THE UNIVERSITY *of* EDINBURGH
informatics

Applied Machine Learning (AML)

Clustering

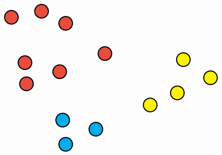
Oisin Mac Aodha • Siddharth N.

Outline

- What is clustering and why is it useful?
- What kinds are there and how are they characterised?
- Explore
  - K-Means
  - Hierarchical Clustering
- How do we evaluate clustering?

Clustering

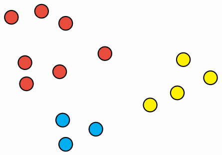
- Discover the underlying structure of data



Clusters in 2D

Clustering

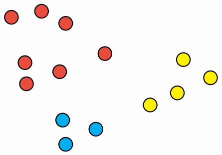
- Discover the underlying structure of data
- What sub-groups exist in the data
  - # clusters, size, …
  - common properties within sub-group
  - potential for further clustering



Clusters in 2D

Clustering

- Discover the underlying structure of data
- What sub-groups exist in the data
  - # clusters, size, …
  - common properties within sub-group
  - potential for further clustering



Clusters in 2D

Applications

- discover classes / structure in an unsupervised manner
  - clustering images of handwritten digits (K=10)
  - finding phylogenetic trees using DNA
- dimensionality reduction: clusters ↔ “latent factors”
  - use cluster id as representation
  - assume relevant characteristics reflected in cluster membership

Features of Clustering Algorithms

Hard vs. Soft

Hard: objects belong to a single cluster

Soft: objects have soft assignments—distribution over clusters

Features of Clustering Algorithms

Hard vs. Soft

Hard: objects belong to a single cluster

Soft: objects have soft assignments—distribution over clusters

Flat vs. Hierarchical

Flat: single group of clusters

Hierarchical: clusters at different levels

Features of Clustering Algorithms

Hard vs. Soft

Hard: objects belong to a single cluster

Soft: objects have soft assignments—distribution over clusters

Flat vs. Hierarchical

Flat: single group of clusters

Hierarchical: clusters at different levels

Monothetic vs. Polythetic

Monothetic: clustered based on common feature (e.g. hair colour)

Polythetic: clustered based on distance measure(s) over features

Features of Clustering Algorithms

Hard vs. Soft

Hard: objects belong to a single cluster

Soft: objects have soft assignments—distribution over clusters

Flat vs. Hierarchical

Flat: single group of clusters

Hierarchical: clusters at different levels

Monothetic vs. Polythetic

Monothetic: clustered based on common feature (e.g. hair colour)

Polythetic: clustered based on distance measure(s) over features

K-Means

K-Means

Characteristics

Hard: a point belongs to just one cluster

Flat: single level of clustering

Polythetic: distance-based similarity within clusters

K-Means

Characteristics

Hard: a point belongs to just one cluster

Flat: single level of clustering

Polythetic: distance-based similarity within clusters

Idea

Ensure points closest to some special point end up in the same cluster

- Top-down approach
- Produces a partition of the data
- Requires defining a distance metric over points

K-Means Algorithm

Require: $\mathcal{D}, K, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ # clusters, points

$\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\} \leftarrow$ random initialisation ▹ centroids of clusters

repeat

  for $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ **do**

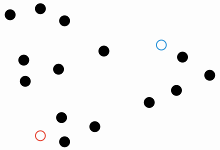
    $\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$ ▹ find nearest centroid id

    $\boldsymbol{c}_k^* \leftarrow \boldsymbol{x}_n$ ▹ assign point to cluster

  for $\boldsymbol{c}_k \in \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ **do**

    $\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \boldsymbol{x}_n$ ▹ update cluster centroids

until cluster assignments do not change



K-Means Algorithm

Require: $\mathcal{D}, K, \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ ▹ # clusters, points

$\{\boldsymbol{c}_1, \dots, \boldsymbol{c}_K\} \leftarrow$ random initialisation ▹ centroids of clusters

repeat

  for $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ **do**

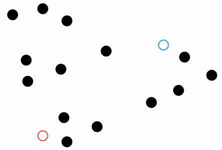
    $\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$ ▹ find nearest centroid id

    $\boldsymbol{c}_k^* \leftarrow \boldsymbol{x}_n$ ▹ assign point to cluster

  for $\boldsymbol{c}_k \in \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_K\}$ **do**

    $\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \boldsymbol{x}_n$ ▹ update cluster centroids

until cluster assignments do not change



K-Means Algorithm

Require: $\mathcal{D}, K, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▷ # clusters, points

$\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\} \leftarrow$ random initialisation ▷ centroids of clusters

repeat

  for $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ **do**

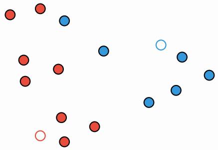
    $\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$ ▷ find nearest centroid id

    $\boldsymbol{c}_k^* \leftarrow \boldsymbol{x}_n$ ▷ assign point to cluster

  for $\boldsymbol{c}_k \in \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ **do**

    $\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \rightarrow \boldsymbol{c}_k} \boldsymbol{x}_n$ ▷ update cluster centroids

until cluster assignments do not change



K-Means Algorithm

Require: $\mathcal{D}, K, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ # clusters, points

$\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\} \leftarrow$ random initialisation ▹ centroids of clusters

repeat

  for $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ **do**

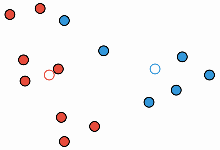
    $\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$ ▹ find nearest centroid id

    $\boldsymbol{c}_k^* \leftarrow \boldsymbol{x}_n$ ▹ assign point to cluster

  for $\boldsymbol{c}_k \in \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ **do**

    $\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \boldsymbol{x}_n$ ▹ update cluster centroids

until cluster assignments do not change



K-Means Algorithm

Require: $\mathcal{D}, K, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ # clusters, points

$\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\} \leftarrow$ random initialisation ▹ centroids of clusters

repeat

  for $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ **do**

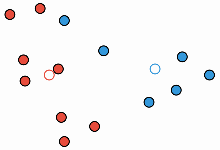
    $\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$ ▹ find nearest centroid id

    $\boldsymbol{c}_k^* \leftarrow \boldsymbol{x}_n$ ▹ assign point to cluster

  for $\boldsymbol{c}_k \in \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ **do**

    $\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \boldsymbol{x}_n$ ▹ update cluster centroids

until cluster assignments do not change



K-Means Algorithm

Require: $\mathcal{D}, K, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ # clusters, points

