Statistical Learning

Clustering

Spring 2024

Unsupervised Learning

Unsupervised Learning

- No response variable Y, only $\{x_i\}_{i=1}^n$.
- Goal: learn patterns in X.
- Examples
  - Estimate the density, covariance, graph (network), etc. of X — could be difficult in high-dimensional settings.
  - Cluster analysis: identify multiple regions of the feature space that contain modes of density.
  - Dimension reduction: identify low-dimensional manifold within the feature space $\mathcal{X}$ that represents high data density.

Cluster Analysis

Cluster Analysis

- Group the dataset into subsets so that those within each subset are more closely related (similar) to each other than those objects assigned to other subsets. Each subset is called a cluster.
- Flat clustering vs. hierarchical clustering: flat clustering divides the dataset into k cluster, and hierarchical clustering arranges the clusters into a natural hierarchy.
- Clustering results are crucially dependent on the measure of similarity (or distance) between the "points" to be clustered.

Distance Metric

- A distance metric or a distance function is a function that defines the similarity of two elements (points, sets, etc.)
- For the distance of two points (with continuous entries), the most commonly used measurement is the Euclidian distance:

$$\begin{aligned} d(u, v) &= \|u - v\|_2 \\ &= \sqrt{\sum_{j=1}^{p} (u_j - v_j)^2} \end{aligned}$$

- For categorical entries, the Hamming distance is usually used

$$d(u, v) = \sum_{j=1}^{p} \mathbf{1}\{u_j \neq v_j\}$$

- Distance measures should be defined based on the application. There is no universally best approach.

Clustering

- Suppose we have a set of n data points.
- We want to form $K \ll n$ clusters, indexed by $k \in \{1, \ldots, K\}$.
- Let $C(\cdot)$ be a cluster index function that assign th ith observation or cluster $C(i)$.
- Consider: search for a function $C : \{1, \ldots, n\} \to \{1, \ldots, K\}$ to minimize the within cluster distance:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i), C(i')=k} d(x_i, x_{i'}).$$

Clustering

- This is equivalent to maximizing the between cluster distance

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d_{ii'}$$

- Note that the total distance can be broke down into

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} d_{ii'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left[\sum_{C(i')=k} d_{ii'} + \sum_{C(i')\neq k} d_{ii'}\right]$$
$$= W(C) + B(C)$$

- The total distance is fixed for a given set of data, hence

$$\text{minimize } W(C) \iff \text{ maximize } B(C)$$

Clustering

- Given a specific distance measure $d(\cdot, \cdot)$, several algorithms can be used to find the clusters
  - Combinatorial algorithm
  - K-means clustering
  - Hierarchical clustering

Combinatorial Algorithm

Combinatorial Algorithms

- For small n and K, we could minimize W by brute-force search.
- However, the number of "tries" needed to complete the search is

$$S(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n$$

- This is not feasible for large n and K, since the number of distinct assignments can be extremely large.
- It calls for more efficient algorithms: may not be optimal but a reasonably good suboptimal partition.

K-means Clustering

K-means Clustering

- Consider an enlarged optimization problem:

$$\min_{C, \{m_k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - m_k\|^2$$

- Hence, we are solving both
  - the cluster index function $C(\cdot)$,
  - and also the cluster centers m_k, $k = 1 \ldots K$.
- This problem is NP-hard for ≥ 2 dimensions.

K-means Clustering

- Combinatorial algorithm is too expensive.
- Instead, consider an algorithm that alternatively updates the two components:
  - C, the cluster assignments
  - $\{m_k\}_{k=1}^K$: the cluster means
- We will do an iterative update by:
  1) Fixing C, find the best $\{m_k\}_{k=1}^K$
  2) Fixing $\{m_k\}_{k=1}^K$, find the best C

K-means Clustering

- Fixing C, we know the cluster label of each subject. For any set $\{i : C(i) = k\}$, finding the mean is

$$m_k = \arg\min_m \sum_{C(i)=k} \|x_i - m\|^2.$$

- This is simply finding the mean within cluster k, i.e.,

$$m_k = \frac{\sum_{C(i)=k} x_i}{\sum_i \mathbf{1}\{C(i) = k\}}$$

K-means Clustering

- Fixing the cluster means $\{m_k\}_{k=1}^K$, to find the new cluster assignments, we simply recalculate the distance from an observation to each of the cluster mean.

