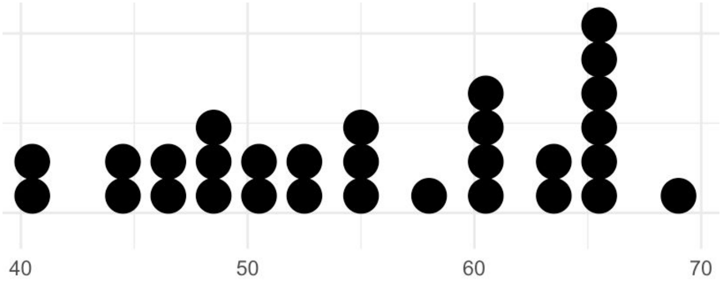
Probabilistic model-based methods

Now, imagine given the dataset without the classes—just the numbers—and being asked to determine the five parameters that characterize the model: μ_A, σ_A, μ_B, σ_B, and p_A (the parameter p_B can be calculated directly from p_A).

That is the **finite-mixture problem**.



μ_A=?, σ_A=?, p_A=? μ_A=?, σ_A=?, p_B=?



The Expectation Maximization Algorithm

Steps:

1. Start with initial guesses for the parameters, use them to calculate the cluster probabilities for each instance (“expectation”),
2. Use these probabilities to reestimate the parameters (“maximization” of the likelihood of the distributions given the data available), and
3. repeat.

The EM algorithm for Gaussian Mixture Model

1. Start with a random (but reasonable) assignment to the parameters
2. Compute the posterior distribution for the cluster assignments for each example

   $$P(\mathrm{A}|x_i) = \frac{P(x_i\,|\,\mathrm{A})\cdot P(\mathrm{A})}{P(x_i)} = \frac{N(x_i;\mu_{\mathrm{A}},\sigma_{\mathrm{A}})\,p_{\mathrm{A}}}{N(x_i;\mu_{\mathrm{A}},\sigma_{\mathrm{A}})\,p_{\mathrm{A}} + N(x_i;\mu_{\mathrm{B}},\sigma_{\mathrm{B}})\,p_{\mathrm{B}}}$$

   Where N() is the normal or Gaussian distribution, i.e., $N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

3. Update the parameters based on the expected class assignments - where probabilities act like weights.
   - If w_i is the probability that instance i belongs to cluster A, the mean and std. dev. are

   $$\mu_{\mathrm{A}} = \frac{w_1x_1 + w_2x_2 + \ldots + w_nx_n}{w_1 + w_2 + \ldots + w_n}$$

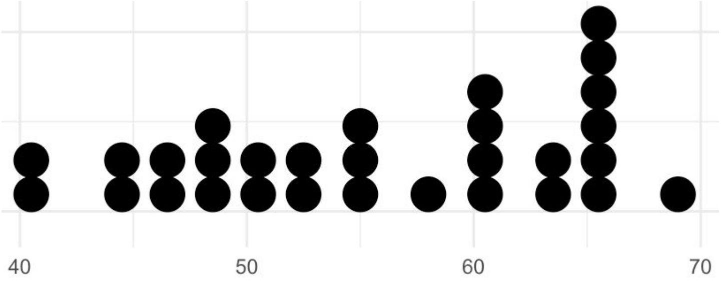
   $$\sigma_{\mathrm{A}}^2 = \frac{w_1(x_1-\mu)^2 + w_2(x_2-\mu)^2 + \ldots + w_n(x_n-\mu)^2}{w_1 + w_2 + \ldots + w_n}$$

Exercise

Given the following data

58 50 55 55 52 47 53 49 45 46 61 51 44 48 41 40 49 55 69 61 63 66 65 64 65 60 66 60 65 65

Estimate a model for a two class Gaussian Mixture Model (GMM).

