Clustering

Dit-Yan Yeung

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

COMP 4211: Machine Learning (Fall 2022)

Introduction

- Clustering is the process of grouping similar objects together.
- Categorization according to input type:
  - Feature-based clustering:
    The input to the clustering algorithm is an $N \times d$ feature matrix where each row is the feature vector for one example.
  - Similarity-based clustering:
    The input to the clustering algorithm is an $N \times N$ distance matrix where each non-diagonal entry corresponds to the pairwise distance between two examples.
- Categorization according to output type:
  - Partitional clustering:
    The output is a partition consisting of disjoint sets of examples.
  - Hierarchical clustering:
    The output is a nested tree of partitions.

Partitional Clustering vs. Hierarchical Clustering

- Partitional clustering algorithms are usually more efficient with $\mathcal{O}(Nd)$ complexity (as opposed to $\mathcal{O}(N^2 \log N)$ for hierarchical clustering algorithms).
- Hierarchical clustering algorithms do not require the number of clusters to be specified.
- Hierarchical clustering algorithms can work directly on a distance matrix without requiring that the examples be provided as feature vectors.

k-Means Clustering Algorithm

- Problem formulation:
  - Given a set $\mathcal{S} = \left\{\mathbf{x}^{(\ell)}\right\}_{\ell=1}^{N}$ of N unlabeled examples.
  - Find k cluster means (a.k.a. reference vectors, prototypes, codebook vectors, or codewords) $\mathbf{m}_i, i = 1, \ldots, k$ which best represent the set.
- We first consider the k-means clustering algorithm for a fixed value of k, and then consider how to choose a good value of k.
- Ideally, we would like the distance between each example and the cluster mean nearest to it to be as small as possible.
- Sum of squared errors (SSE):

$$E(\{\mathbf{m}_i\}; \mathcal{S}) = \sum_{\ell} \sum_{i} b_i^{(\ell)} \|\mathbf{x}^{(\ell)} - \mathbf{m}_i\|^2,$$

where

$$b_i^{(\ell)} = \begin{cases} 1 & \text{if } i = \arg\min_j \|\mathbf{x}^{(\ell)} - \mathbf{m}_j\| \\ 0 & \text{otherwise.} \end{cases}$$

k-Means Clustering as an Optimization Problem

- Given $\mathcal{S}$, $E(\{\mathbf{m}_i\}; \mathcal{S})$ is a function of $\{\mathbf{m}_i\}$. So the k-means clustering algorithm is an optimization problem which tries to find the cluster means that minimize the SSE.
- If the indicator variables $\{b_i^{(\ell)}\}$ did not depend on $\{\mathbf{m}_i\}$, the optimal solution could be found in closed form because $E(\{\mathbf{m}_i\}; \mathcal{S})$ would be quadratic with respect to each cluster mean $\mathbf{m}_i$. However, $\{b_i^{(\ell)}\}$ actually depend on $\{\mathbf{m}_i\}$.
- The k-means clustering algorithm is an iterative algorithm for solving the optimization problem by iterating between two steps:
  - Updating $\{b_i^{(\ell)}\}$ with $\{\mathbf{m}_i\}$ fixed
  - Updating $\{\mathbf{m}_i\}$ with $\{b_i^{(\ell)}\}$ fixed.

Algorithm

Initialize $\mathbf{m}_i, i = 1, \ldots, k$ (e.g., k randomly selected $\mathbf{x}^{(\ell)}$)
repeat
  for each $\mathbf{x}^{(\ell)} \in \mathcal{S}$ **do**
    # assign example to nearest cluster