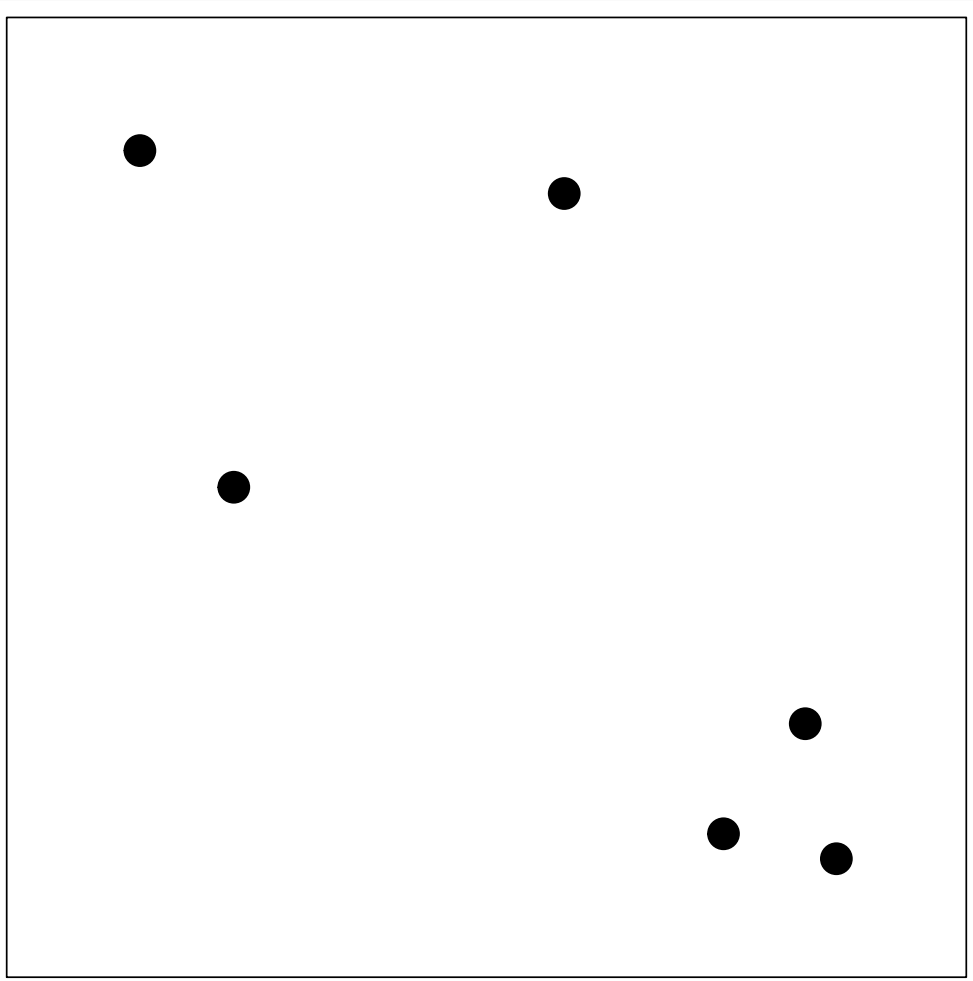
$$C(i) = \underset{k \in \{1,\ldots,K\}}{\arg\min} \; d(x_i, m_k)$$

- Hence each point will be assigned to the closest cluster mean

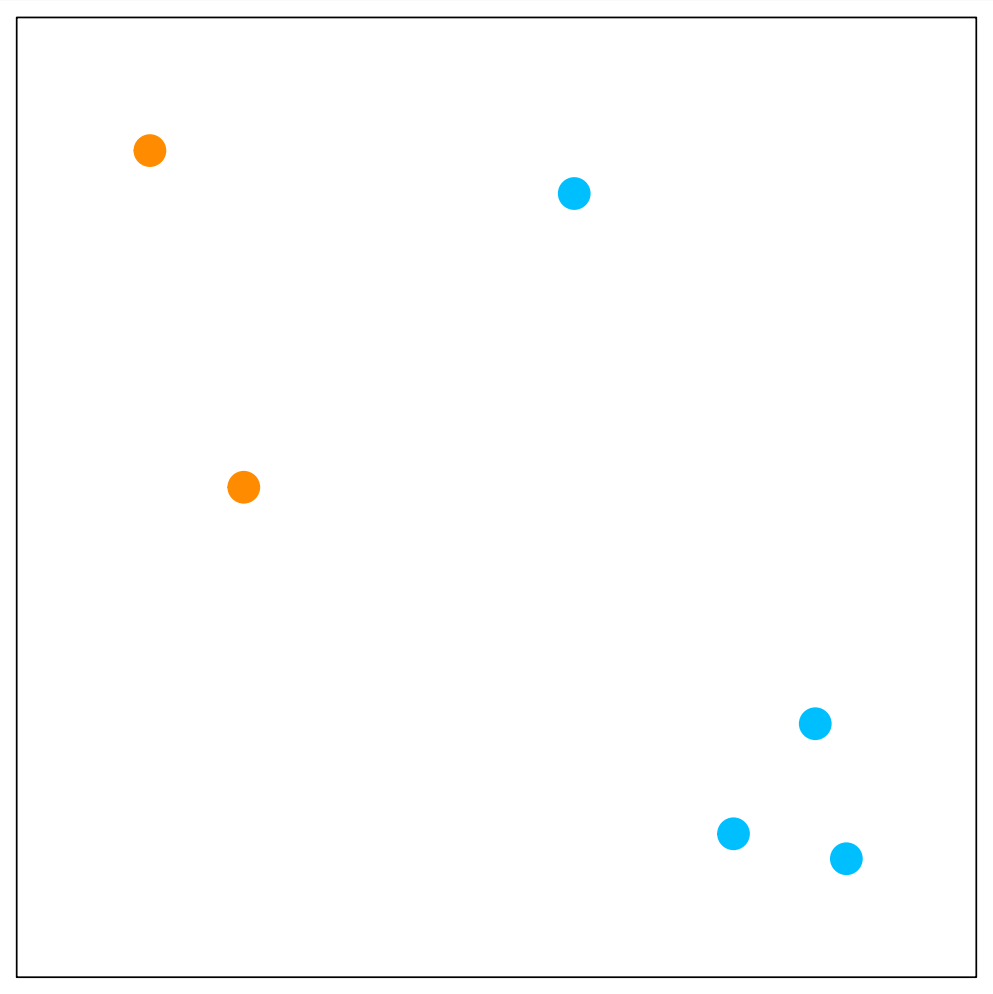
K-means Clustering

- A K-means Clustering algorithm:
  1) Randomly split the dataset into K different subsets. Assign each subsets a cluster label. Then iterate between 2) and 3).
  2) Given the cluster assignment C, calculate the cluster mean vectors $m_1, \ldots, m_K$.
  3) Given the current set of means $\{m_1, \ldots, m_K\}$, assign each observation to the closest current cluster mean.
- Stop the algorithm when C does not change