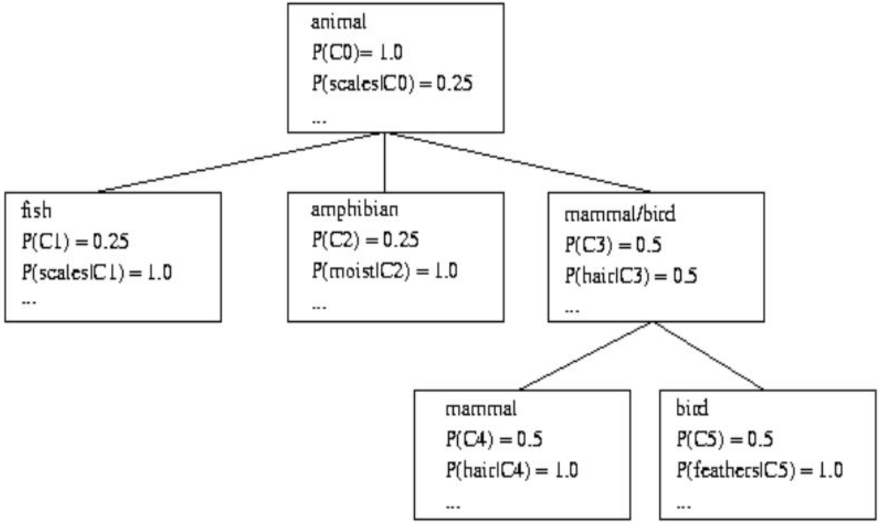


Incremental Clustering

- Batch clustering algorithms, such as k-means, process all data points at essentially the same time.
- In contrast, incremental clustering algorithms accept a stream of data, and process the data one element at a time.
- Important when the system needs to deal with very large, continuously growing datasets

COBWEB: An incremental hierarchical conceptual clustering

- COBWEB recursively sorts instances into a hierarchy of increasingly specific clusters.
- Each node refers to a concept and contains a probabilistic description of that concept.

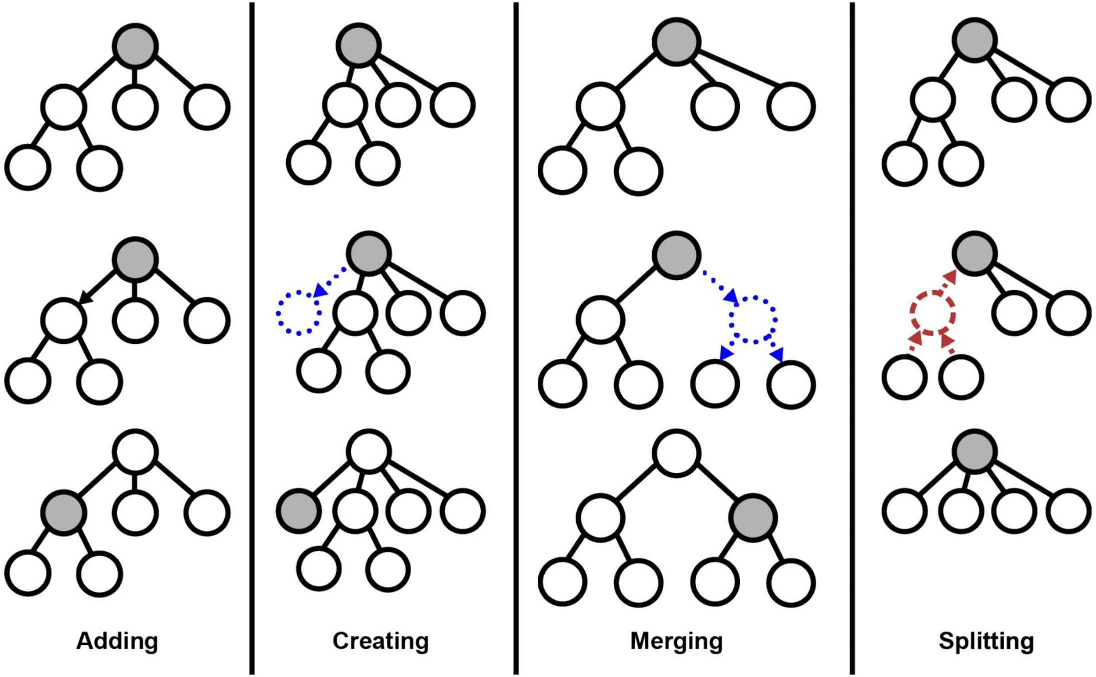


COBWEB: An incremental hierarchical conceptual clustering

- Each incoming object is incorporated into the hierarchy in a top down manner by classifying the object by descending the tree along an appropriate path. At each level, one of the four operations is performed:
    - Classifying the object with respect to an existing class,
    - Creating a new class,
    - Combining two classes into a single class (*merging*), and
    - Dividing a class into several classes (*splitting*).
- Which operation is selected depends on the **category utility** of the classification achieved by applying it.