$\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\} \leftarrow$ random initialisation ▹ centroids of clusters

repeat

  for $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ **do**

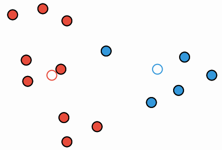
    $\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$ ▹ find nearest centroid id

    $\boldsymbol{c}_k^* \leftarrow \boldsymbol{x}_n$ ▹ assign point to cluster

  for $\boldsymbol{c}_k \in \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ **do**

    $\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \boldsymbol{x}_n$ ▹ update cluster centroids

until cluster assignments do not change



K-Means Algorithm

Require: $\mathcal{D}, K, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ # clusters, points

$\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\} \leftarrow$ random initialisation ▹ centroids of clusters

repeat

  for $\boldsymbol{x}_n \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ **do**

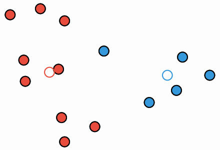
    $\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$ ▹ find nearest centroid id

    $\boldsymbol{c}_k^* \leftarrow \boldsymbol{x}_n$ ▹ assign point to cluster

  for $\boldsymbol{c}_k \in \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ **do**

    $\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \boldsymbol{x}_n$ ▹ update cluster centroids

until cluster assignments do not change

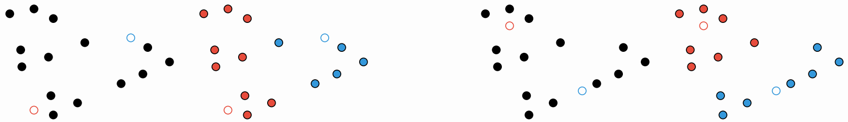


K-Means Properties

- Minimises aggregate intra-cluster distance: $V = \sum_{k} \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$
  - if $\mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k) = \|\boldsymbol{x}_n - \boldsymbol{c}_k\|_2^2$, i.e., Euclidean distance, then V is proportional to variance

K-Means Properties

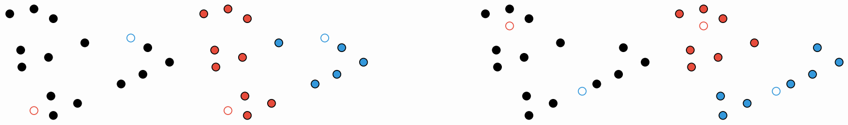
- Minimises aggregate intra-cluster distance: $V = \sum_k \sum_{\boldsymbol{x}_n \to \boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$
  - if $\mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k) = \|\boldsymbol{x}_n - \boldsymbol{c}_k\|_2^2$, i.e., Euclidean distance, then V is proportional to variance
- Converges to *local* minimum
  - *different* initialisations lead to *different* clustering results
  - repeat several random initialisations and pick one with smallest aggregate distance



K-Means Properties

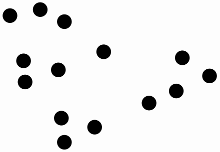
- Minimises aggregate intra-cluster distance: $V = \sum_{k} \sum_{\boldsymbol{x}_n \rightarrow \boldsymbol{c}_k} \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k)$
  - if $\mathcal{D}(\boldsymbol{x}_n, \boldsymbol{c}_k) = \|\boldsymbol{x}_n - \boldsymbol{c}_k\|_2^2$, i.e., Euclidean distance, then V is proportional to variance

- Converges to *local* minimum
  - *different* initialisations lead to *different* clustering results
  - repeat several random initialisations and pick one with smallest aggregate distance



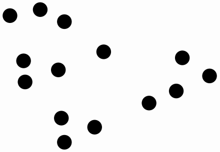
- 'Adjacent' points can end up in different clusters

Estimating Number of Clusters

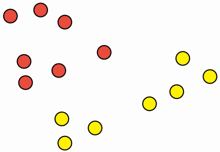


data

Estimating Number of Clusters

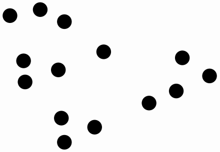


data

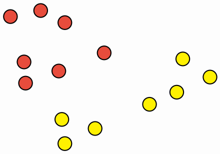


2 clusters

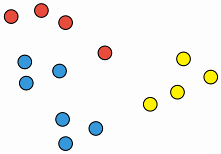
Estimating Number of Clusters



data

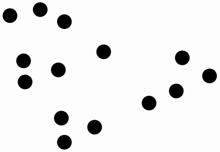


2 clusters

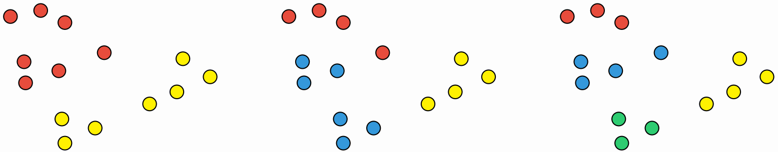


3 clusters

Estimating Number of Clusters

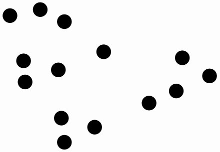


data

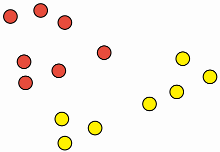


2 clusters 3 clusters 4 clusters

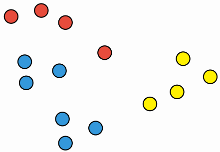
Estimating Number of Clusters



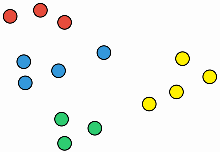
data



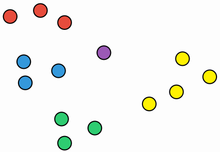
2 clusters



3 clusters



4 clusters



5 clusters

Estimating Number of Clusters

How many clusters does your data have?

- Get (K) from class labels (e.g. digits 0…9)

Estimating Number of Clusters

How many clusters does your data have?

- Get (K) from class labels (e.g. digits 0…9)
- Find an “appropriate” K: optimise for V

Estimating Number of Clusters

How many clusters does your data have?

- Get (K) from class labels (e.g. digits 0…9)
- Find an "appropriate" K: optimise for V
  - Run K-Means for $K = 1, 2, \ldots$; choose K with smallest V

Estimating Number of Clusters

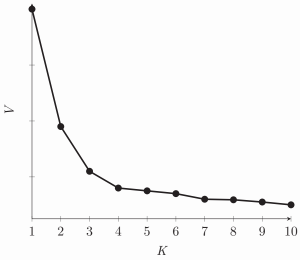
How many clusters does your data have?