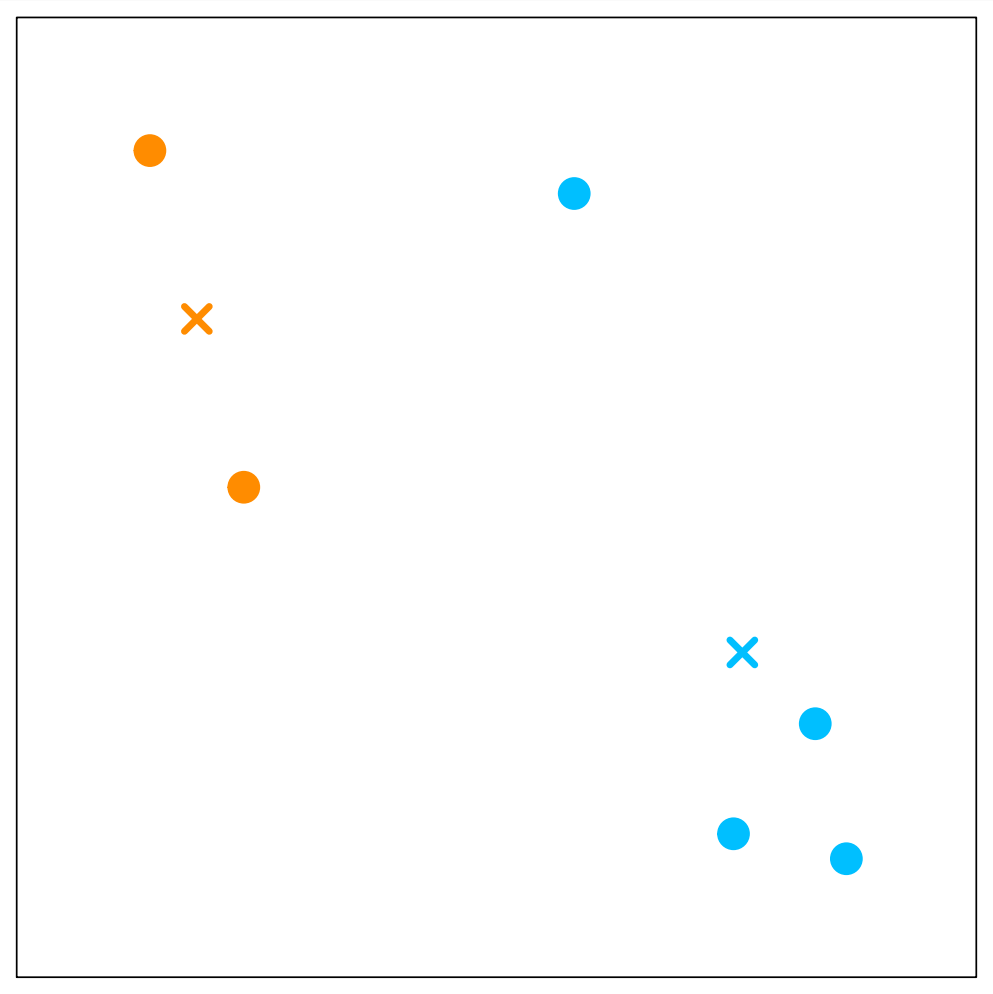
Demonstration



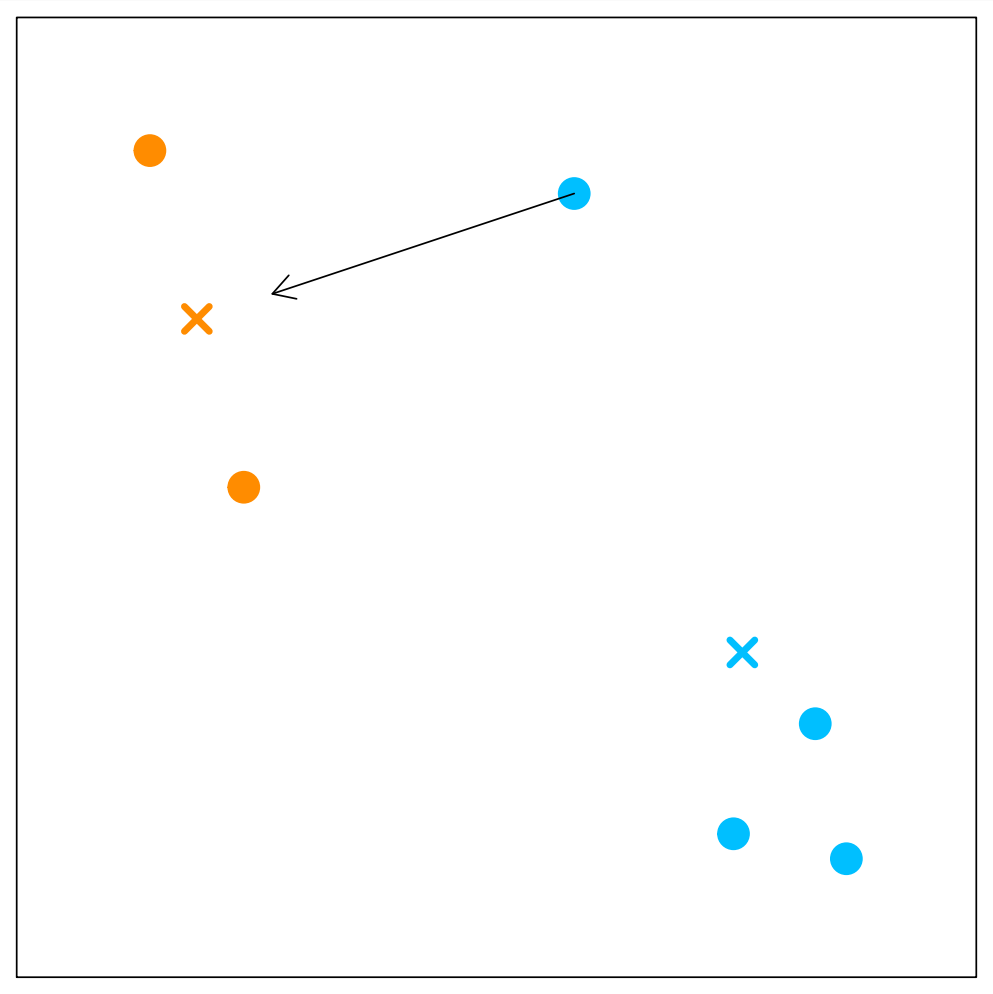
Demonstration



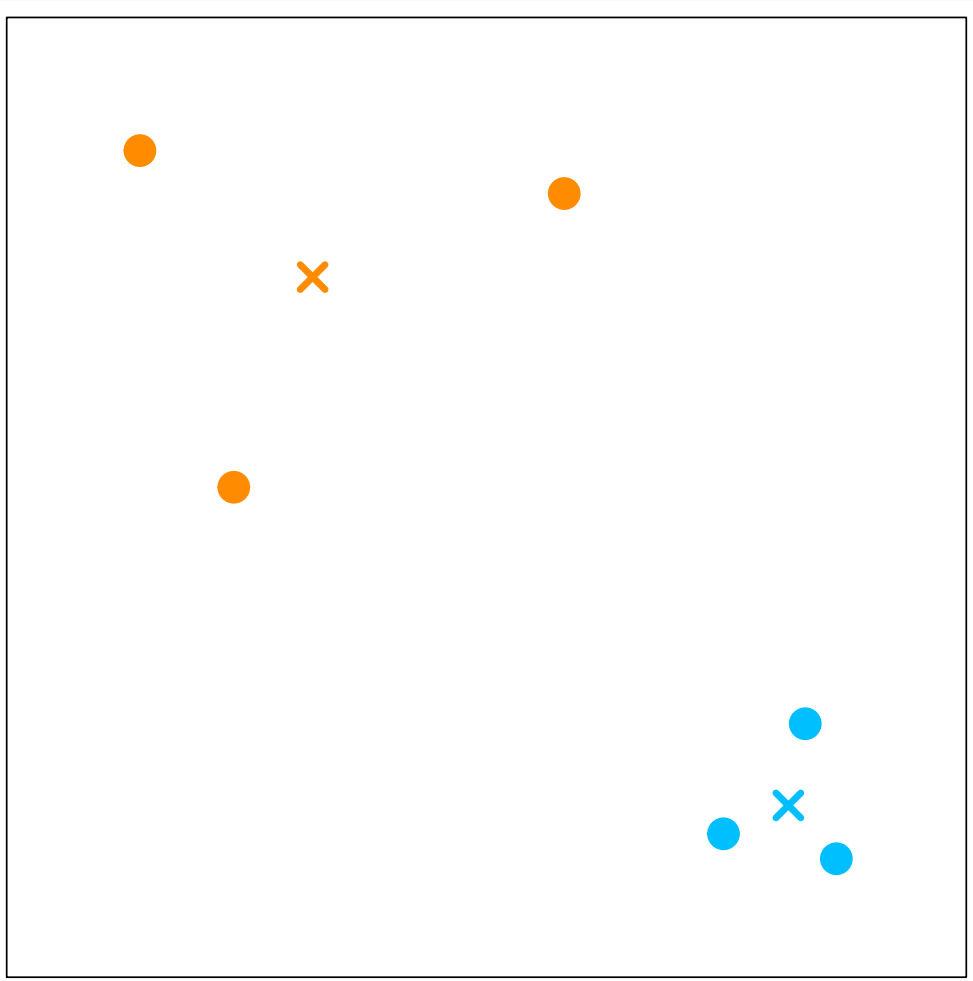
Demonstration



Demonstration



Demonstration



Alternative Version: K-medoids

- K-medoids is an alternative version of K-means:
- Replace the second step by searching for the one observation that minimizes the distance to all others in the cluster

$$i_k^* = \underset{i:C(i)=k}{\arg\min} \sum_{C(i')=k} D(x_i, x_{i'})$$