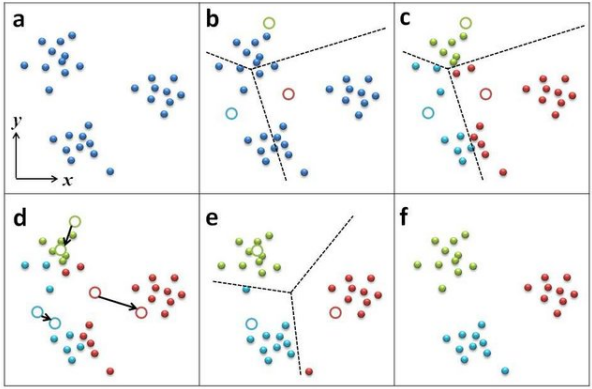
$$b_i^{(\ell)} = \begin{cases} 1 & \text{if } i = \arg\min_j \left\| \mathbf{x}^{(\ell)} - \mathbf{m}_j \right\| \\ 0 & \text{otherwise} \end{cases}$$

  end for
  for each $\mathbf{m}_i, i = 1, \ldots, k$ **do**
    # update cluster mean based on examples assigned to cluster

$$\mathbf{m}_i = \frac{\sum_\ell b_i^{(\ell)} \mathbf{x}^{(\ell)}}{\sum_\ell b_i^{(\ell)}}$$

  end for
until cluster means stop changing.

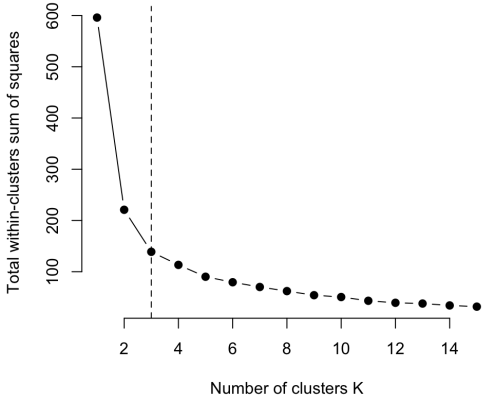
Illustrative Example



Using the Elbow Method to Determine k

- The idea of the elbow method is to run the k-means clustering algorithm on the set of examples for a range of values of k (say, k from 1 to 10).
- For each value of k, calculate the SSE after the k-means clustering algorithm has converged.
- Plot a line chart of the SSE for each value of k. If the line chart looks like an arm, then the "elbow" on the arm is the best value of k.
- The elbow usually represents where we start to have diminishing returns if k further increases.

Elbow in SSE Curve



Pros and Cons of k-Means

- Pros:
  - Relatively simple to implement.
  - Scales well to large datasets even with multiple runs.
  - Guarantees convergence.
  - Easy to interpret the results.
- Cons:
  - Number of clusters determined manually.
  - Sensitive to initialization.
  - Sensitive to outliers.
  - Difficult for data where clusters are of varying size and density.
  - Not good for high-dimensional data.

Variations of k-Means

- The k-medians clustering algorithm is a variant of the k-means clustering algorithm.
- Instead of using the mean to determine the centroid for each cluster, the median is used instead.
- The median is computed for each feature dimension separately.
- Advantages of using medians:
  - Less sensitive to outliers (e.g., the mean of $2, 4, 6, 8, 80$ is 20 while the median is 6).
  - More reasonable for discrete feature values.
- Another variant is the k-medoids clustering algorithm. Unlike a median, a medoid must be an actual example in the set.

Two Hierarchical Clustering Approaches

- Agglomerative (or bottom-up) approach:
  Start with the examples as singleton clusters and, at each step, merge the closest pair of clusters.
- Divisive (or top-down) approach:
  Start with one cluster including all the examples and, at each step, split a cluster until only singleton clusters remain.

Agglomerative Hierarchical Clustering Algorithm (Sketch)

Compute an $N \times N$ distance matrix (if not yet available)

repeat

Merge the closest pair of clusters

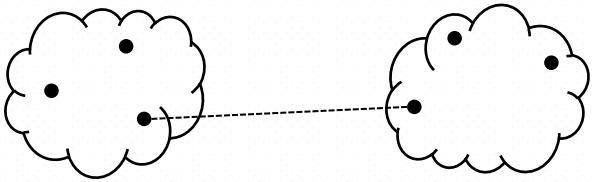
Update the distance matrix to reflect the proximity between the new cluster and the original clusters

until only one cluster remains.