- Get (K) from class labels (e.g. digits 0…9)
- Find an “appropriate” K: optimise for V
  - Run K-Means for $K = 1, 2, \ldots$; choose K with smallest V
  - **Issue:** What is V when $K = N$?
    – choose best K on *validation* data

Estimating Number of Clusters

How many clusters does your data have?

- Get (K) from class labels (e.g. digits 0…9)
- Find an "appropriate" K: optimise for V
  - Run K-Means for $K = 1, 2, \ldots$; choose K with smallest V
  - **Issue:** What is V when $K = N$?
    – choose best K on *validation* data
  - Choose visually from a ***elbow*** plot
    – point that maximises the 2nd derivative of V



K-Means: Example

Colour Quantisation

- Original Image: 96,615 colours

Original

Figures: Scikit Learn: Colour Quantization using K-Means

K-Means: Example

Colour Quantisation

- Original Image: 96,615 colours
- Quantised Image: 64 colours (K-Means)
    - Replace pixel value $\boldsymbol{x}_i$ with cluster centroid $\boldsymbol{c}_k$ value

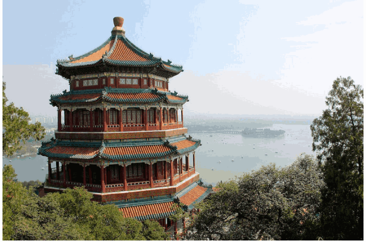
$$\boldsymbol{x}_i \in \mathbb{R}^3 \qquad \text{(pixel values in RGB)}$$

$$\mathcal{D}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$$

$$K = 64$$

Original

K-Means Quantised



Figures: Scikit Learn: Colour Quantization using K-Means

K-Means: Example

Colour Quantisation

- Original Image: 96,615 colours
- Quantised Image: 64 colours (K-Means)
  - Replace pixel value $\boldsymbol{x}_i$ with cluster centroid $\boldsymbol{c}_k$ value
- Quantised Image: 64 colours (Random)
  - Select random set of K pixels as "centroids"
  - Replace pixel value $\boldsymbol{x}_i$ with nearest "centroid" value

$$\boldsymbol{x}_i \in \mathbb{R}^3 \qquad \text{(pixel values in RGB)}$$

$$\mathcal{D}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$$

$$K = 64$$

Original

Random Quantised

Figures: Scikit Learn: Colour Quantization using K-Means

K-Means: Example

Clustering Handwritten Digits

- High-dimensional data

$$\boldsymbol{x} \in \mathbb{R}^{784}$$

K-Means: Example

Clustering Handwritten Digits

- High-dimensional data
- Dimensionality reduction (e.g. PCA)

$$\boldsymbol{x} \in \mathbb{R}^{784}$$

$$\boldsymbol{e} \in \mathbb{R}^{2} \qquad \text{(PCA)}$$

K-Means: Example

Clustering Handwritten Digits

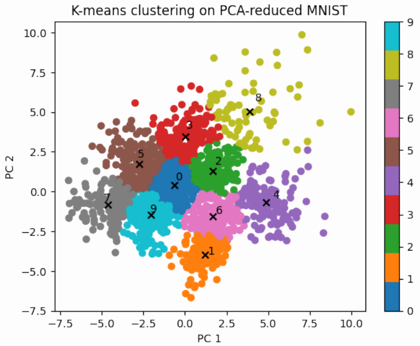
- High-dimensional data
- Dimensionality reduction (e.g. PCA)
- K-Means on embeddings

$$\boldsymbol{x} \in \mathbb{R}^{784}$$

$$\boldsymbol{e} \in \mathbb{R}^{2} \qquad \text{(PCA)}$$

$$\mathcal{D}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2^2$$

$$K = 10$$



Hierarchical Clustering

Hierarchical Clustering

Choosing number of clusters

- Depends a lot on *granularity*

Hierarchical Clustering

Choosing number of clusters

- Depends a lot on *granularity*
  - data (e.g. satellite maps—how much does 1 pixel cover?)

Hierarchical Clustering

Choosing number of clusters

- Depends a lot on *granularity*
    - data (e.g. satellite maps—how much does 1 pixel cover?)
    - context—what do we care about? High vs. low level?

Hierarchical Clustering

Choosing number of clusters

- Depends a lot on *granularity*
  - data (e.g. satellite maps—how much does 1 pixel cover?)
  - context—what do we care about? High vs. low level?
- No magical algorithm to give you *correct* K

Hierarchical Clustering

Choosing number of clusters

- Depends a lot on *granularity*
  - data (e.g. satellite maps—how much does 1 pixel cover?)
  - context—what do we care about? High vs. low level?
- No magical algorithm to give you *correct* K

Hierarchical Clustering

Choosing number of clusters

- Depends a lot on *granularity*
  - data (e.g. satellite maps—how much does 1 pixel cover?)
  - context—what do we care about? High vs. low level?
- No magical algorithm to give you *correct* K

Find a hierarchy of structure

- **Upper levels:** coarse groups (e.g. collection of objects; bedroom, kitchen, etc.)
- **Lower levels:** fine-grained (e.g. object parts; chair leg, table top, etc.)
- Stategies
  - Top-Down: start with everything in one cluster, then split recursively
  - Bottom-up: start with each item separately, then merge recursively

Hierarchical K-Means

- Top-Down approach

Hierarchical K-Means

- Top-Down approach
  - perform K-Means on data

Hierarchical K-Means

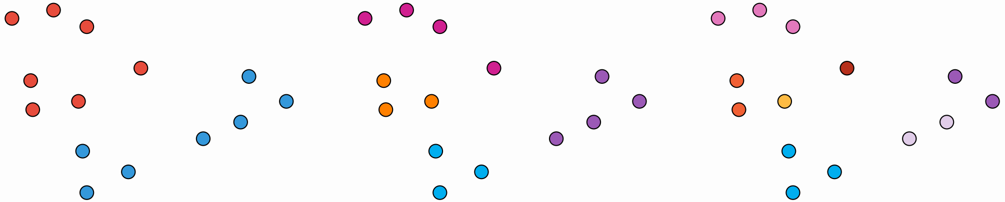
- Top-Down approach
  - perform K-Means on data
  - for each resulting cluster c_i, run K-Means within c_i

Hierarchical K-Means

- Top-Down approach
  - perform K-Means on data
  - for each resulting cluster c_i, run K-Means within c_i
- **Fast:** recursive calls on successively smaller datasets

Hierarchical K-Means

- Top-Down approach
  - perform K-Means on data
  - for each resulting cluster c_i, run K-Means within c_i
- **Fast:** recursive calls on successively smaller datasets
- **Greedy:** once cluster has been determined at top level; cannot change