COBWEB

The four operations



COBWEB

- Category Utility measures the overall quality of a partition of instances into clusters.
- It is a tradeoff between intra-class similarity and inter-class dissimilarity of objects.
- The larger the intra-class similarity, $P(A_i = V_{ij} \mid C_k)$, the greater the proportion of class members sharing the value and the more **predictable** the value is of class members.
- The larger the inter-class similarity, $P(C_k \mid A_i = V_{ij})$, the fewer the objects in contrasting classes that share this value and the more **predictive** the value is of the class.

COBWEB

Category Utility of a set of children/clusters $\{C_1, C_2, \cdots, C_n\}$ is

$$CU(\{C_1, C_2, \cdots, C_n\}) = \frac{\sum_{k=1}^{n} P(C_k)[\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2]}{n}$$

Where

$P(C_k)$ is the probability of a particular concept given its parent,

$P(A_i = V_{ij} |C_k)$ is the probability of attribute A_i having value V_{ij} in the child concept C_k,

$P(A_i = V_{ij})$ is the probability of attribute A_i having value V_{ij} in the parent concept, and

n is the number of child concepts.

COBWEB

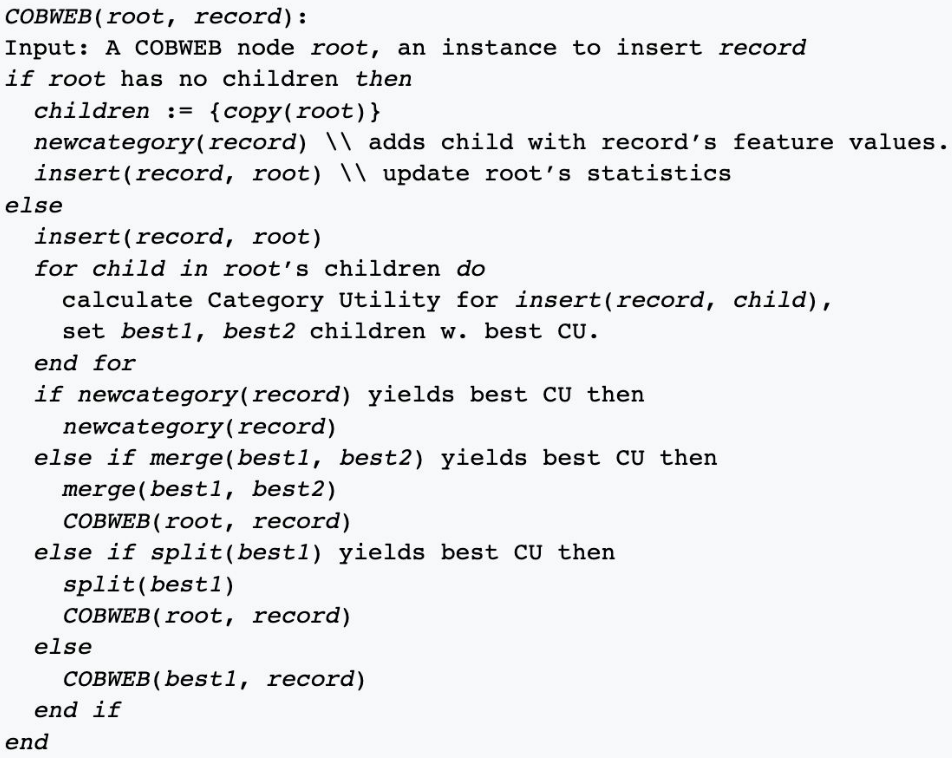
Category Utility of a set of children/clusters $\{C_1, C_2, \cdots, C_n\}$ is

$$CU(\{C_1, C_2, \cdots, C_n\}) \quad = \quad \frac{\sum_{k=1}^{n} P(C_k)[\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2]}{n}$$

This measure rewards clusters, C_k, that increase the predictability of variable values within C_k relative to their predictability in the population as a whole.