- Use $x_{i_k^*}$ as the "center" of cluster k.

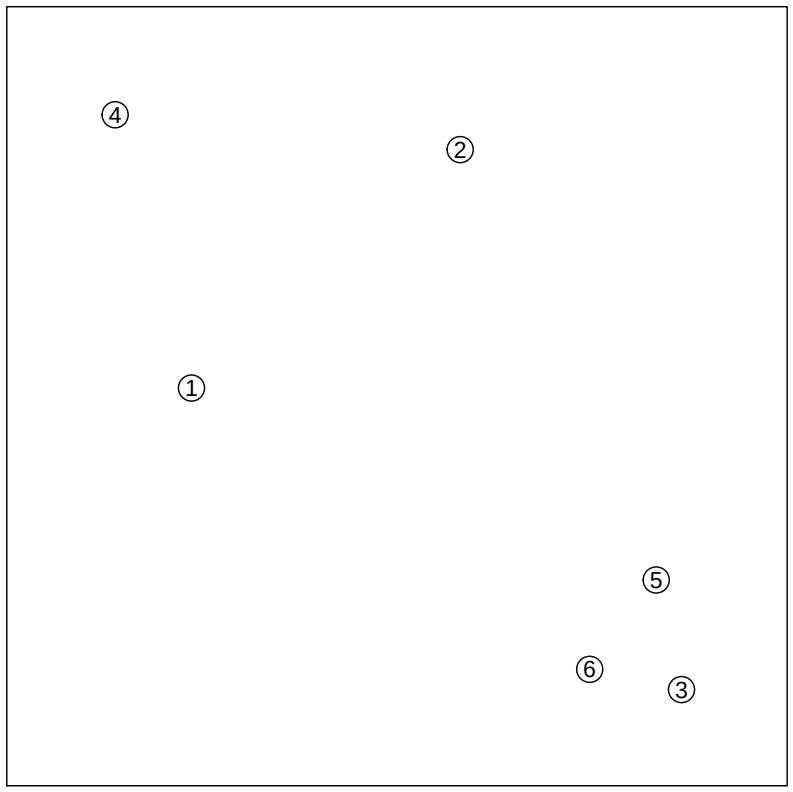
Hierarchical Clustering

Hierarchical Clustering

- Choosing the number of clusters K can be difficult.
- A hierarchical representation which
  - at the lowest level, each cluster contains a single observation.
  - at the highest level there is only one cluster containing all observations.
- Use dendrogram to display the clustering result.