Agglomerative Clustering

Characteristics

Hard: a point belongs to just one cluster

Hierarchical: multiple levels of clustering

Polythetic: distance-based similarity within clusters

Agglomerative Clustering

Characteristics

Hard: a point belongs to just one cluster

Hierarchical: multiple levels of clustering

Polythetic: distance-based similarity within clusters

Idea

Ensure “nearby” points end up in the same cluster

- Bottom-up approach
- Generates a dendrogram: hierarchical tree of clusters
- Requires defining a distance metric over *clusters*

Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

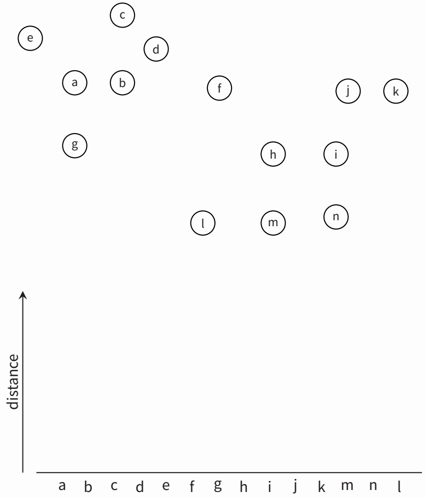
$\quad \boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

$\quad \boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

$\quad C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

$\quad C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining



Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▷ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▷ initial clusters

repeat

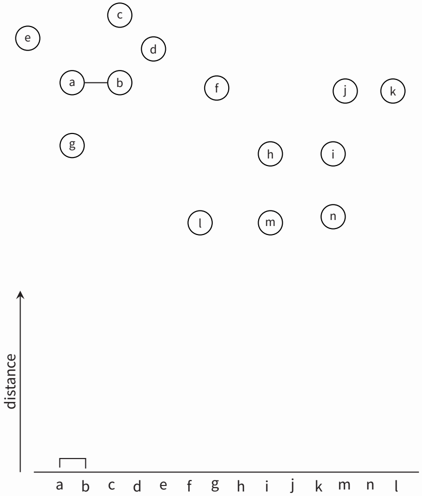
  $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \underset{\boldsymbol{c}_i, \boldsymbol{c}_j}{\arg\min}\, \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▷ find closest pair

  $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▷ merge into new cluster

  $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▷ remove pair of clusters

  $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▷ add merged cluster

until only one cluster remaining



Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \dots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

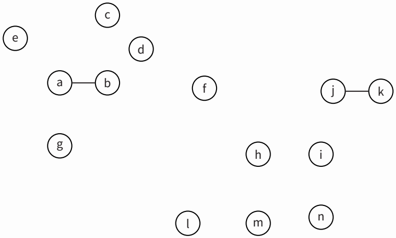
    $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

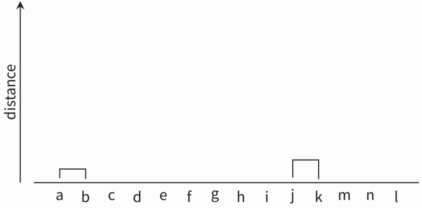
    $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

    $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

    $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

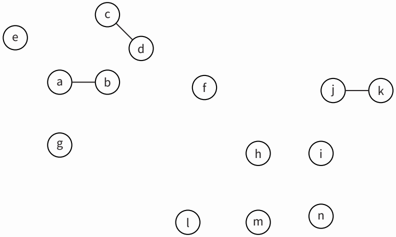
  $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

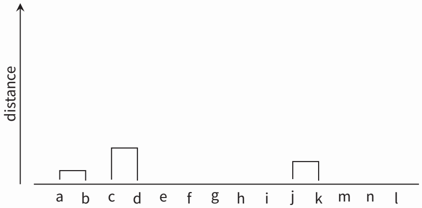
  $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

  $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

  $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

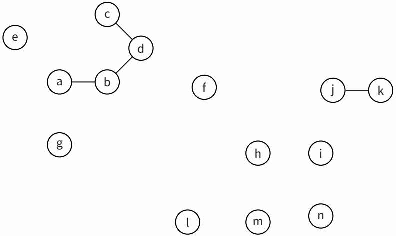
  $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

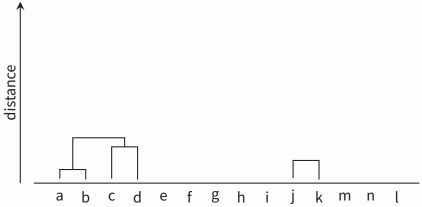
  $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

  $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

  $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining

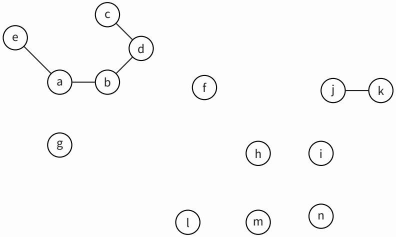




Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points



Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

1: $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

2: **repeat**

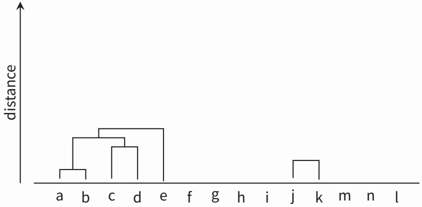
3: $\quad \boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

4: $\quad \boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

5: $\quad C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

6: $\quad C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

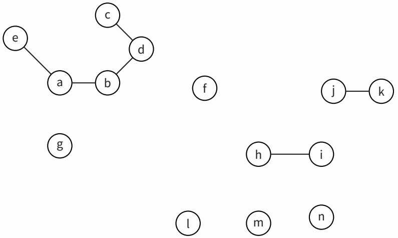
7: **until** only one cluster remaining

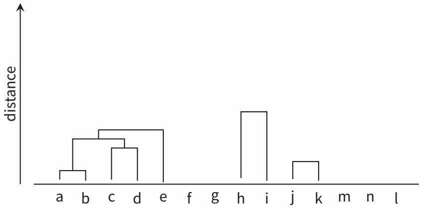


Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*
$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points
1: $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters
2: **repeat**
3: $\quad \boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair
4: $\quad \boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster
5: $\quad C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters
6: $\quad C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster
7: **until** only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

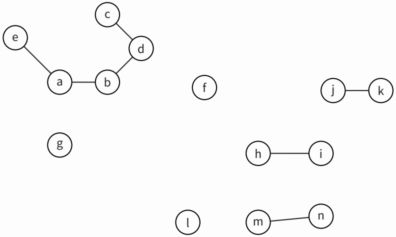
  $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