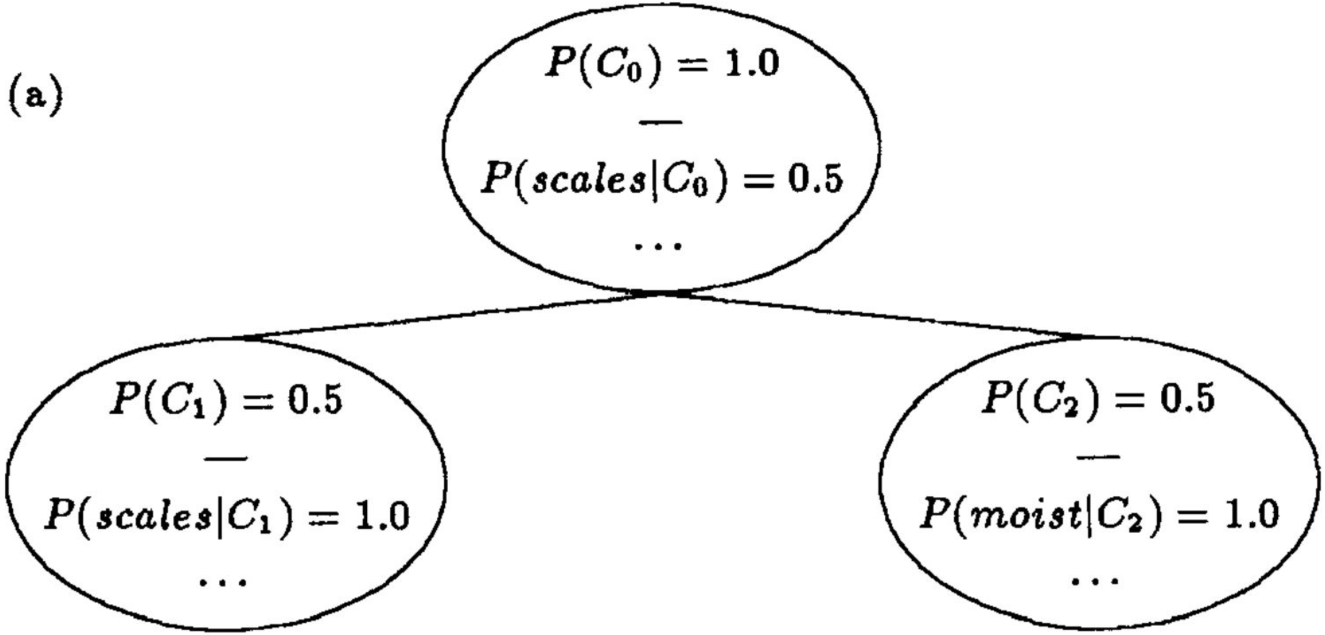
A cluster is well-separated or decoupled from other clusters if many variable values are predictive of the cluster.