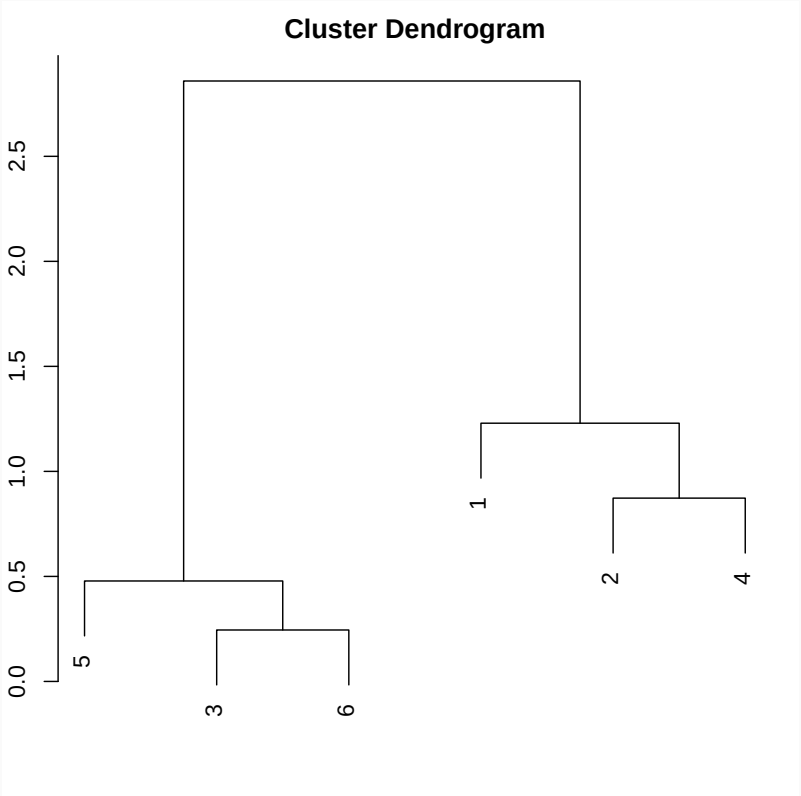
Hierarchical Clustering

- Suppose we have a set of 6 observations

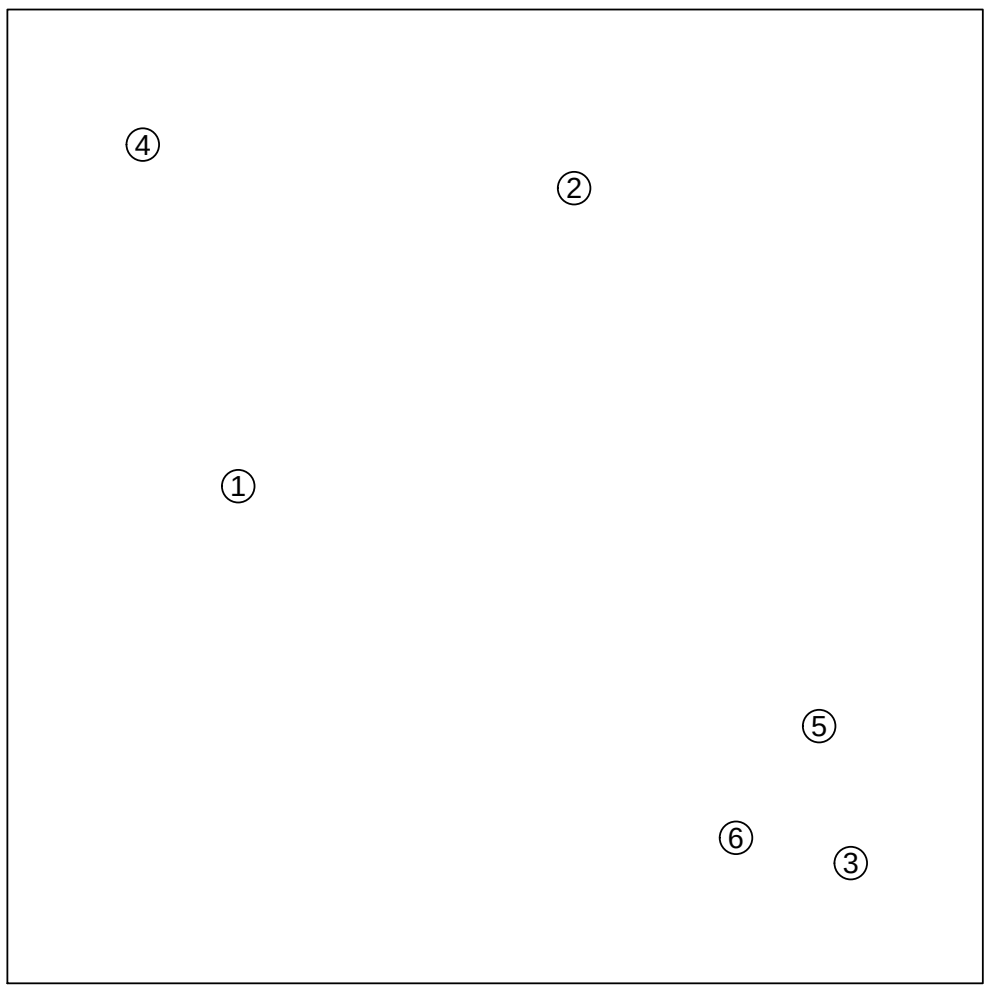