Defining Proximity between Clusters

- The proximity between clusters can be defined using a distance function $D(\mathcal{C}_i, \mathcal{C}_j)$ between clusters $\mathcal{C}_i$ and $\mathcal{C}_j$.
- Some common proximity measures:
  - Single-link proximity
  - Complete-link proximity
  - Average-link proximity

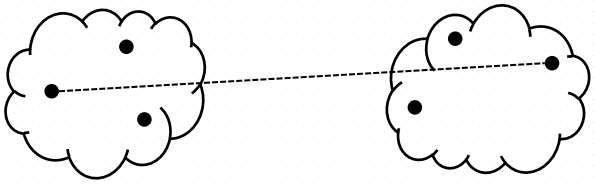
Single-Link Proximity



- Single-link proximity defines the distance between two clusters as the shortest distance between two points that are in different clusters:

$$D_s(\mathcal{C}_i, \mathcal{C}_j) = \min_{\mathbf{x}^{(\ell)} \in \mathcal{C}_i,\, \mathbf{x}^{(\ell')} \in \mathcal{C}_j} d(\mathbf{x}^{(\ell)}, \mathbf{x}^{(\ell')}).$$

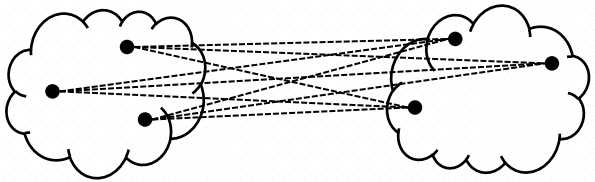
Complete-Link Proximity



- Complete-link proximity defines the distance between two clusters as the longest distance between two points that are in different clusters:

$$D_c(\mathcal{C}_i, \mathcal{C}_j) = \max_{\mathbf{x}^{(\ell)} \in \mathcal{C}_i,\, \mathbf{x}^{(\ell')} \in \mathcal{C}_j} d(\mathbf{x}^{(\ell)}, \mathbf{x}^{(\ell')}).$$

Average-Link Proximity



- Average-link proximity defines the distance between two clusters as the average distance between two points that are in different clusters:

$$D_a(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{|\mathcal{C}_i| \cdot |\mathcal{C}_j|} \sum_{\mathbf{x}^{(\ell)} \in \mathcal{C}_i,\, \mathbf{x}^{(\ell')} \in \mathcal{C}_j} d(\mathbf{x}^{(\ell)}, \mathbf{x}^{(\ell')}),$$

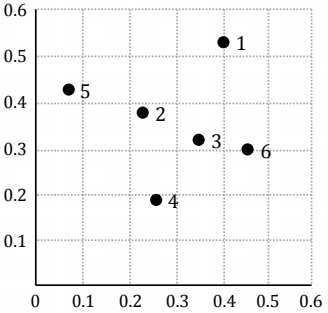
where $|\mathcal{C}|$ denotes the size of cluster $\mathcal{C}$.

Properties of Different Proximity Measures

- Single-link proximity tends to form elongated, chain-like clusters, but it can be sensitive to outliers.
- Complete-link proximity usually breaks large clusters into smaller ones and favors globular shapes.
- Average-link proximity is an intermediate approach between the single-link and complete-link approaches.

From Feature Matrix to Distance Matrix

Set of 6 Two-Dimensional Points



xy Coordinates of 6 Points

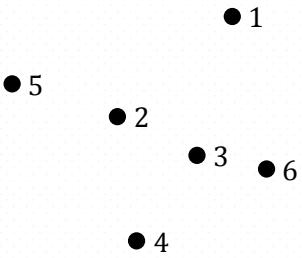
Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Euclidean Distance Matrix for 6 Points

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Single-Link Clustering Example

Nested Cluster Diagram

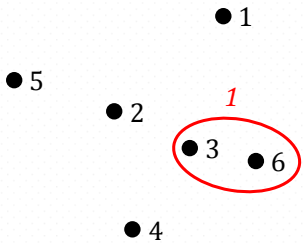


Single-link Distance Matrix

	1	2	3	4	5	6
1	0	0.24	0.22	0.37	0.34	0.23
2		0	0.15	0.20	0.14	0.25
3			0	0.15	0.28	0.11
4				0	0.29	0.22
5					0	0.39
6						0