COBWEB



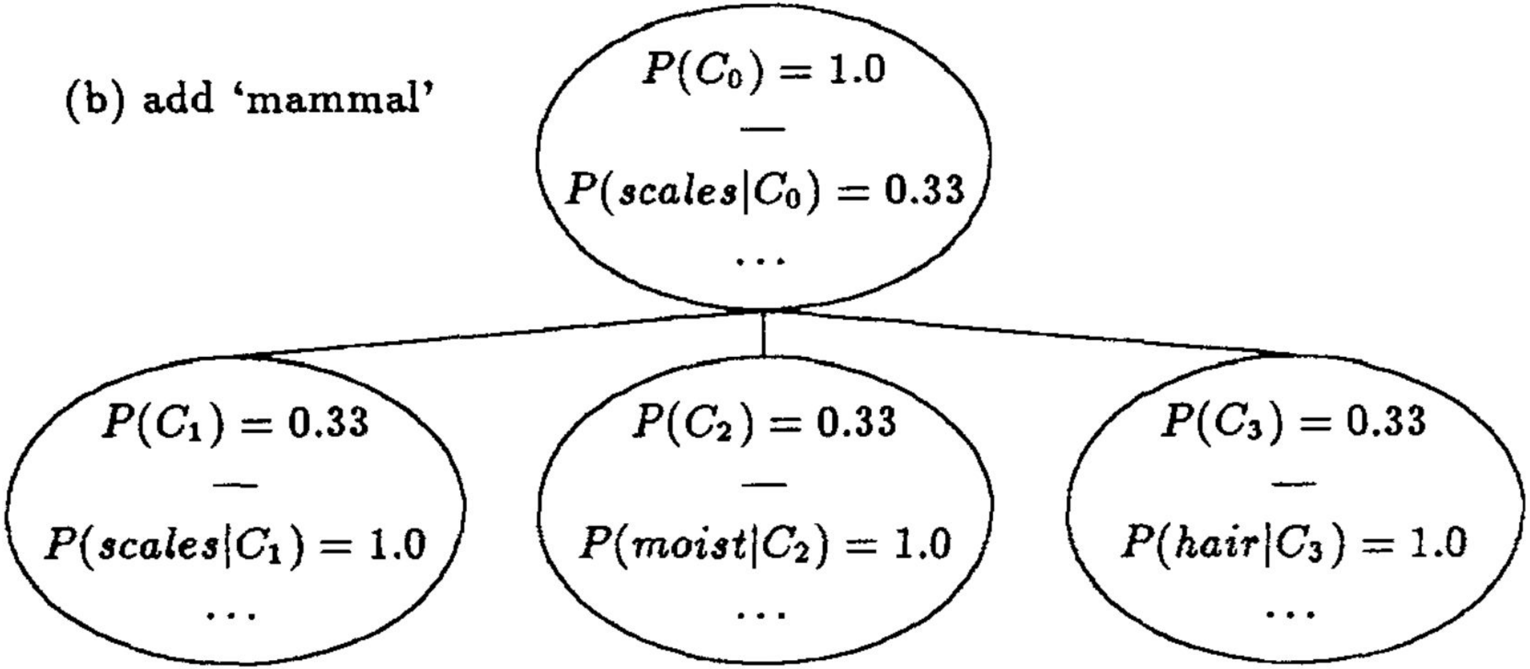
```
COBWEB(root, record):
Input: A COBWEB node root, an instance to insert record
if root has no children then
  children := {copy(root)}
  newcategory(record) \\ adds child with record's feature values.
  insert(record, root) \\ update root's statistics
else
  insert(record, root)
  for child in root's children do
    calculate Category Utility for insert(record, child),
    set best1, best2 children w. best CU.
  end for
  if newcategory(record) yields best CU then
    newcategory(record)
  else if merge(best1, best2) yields best CU then
    merge(best1, best2)
    COBWEB(root, record)
  else if split(best1) yields best CU then
    split(best1)
    COBWEB(root, record)
  else
    COBWEB(best1, record)
  end if
end
```

COBWEB: Example



COBWEB: Example

(b) add ‘mammal’