Dendrogram

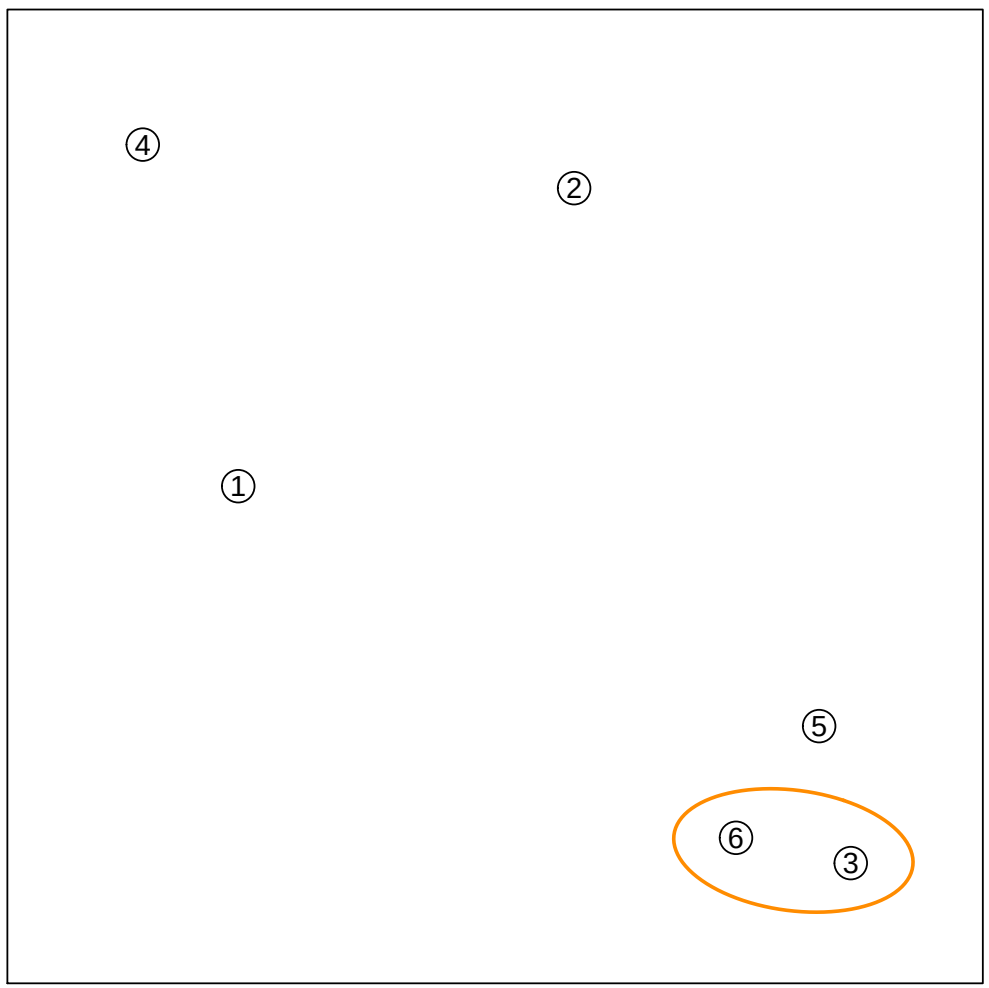
- A typical dendrogram from hierarchical clustering
- How to construct this?