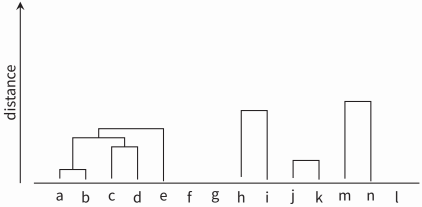
  $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

  $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

  $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

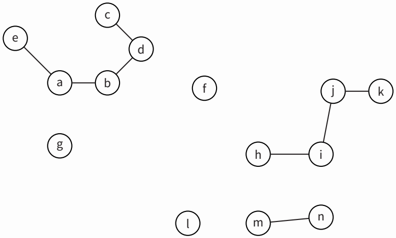
  $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

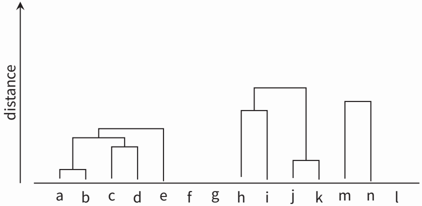
  $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

  $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

  $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

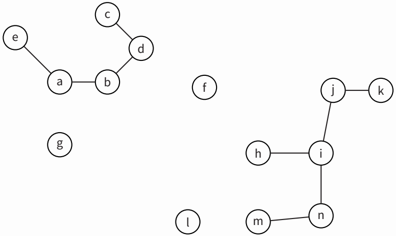
  $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

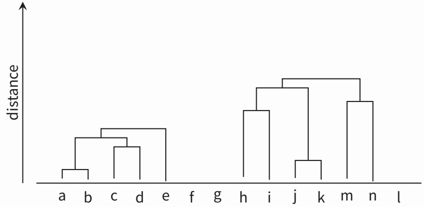
  $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

  $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

  $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

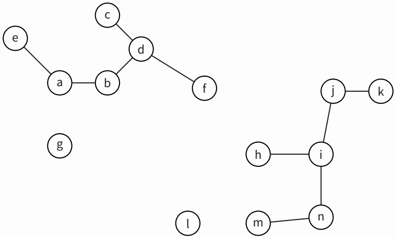
$\quad \boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \underset{\boldsymbol{c}_i, \boldsymbol{c}_j}{\arg\min}\, \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

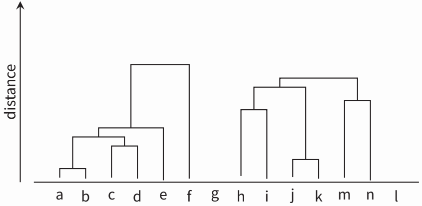
$\quad \boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

$\quad C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

$\quad C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

1: $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

2: **repeat**

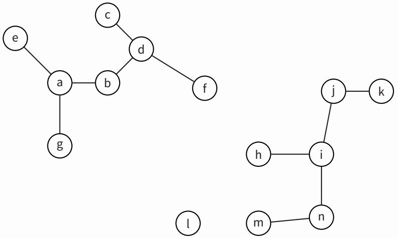
3: $\quad \boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

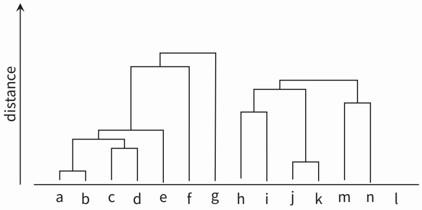
4: $\quad \boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

5: $\quad C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

6: $\quad C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

7: **until** only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

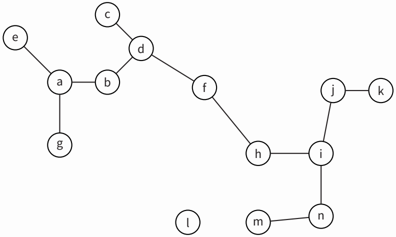
  $\boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

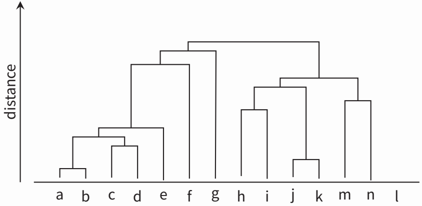
  $\boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

  $C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

  $C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

repeat

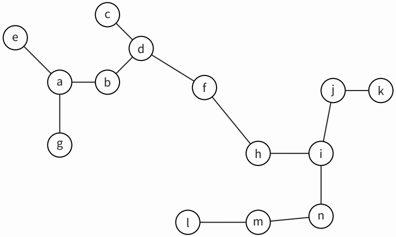
$\quad \boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

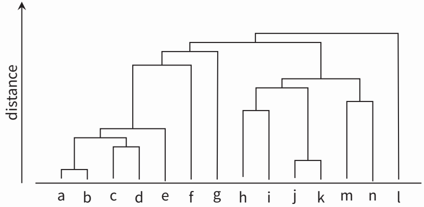
$\quad \boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

$\quad C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

$\quad C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

until only one cluster remaining





Agglomerative Clustering: Sketch

$\mathcal{D}(\boldsymbol{x}_l, \boldsymbol{x}_m)$ —distance between *points*

$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ —distance between *clusters* of points

Require: $\mathcal{G}_{\mathcal{D}}, \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ ▹ points

1: $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\} = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}\}$ ▹ initial clusters

2: **repeat**

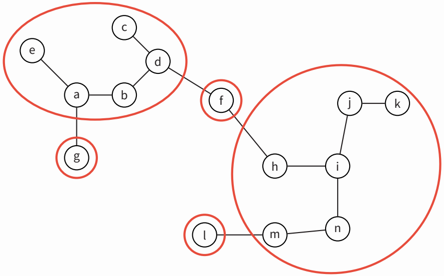
3: $\quad \boldsymbol{c}_i^*, \boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_i, \boldsymbol{c}_j} \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$ ▹ find closest pair

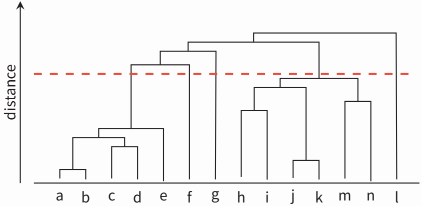
4: $\quad \boldsymbol{c}_{i \cdot j} \leftarrow \boldsymbol{c}_i^*, \boldsymbol{c}_j^*$ ▹ merge into new cluster

5: $\quad C = C \setminus \{\boldsymbol{c}_i^*, \boldsymbol{c}_j^*\}$ ▹ remove pair of clusters