$P(C_0) = 1.0$

—

$P(scales|C_0) = 0.33$

$\ldots$

$P(C_1) = 0.33$

—

$P(scales|C_1) = 1.0$

$\ldots$

$P(C_2) = 0.33$

—

$P(moist|C_2) = 1.0$

$\ldots$

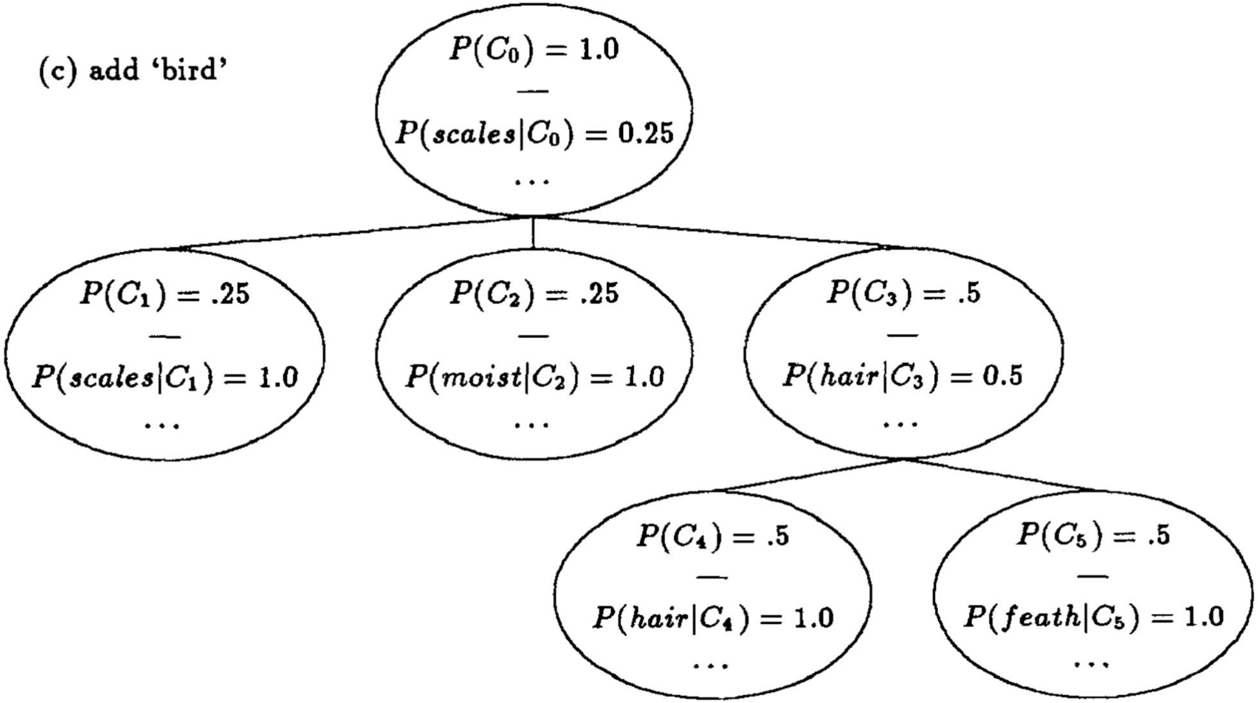
$P(C_3) = 0.33$

—

$P(hair|C_3) = 1.0$

$\ldots$

COBWEB: Example



COBWEB Implementation

https://github.com/cmaclell/concept_formation

Evaluation of Clustering

Evaluation of Clustering

Cluster evaluation assesses the feasibility of clustering analysis on a data set and the quality of the results generated by a clustering method.

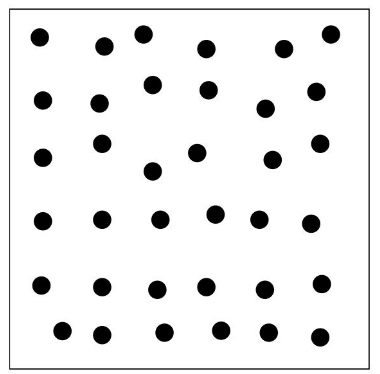
Major tasks:

- Assessing clustering tendency
- Determining the number of clusters in a data set
- Measuring clustering quality

Assessing Clustering Tendency

Clustering tendency assessment determines whether a given data set has a non-random structure, which may lead to meaningful clusters.

Applying clustering algorithms to a dataset without a non-random structure, such as a set of uniformly distributed points in a data space, is not useful.



A uniformly distributed dataset

Assessing Clustering Tendency

Hopkins-Skellam test of Complete Spatial Randomness can be used to measure the probability that the data points are generated by a uniform data distribution.

Null Hypothesis (homogeneous hypothesis):
Data points are uniformly distributed and thus contains no meaningful clusters

Alternative Hypothesis (nonhomogeneous hypothesis):
Data points are not uniformly distributed (presence of clusters)

We compute the Hopkins Statistic, H.