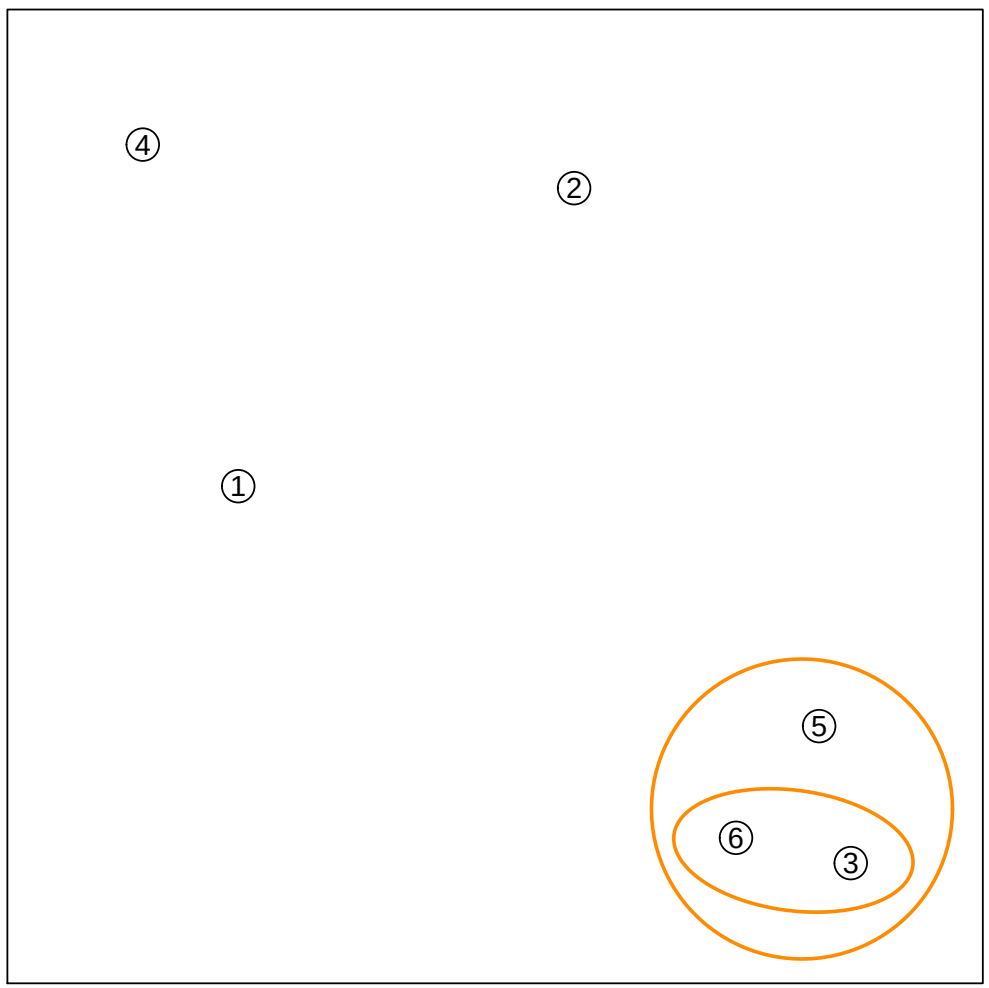
Demonstration



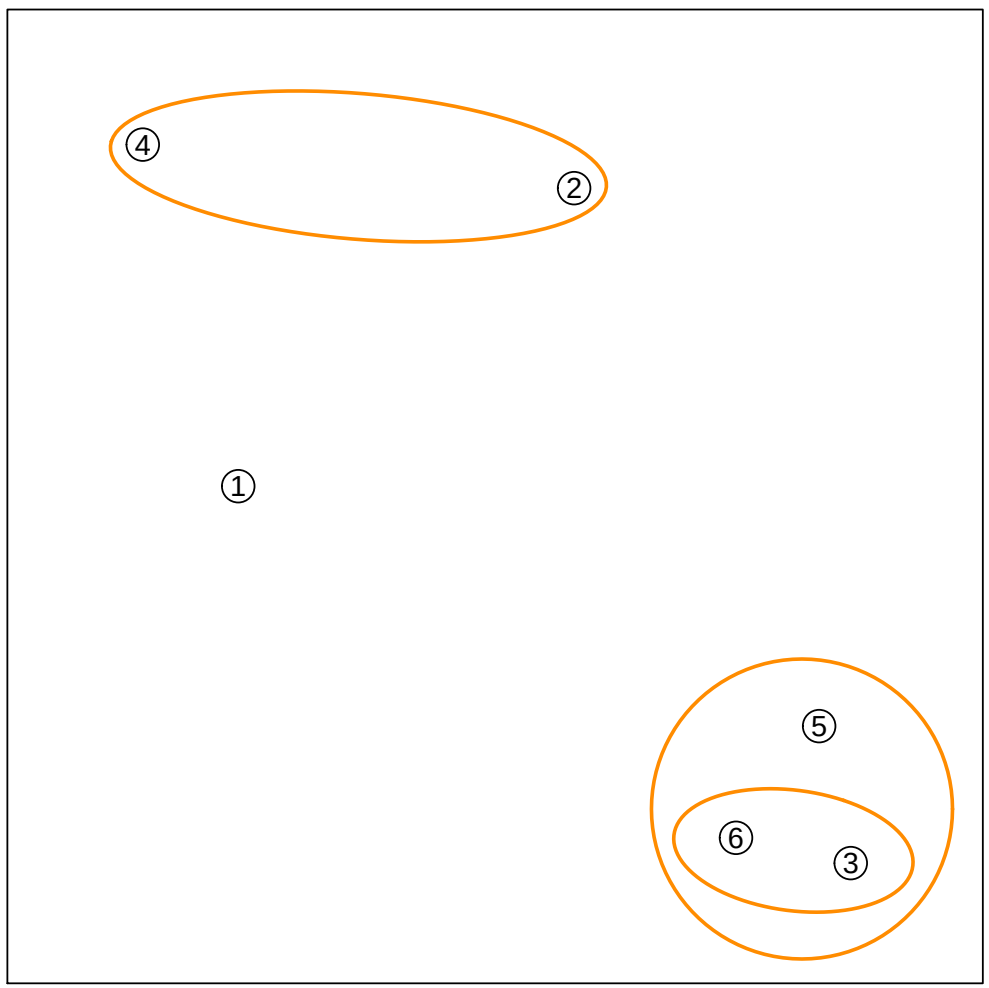
Demonstration



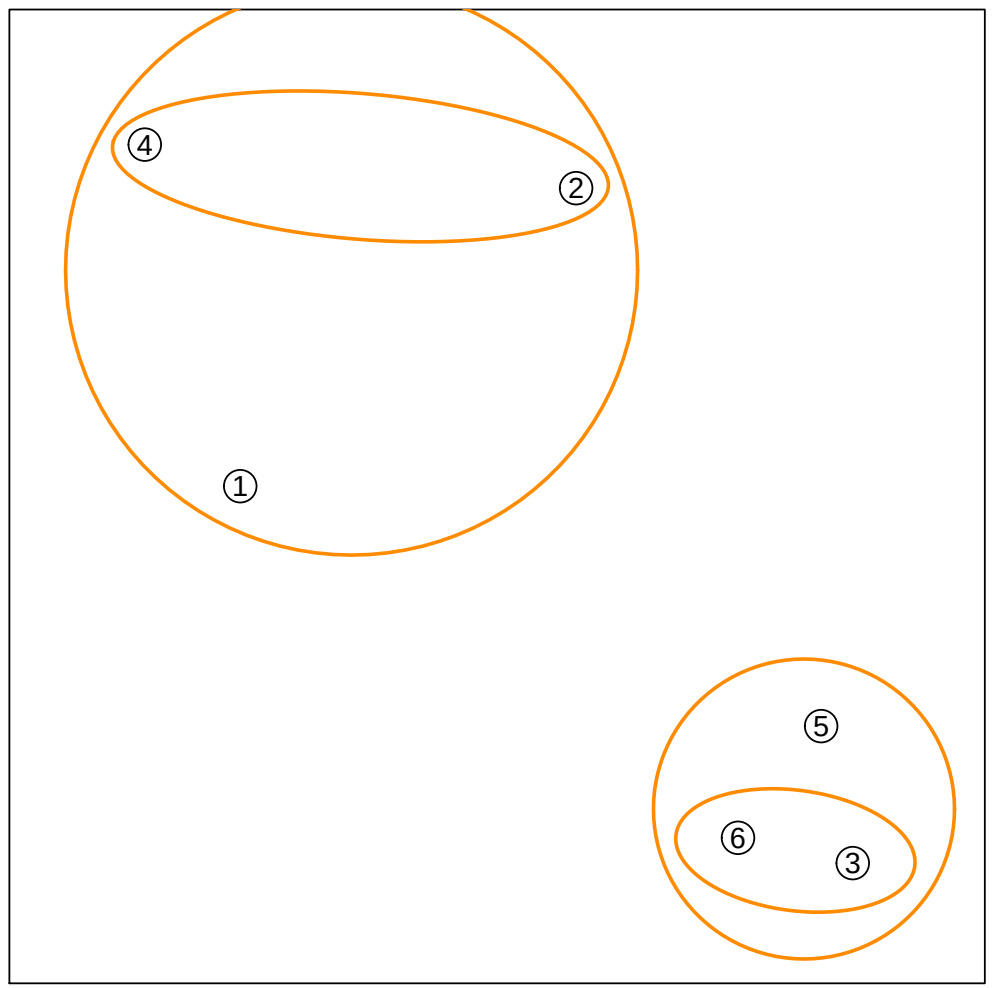
Demonstration



Demonstration



Demonstration



Algorithm

- An agglomerative algorithm is a "bottom up" approach:
  - Begin with every observation representing a singleton cluster.
  - At each step, merge two "closest" clusters into one cluster and reduce the number of clusters by one.
  - Stop when all observations are in the same cluster.
- How to choose which two clusters to merge?
- This requires:
  - A distance measure between any two observations $d(x_i, x_{j'})$
  - A distance measure between any two sets of observations $d(G, H)$

Distance Measures

- Distance $d(G, H)$ between two clusters G and H can be defined in different ways:
  - Complete linkage (default of `hclust()`): the furthest pair

$$d(G, H) = \max_{i \in G, i' \in H} d_{ii'}$$

  - Single linkage: the closest pair

$$d(G, H) = \min_{i \in G, i' \in H} d_{ii'}$$

  - Average linkage: average dissimilarity

$$d(G, H) = \frac{1}{n_G n_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$

- Different choices may result in different hierarchical structures

Distance Matrix

- To perform a hierarchical clustering, a matrix of all the pairwise distances is sufficient
- We don't have to know the values of the original observations
- This is an $n \times n$ matrix: the (i, i')'s element represents the distance between x_i and $x_{i'}$
- This matrix is also called a dissimilarity matrix.
  - symmetric
  - diagonal elements are zero

Spectral Clustering

Similarity Graph

- In some applications, we do not have the value of each data point, instead, we have the similarities between data points.
- Note: for similarity measures, larger means more similar, while for distance measures, larger is further away.
- A nice way to represent the data is the Similarity Graph $G = (V, E)$ — an undirected graph
  - V is a set of vertices: $\{x_1, x_2, \ldots, x_n\}$
  - E is the set of edges: $\{(i, j)\}_{ij}$