6: $\quad C = C \cup \{\boldsymbol{c}_{i \cdot j}\}$ ▹ add merged cluster

7: **until** only one cluster remaining





Cluster Distance Measures

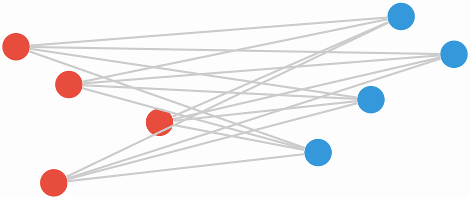
Single Link

$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \min_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$

Cluster Distance Measures

Single Link

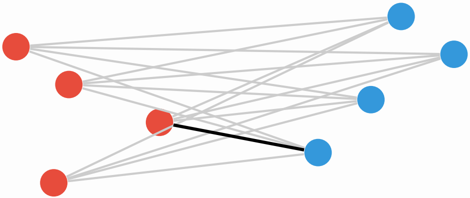
$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \min_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$



Cluster Distance Measures

Single Link

$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \min_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$



Cluster Distance Measures

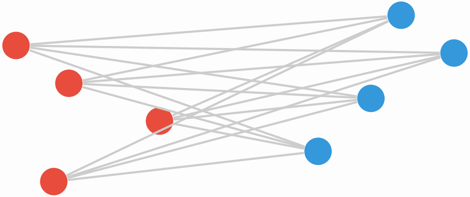
Complete Link

$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \max_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$

Cluster Distance Measures

Complete Link

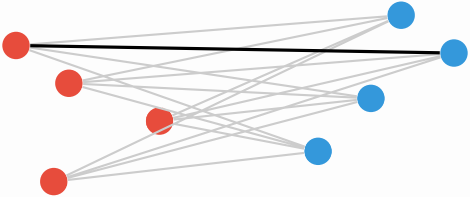
$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \max_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$



Cluster Distance Measures

Complete Link

$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \max_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$



Cluster Distance Measures

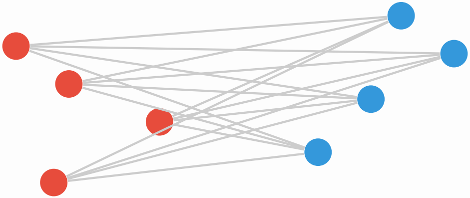
Average Link

$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \frac{1}{|\boldsymbol{c}_i|\,|\boldsymbol{c}_j|} \sum_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$

Cluster Distance Measures

Average Link

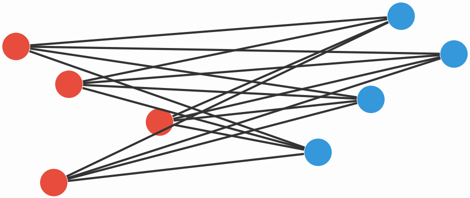
$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \frac{1}{|\boldsymbol{c}_i|\,|\boldsymbol{c}_j|} \sum_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$



Cluster Distance Measures

Average Link

$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \frac{1}{|\boldsymbol{c}_i|\,|\boldsymbol{c}_j|} \sum_{\substack{\boldsymbol{x}_{i,l} \in \boldsymbol{c}_i \\ \boldsymbol{x}_{j,m} \in \boldsymbol{c}_j}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$



Cluster Distance Measures

Ward's Method

$$\bar{\boldsymbol{x}}_{ij} = \frac{1}{|\boldsymbol{c}_{ij}|} \sum_{\boldsymbol{x}_l \in \boldsymbol{c}_{ij}} \boldsymbol{x}_l \qquad (\boldsymbol{c}_{ij} = \boldsymbol{c}_i \cup \boldsymbol{c}_j)$$

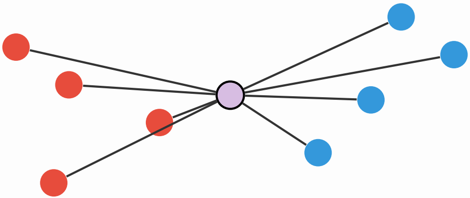
$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \frac{1}{|\boldsymbol{c}_{ij}|} \sum_{\boldsymbol{x}_l \in \boldsymbol{c}_{ij}} \mathcal{D}(\boldsymbol{x}_l, \bar{\boldsymbol{x}}_{ij}) = \frac{1}{|\boldsymbol{c}_{ij}|} \sum_{\boldsymbol{x}_l \in \boldsymbol{c}_{ij}} \|\boldsymbol{x}_l - \bar{\boldsymbol{x}}_{ij}\|^2$$

Cluster Distance Measures

Ward's Method

$$\bar{\boldsymbol{x}}_{ij} = \frac{1}{|\boldsymbol{c}_{ij}|} \sum_{\boldsymbol{x}_l \in \boldsymbol{c}_{ij}} \boldsymbol{x}_l \qquad (\boldsymbol{c}_{ij} = \boldsymbol{c}_i \cup \boldsymbol{c}_j)$$

$$\mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \frac{1}{|\boldsymbol{c}_{ij}|} \sum_{\boldsymbol{x}_l \in \boldsymbol{c}_{ij}} \mathcal{D}(\boldsymbol{x}_l, \bar{\boldsymbol{x}}_{ij}) = \frac{1}{|\boldsymbol{c}_{ij}|} \sum_{\boldsymbol{x}_l \in \boldsymbol{c}_{ij}} \|\boldsymbol{x}_l - \bar{\boldsymbol{x}}_{ij}\|^2$$



Unified Formulation

Lance-Williams Algorithm

- When merging two clusters to get $\boldsymbol{c}_{i \cdot j}$
- Need to compute updated distances to all other clusters

Unified Formulation

Lance-Williams Algorithm

- When merging two clusters to get $\boldsymbol{c}_{i \cdot j}$
- Need to compute updated distances to all other clusters

For each remaining cluster $\boldsymbol{c}_k$, denoting $G_{i,j} = \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$

$$G_{k,i \cdot j} = \alpha_i G_{k,i} + \alpha_j G_{k,j} + \beta G_{i,j} + \gamma |G_{k,i} - G_{k,j}|$$

Unified Formulation

Lance-Williams Algorithm

- When merging two clusters to get $\boldsymbol{c}_{i \cdot j}$
- Need to compute updated distances to all other clusters

For each remaining cluster $\boldsymbol{c}_k$, denoting $G_{i,j} = \mathcal{G}_{\mathcal{D}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$

$$G_{k,i \cdot j} = \alpha_i G_{k,i} + \alpha_j G_{k,j} + \beta G_{i,j} + \gamma |G_{k,i} - G_{k,j}|$$