Calculating the Hopkins Statistic, H

1. Sample n points, p_1, … , p_n, uniformly from D. For each point, p_i, we find the nearest neighbor of p_i $(1 \leq i \leq n)$ in D, and let x_i be the distance between p_i and its nearest neighbor in D.

$$x_i = \min_{v \in D}\{dist(p_i, v)\}$$

2. Sample n points, q_1, … , q_n, uniformly from D. For each q_i $(1 \leq i \leq n)$, we find the nearest neighbor of q_i in D — {q_i}, and let y_i be the distance between q_i and its nearest neighbor in D—{q_i}.

$$y_i = \min_{v \in D, v \neq q_i}\{dist(q_i, v)\}$$

Calculating the Hopkins Statistic, H

3. Calculate the Hopkins Statistic, H, as

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

A value close to 1 tends to indicate the data is highly clustered, random data will tend to result in values around 0.5, and uniformly distributed data will tend to result in values close to 0.

Determining the Number of Clusters

Domain knowledge can be used to determine the number of clusters.

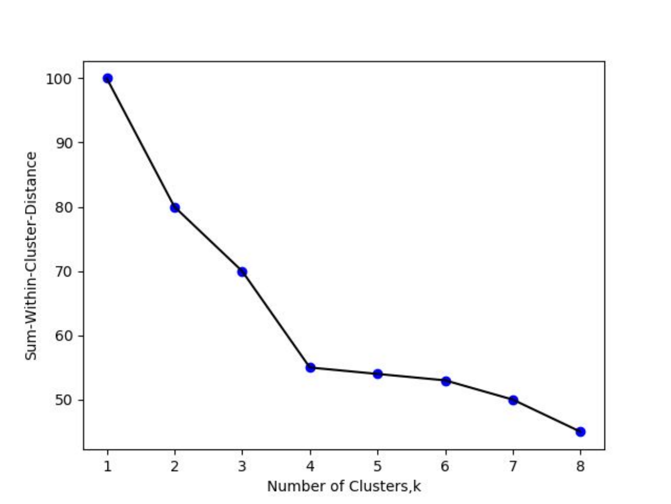
If domain knowledge is not available, mathematical methods can be applied.

- Set the number of clusters to about $\sqrt{(n/2)}$ for a data set of n points. In expectation, each cluster has $\sqrt{2n}$ points.
- The elbow method

The elbow method

- Run the clustering algorithm for k clusters, and calculate the sum of within-cluster variances, var(k).

- Plot the curve of var with respect to k. The first (or most significant) turning point of the curve suggests the "right" number.



Source: https://towardsdatascience.com/clustering-evaluation-strategies-98a4006fcfc

Measuring Clustering Quality

- **Extrinsic methods** compare the clustering against the ground truth and measure.

  Ground truth is the ideal clustering that is often built using human experts.

  BCubed precision and recall

- **Intrinsic methods** evaluate the goodness of a clustering by considering how well the clusters are separated.

  Silhouette coefficient

Silhouette Coefficient

For a data set, D, of n objects, suppose D is partitioned into k clusters, C1,...,Ck.

The Silhouette Coefficient of an object o ∈ D is defined as

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

where, a(o) is the average distance between o and all other objects in the cluster to which o belongs,

$$a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o, o')}{|C_i| - 1}$$

b(o) is the minimum average distance from o to all clusters to which o does not belong.

$$b(o) = \min_{C_j : 1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|} \right\}$$

Silhouette Coefficient

- The smaller the value of a(o) is, the more compact the cluster.
- The larger the value of b(o) is, the more separated o is from other clusters.
- When s(o) approaches 1, the cluster containing o is compact and o is far away from other clusters.
- When s(o) is negative, o is closer to the objects in another cluster than to the objects in the same cluster as o.
- To measure the quality of a clustering, we can use the average Silhouette Coefficient value of all objects in the data set.
- The Silhouette Coefficient can also be used in the elbow method in place of the sum of within-cluster variances.