Similarity Graph

- For our case, this graph is weighted by an adjacency matrix

$$\mathbf{W}_{n\times n} = \{w_{ij}\}_{ij}$$

  - You can define $\mathbf{W}$ in many different ways
  - Each w_{ij} is the similarity between vertices i and j
  - $\mathbf{W}$ is symmetric: $w_{ij} = w_{ji}$
- We also define the degree matrix $\mathbf{D}$ as a diagonal matrix

$$\text{diag}(d_1, \ldots, d_n)$$

  where the d_i is the degree of vertex i:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

Graph Laplacian

- There are many different ways to define a graph Laplacian matrix. We give a few examples.
- Unnormalized graph Laplacian

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

- Normalized graph Laplacians

$$\mathbf{L}_{\text{sym}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$
$$\mathbf{L}_{\text{rw}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

- Each of them have some unique properties.

Unnormalized graph Laplacian

- For the unnormalized graph Laplacian, we have, for any f

$$\begin{aligned} f'\mathbf{L}f = f'\mathbf{D}f - f'\mathbf{W}f &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} \\ &= \frac{1}{2}\left\{ \sum_i d_i f_i^2 - 2\sum_{i,j} f_i f_j w_{ij} + \sum_j d_j f_j^2 \right\} \\ &= \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2 \end{aligned}$$

- We also have:
  - $\mathbf{L}$ is positive semi-definite
  - The smallest eigen-value is 0, with eigen-vector $\mathbf{1}$
  - The number of 0 eigen-values depends on the number of connected components

Algorithm

- The spectral clustering is very simple:
  - Construct a weighted adjacency matrix $\mathbf{W}$, using e.g.,

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

  - Compute the Laplacian $\mathbf{L}$ (or normalized versions $\mathbf{L}_{\text{sym}}$, $\mathbf{L}_{\text{rw}}$)
  - Compute the smallest k eigenvectors, denote them collectively as $\mathbf{V}_{n\times k}$
  - Treat $\mathbf{V}_{n\times k}$ as the matrix of the observed data, and perform k-means clustering
  - Output the k cluster labels