Method	α_i	α_j	β	γ
Single Link	0.5	0.5	0	-0.5
Complete Link	0.5	0.5	0	0.5
Average Link	$\frac{\|\boldsymbol{c}_i\|}{\|\boldsymbol{c}_i\|+\|\boldsymbol{c}_j\|}$	$\frac{\|\boldsymbol{c}_j\|}{\|\boldsymbol{c}_i\|+\|\boldsymbol{c}_j\|}$	0	0
Ward's Method	$\frac{\|\boldsymbol{c}_i\|+\|\boldsymbol{c}_k\|}{\|\boldsymbol{c}_i\|+\|\boldsymbol{c}_j\|+\|\boldsymbol{c}_k\|}$	$\frac{\|\boldsymbol{c}_j\|+\|\boldsymbol{c}_k\|}{\|\boldsymbol{c}_i\|+\|\boldsymbol{c}_j\|+\|\boldsymbol{c}_k\|}$	$\frac{-\|\boldsymbol{c}_k\|}{\|\boldsymbol{c}_i\|+\|\boldsymbol{c}_j\|+\|\boldsymbol{c}_k\|}$	0

Evaluation

Evaluation

Extrinsic

Helps solve downstream task

- **Quantisation:** represent data with cluster features
  - colour quantisation—use centroid value
  - feature extraction—use cluster index

Evaluation

Extrinsic

Helps solve downstream task

- **Quantisation:** represent data with cluster features
  - colour quantisation—use centroid value
  - feature extraction—use cluster index
- **Partition:** treat clusters as different datasets
  - train separate classifiers for each sub-group
  - e.g. MNIST 1 vs. not 1; 2 vs. not 2 …

Evaluation

Extrinsic

Helps solve downstream task

- **Quantisation:** represent data with cluster features
    - colour quantisation—use centroid value
    - feature extraction—use cluster index
- **Partition:** treat clusters as different datasets
    - train separate classifiers for each sub-group
    - e.g. MNIST 1 vs. not 1; 2 vs. not 2 …
- **Key:** Does it help perform task better?

Evaluation

Intrinsic

Helps understand qualitative makeup of data

- **Unsupervised:** measure how well-separated clusters are
  - compare intra-cluster distances to inter-cluster distances
  - e.g. silhouette scores

Evaluation

Intrinsic

Helps understand qualitative makeup of data

- **Unsupervised:** measure how well-separated clusters are
  - compare intra-cluster distances to inter-cluster distances
  - e.g. silhouette scores
- **Supervised:** measure alignment of clusters to known labels
  - can treat as evaluation of classification
  - reason in terms of pairs belonging to cluster / label
  - **issue:** # cluster ≠ # labels

Evaluation

Intrinsic

Helps understand qualitative makeup of data

- **Unsupervised:** measure how well-separated clusters are
  - compare intra-cluster distances to inter-cluster distances
  - e.g. silhouette scores
- **Supervised:** measure alignment of clusters to known labels
  - can treat as evaluation of classification
  - reason in terms of pairs belonging to cluster / label
  - **issue:** # cluster $\neq$ # labels
- **Human:** compare judgements to humans on exemplars
  - ask human if pair $\boldsymbol{x}_i, \boldsymbol{x}_j$ belong together
  - compute match between human judgements and predictions: F1-score, κ, etc.

Intrinsic Evaluation: Unsupervised

In the absence of labels, or any other external measure of utility, can compute a generic measure of how well-clustered the data is.

Silhouette Score

Let data point $\boldsymbol{x}_l \in \boldsymbol{c}_i$ be denoted $\boldsymbol{x}_{i,l}$, then

Intrinsic Evaluation: Unsupervised

In the absence of labels, or any other external measure of utility, can compute a generic measure of how well-clustered the data is.

Silhouette Score

Let data point $\boldsymbol{x}_l \in \boldsymbol{c}_i$ be denoted $\boldsymbol{x}_{i,l}$, then

$$a_l = \frac{1}{|\boldsymbol{c}_i| - 1} \sum_{\substack{\boldsymbol{x}_{i,m} \in \boldsymbol{c}_i \\ m \neq l}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{i,m})$$

mean distance within cluster

Intrinsic Evaluation: Unsupervised

In the absence of labels, or any other external measure of utility, can compute a generic measure of how well-clustered the data is.

Silhouette Score

Let data point $\boldsymbol{x}_l \in \boldsymbol{c}_i$ be denoted $\boldsymbol{x}_{i,l}$, then

$$a_l = \frac{1}{|\boldsymbol{c}_i| - 1} \sum_{\substack{\boldsymbol{x}_{i,m} \in \boldsymbol{c}_i \\ m \neq l}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{i,m})$$

mean distance within cluster

$$b_l = \min_{j \neq i} \frac{1}{|\boldsymbol{c}_j|} \sum_{\boldsymbol{x}_{j,m} \in \boldsymbol{c}_j} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$

mean distance with *nearest* cluster

Intrinsic Evaluation: Unsupervised

In the absence of labels, or any other external measure of utility, can compute a generic measure of how well-clustered the data is.

Silhouette Score

Let data point $\boldsymbol{x}_l \in \boldsymbol{c}_i$ be denoted $\boldsymbol{x}_{i,l}$, then

$$a_l = \frac{1}{|\boldsymbol{c}_i| - 1} \sum_{\substack{\boldsymbol{x}_{i,m} \in \boldsymbol{c}_i \\ m \neq l}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{i,m})$$

mean distance within cluster

$$b_l = \min_{j \neq i} \frac{1}{|\boldsymbol{c}_j|} \sum_{\boldsymbol{x}_{j,m} \in \boldsymbol{c}_j} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$

mean distance with *nearest* cluster

$$s_l = \frac{b_l - a_l}{\max\{a_l, b_l\}} \quad |\boldsymbol{c}_i| > 1$$

Intrinsic Evaluation: Unsupervised

In the absence of labels, or any other external measure of utility, can compute a generic measure of how well-clustered the data is.

Silhouette Score

Let data point $\boldsymbol{x}_l \in \boldsymbol{c}_i$ be denoted $\boldsymbol{x}_{i,l}$, then

$$a_l = \frac{1}{|\boldsymbol{c}_i| - 1} \sum_{\substack{\boldsymbol{x}_{i,m} \in \boldsymbol{c}_i \\ m \neq l}} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{i,m})$$

mean distance within cluster

$$b_l = \min_{j \neq i} \frac{1}{|\boldsymbol{c}_j|} \sum_{\boldsymbol{x}_{j,m} \in \boldsymbol{c}_j} \mathcal{D}(\boldsymbol{x}_{i,l}, \boldsymbol{x}_{j,m})$$

mean distance with *nearest* cluster

$$s_l = \frac{b_l - a_l}{\max\{a_l, b_l\}} \qquad |\boldsymbol{c}_i| > 1$$

$$s = \frac{1}{N} \sum_{l=1}^{N} s_l \qquad -1 \leq s \leq 1$$

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$
  - still cannot compare directly—permutation unknown!