Single-Link Clustering Example (2)

Nested Cluster Diagram

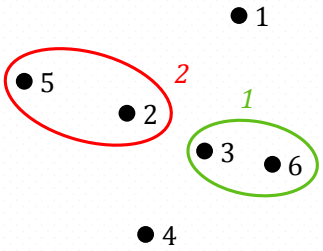


Single-link Distance Matrix

	1	2	3	4	5	6
1	0	0.24	0.22	0.37	0.34	0.23
2		0	0.15	0.20	0.14	0.25
3			0	0.15	0.28	0.11
4				0	0.29	0.22
5					0	0.39
6						0

Single-Link Clustering Example (3)

Nested Cluster Diagram

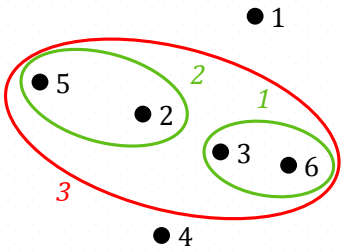


Single-link Distance Matrix

	1	2	4	5	3,6
1	0	0.24	0.37	0.34	0.22
2		0	0.20	0.14	0.15
4			0	0.29	0.15
5				0	0.28
3,6					0

Single-Link Clustering Example (4)

Nested Cluster Diagram

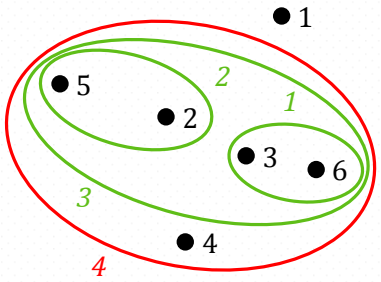


Single-link Distance Matrix

	1	4	2,5	3,6
1	0	0.37	0.24	0.22
4		0	0.20	0.15
2,5			0	0.15
3,6				0

Single-Link Clustering Example (5)

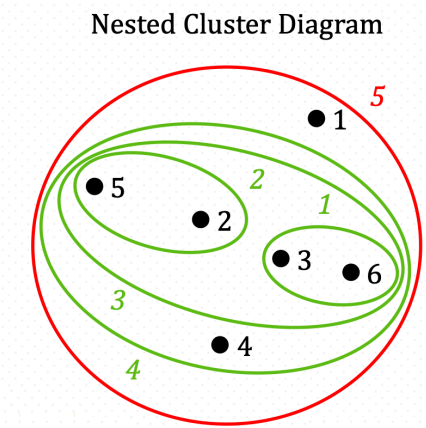
Nested Cluster Diagram



Single-link Distance Matrix

	1	4	2,5,3,6
1	0	0.37	0.22
4		0	0.15
2,5,3,6			0

Single-Link Clustering Example (6)

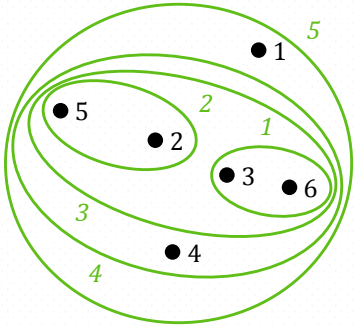


Single-link Distance Matrix

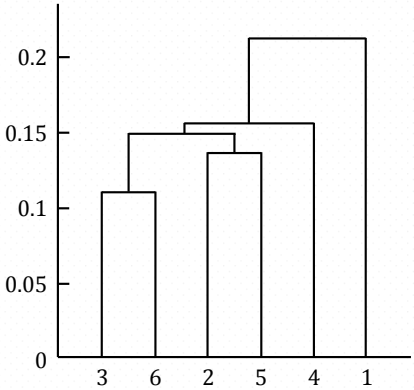
	1	4,2,5,3,6
1	0	0.22
4,2,5,3,6		0

Single-Link Clustering Example (7)

Nested Cluster Diagram



Hierarchical Tree Diagram



To Learn More...

- Spectral clustering