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$
  - still cannot compare directly—permutation unknown!
  - which u corresponds to which v?

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$
  - still cannot compare directly—permutation unknown!
  - which u corresponds to which v?
  - if $u \leftrightarrow v$ matching known

    standard measures: accuracy, F1-score, etc.

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$
  - still cannot compare directly—permutation unknown!
  - which u corresponds to which v?
  - if $u \leftrightarrow v$ matching known

    standard measures: accuracy, F1-score, etc.
- if $U \neq V$

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$
  - still cannot compare directly—permutation unknown!
  - which u corresponds to which v?
  - if $u \leftrightarrow v$ matching known

    standard measures: accuracy, F1-score, etc.
- if $U \neq V$
  - need to *also* find best alignment

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$
  - still cannot compare directly—permutation unknown!
  - which u corresponds to which v?
  - if $u \leftrightarrow v$ matching known

    standard measures: accuracy, F1-score, etc.
- if $U \neq V$
  - need to *also* find best alignment
  - can have multiple $c_u \longrightarrow$ *same* r_v

Intrinsic Evaluation: Supervised

Issue: Alignment

Clustering produces clusters $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_U\}$

Labels induce *reference* clusters $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_V\}$

- if $U = V$
  - still cannot compare directly—permutation unknown!
  - which u corresponds to which v?
  - if $u \leftrightarrow v$ matching known

    standard measures: accuracy, F1-score, etc.
- if $U \neq V$
  - need to *also* find best alignment
  - can have multiple $c_u \longrightarrow$ *same* r_v
  - can have multiple $r_v \longrightarrow$ *same* c_u

Intrinsic Evaluation: Supervised

Key Idea: Evaluate relationship between *pairs* of data points $\boldsymbol{x}_l, \boldsymbol{x}_m$

Rand Index (RI)

- $+:$ $\boldsymbol{x}_l, \boldsymbol{x}_m$ are in the same cluster
- $-:$ $\boldsymbol{x}_l, \boldsymbol{x}_m$ are in different clusters

Intrinsic Evaluation: Supervised

Key Idea: Evaluate relationship between *pairs* of data points $\boldsymbol{x}_l, \boldsymbol{x}_m$

Rand Index (RI)

- $+$: $\boldsymbol{x}_l, \boldsymbol{x}_m$ are in the same cluster
- $-$: $\boldsymbol{x}_l, \boldsymbol{x}_m$ are in different clusters

True ($\mathcal{R}$) \ Predicted (C)	+	−
+	TP	FN
−	FP	TN

$$\mathrm{RI} = \frac{TP + TN}{TP + TN + FP + FN}$$

$=$ Accuracy!

Intrinsic Evaluation: Supervised

Issue: Expected value of RI of two *random* partitions $\neq 0$ (or any constant)

Intrinsic Evaluation: Supervised

Issue: Expected value of RI of two *random* partitions $\neq 0$ (or any constant)

Adjusted Rand Index (ARI)

	$\boldsymbol{c}_1$	$\boldsymbol{c}_2$		$\boldsymbol{c}_U$	sum
$\boldsymbol{r}_1$	N_{11}	N_{12}	$\cdots$	N_{1U}	a_1
$\boldsymbol{r}_2$	N_{21}	N_{22}	$\cdots$	N_{2U}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\boldsymbol{r}_V$	N_{V1}	N_{V2}	$\cdots$	N_{VU}	a_V
sum	b_1	b_2	$\cdots$	b_U	N

$N_{ij} = |\boldsymbol{r}_i \cap \boldsymbol{c}_j|$ $\quad \binom{N}{2} = \frac{N(N-1)}{2}$

Intrinsic Evaluation: Supervised

Issue: Expected value of RI of two *random* partitions $\neq 0$ (or any constant)

Adjusted Rand Index (ARI)

	$\boldsymbol{c}_1$	$\boldsymbol{c}_2$		$\boldsymbol{c}_U$	sum
$\boldsymbol{r}_1$	N_{11}	N_{12}	$\cdots$	N_{1U}	a_1
$\boldsymbol{r}_2$	N_{21}	N_{22}	$\cdots$	N_{2U}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\boldsymbol{r}_V$	N_{V1}	N_{V2}	$\cdots$	N_{VU}	a_V
sum	b_1	b_2	$\cdots$	b_U	N

$N_{ij} = |\boldsymbol{r}_i \cap \boldsymbol{c}_j|$ $\quad \binom{N}{2} = \frac{N(N-1)}{2}$

$$\text{TP} = \sum_{ij} \binom{N_{ij}}{2}$$

$$\text{Expected RI} = \frac{1}{\binom{N}{2}} \left[\sum_v \binom{a_v}{2} \cdot \sum_u \binom{b_u}{2} \right]$$

$$\text{Max RI} = \frac{1}{2} \left[\sum_v \binom{a_v}{2} + \sum_u \binom{b_u}{2} \right]$$

$$\text{ARI} = \frac{\text{TP} - \text{Expected RI}}{\text{Max RI} - \text{Expected RI}}$$

Summary

- **Clustering:** Means of discovering structure / sub-groups in data

Summary

- **Clustering:** Means of discovering structure / sub-groups in data
- K-Means
  - Hard; Flat; Polythetic
  - Requires knowing K; search for best K
  - Fast; Iterative; Local Minima

Summary

- **Clustering:** Means of discovering structure / sub-groups in data
- K-Means
  - Hard; Flat; Polythetic
  - Requires knowing K; search for best K
  - Fast; Iterative; Local Minima
- Hierarchical Clustering
  - Hard; Hierarchical; Polythetic
  - Top-Down: Hierarchical K-Means
  - Bottom-Up: Agglomerative Clustering
  - multiple variants: single, complete, etc.

Summary

- **Clustering:** Means of discovering structure / sub-groups in data
- K-Means
  - Hard; Flat; Polythetic
  - Requires knowing K; search for best K
  - Fast; Iterative; Local Minima
- Hierarchical Clustering
  - Hard; Hierarchical; Polythetic
  - Top-Down: Hierarchical K-Means
  - Bottom-Up: Agglomerative Clustering
  - multiple variants: single, complete, etc.
- Evaluation
  - Unsupervised, Supervised, and Human-judgement driven