ML&MEA (2024)
Lecture 13 - Clustering

Quanying Liu

BME, SUSTech

2024.4.18

Content

PCA: 最大投影方差，最小重构误差

Objective: maximize the variance after projection:

$$\arg\min -\mathbf{z}_1^T S\mathbf{z}_1$$
$$\text{s.t. } \mathbf{z}_1^T\mathbf{z}_1 = 1$$

Let's set λ as a Lagrange multiplier. The generalized Lagrangian function is:

$$\mathcal{L}(\mathbf{z}_1, \lambda) = -\mathbf{z}_1^T S\mathbf{z}_1 + \lambda(\mathbf{z}_1^T\mathbf{z}_1 - 1)$$

The derivative of $\mathcal{L}$ is:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}_1} = -2S\mathbf{z}_1 + 2\lambda\mathbf{z}_1 = S\mathbf{z}_1 - \lambda\mathbf{z}_1 = 0$$

$\mathbf{z}_1^*$ is an eigenvector of S, $\lambda = \lambda_1$ is the largest eigenvalue.

LDA: 类内小，类间大

- Between-class (类间): $(\bar{z}_1 - \bar{z}_2)^2$
- Within-class (类内): $S_1 + S_2$
- Objective function: $J(\mathbf{w}) = \frac{(\bar{z}_1 - \bar{z}_2)^2}{S_1 + S_2}$
- We can derive the objective function as,

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$$

$$S_b = (\bar{\mathbf{x}}_{C_1} - \bar{\mathbf{x}}_{C_2})(\bar{\mathbf{x}}_{C_1} - \bar{\mathbf{x}}_{C_2})^T$$
$$S_w = S_{C_1} + S_{C_2}$$

$\mathbf{w}^*$ is the eigenvector of $S_w^{-1} S_b$.

LDA vs PCA

- **Similarity**:
  1. Both LDA and PCA reduce dimension.
  2. Both construct new features which are linear combination of original features.
  3. Both use Eigen Decomposition. (PCA: S; LDA: $S_w^{-1}S_b$)
- **Difference**:
  1. PCA is unsupervised learning, which does not consider class label. PCA finds components along maximum variability of the data.
  2. LDA is supervised, which considers the class label. LDA finds components to maximially separate the classes.

A gentle introduction to clustering

Clustering is a type of unsupervised learning technique used in ML to group a set of objects in such a way that objects in the same group (called a cluster) are more "similar" to each other than to those in other groups.

- High within-cluster similarity
- Low between-cluster similarity
- It is form of unsupervised learning

Unsupervised learning = learning from data (unlabeled, unannotated, etc), as opposed to supervised data where the labels of data are given.

Why clustering?

Two main uses:

- Pattern Discovery: looking for new insights into the structure of data. Clustering helps to identify inherent groupings within data. By discovering these patterns, we can understand the structure of data, which is especially valuable in exploratory data analysis. This process can reveal characteristics and behaviors in data that were not initially apparent. For example, find groups of patients that have similar symptoms.
- Dimensionality Reduction: deriving a reduced representation of the full data set. By summarizing or representing a large number of data points with a few clusters, we can simplify the data, making it easier to visualize and analyze.

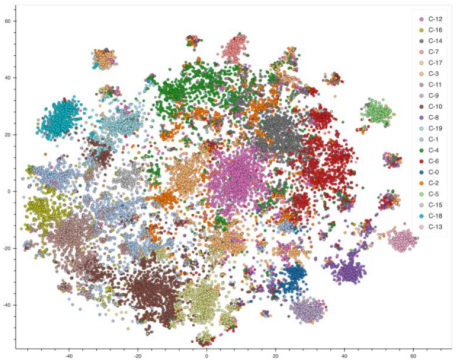
There are other uses too:

- Anomaly Detection
- Image Segmentation
- Summarization of Information

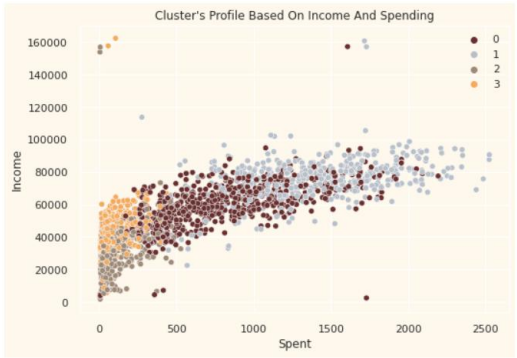
Applications of clustering

A widely-used and important task that finds many applications in Science, Engineering, Information Science, and other fields

- Group genes that perform the same function
- Group customers that share similar characteristics
- Group individuals that have similar political views
- Group documents that have similar topics



COVID-19 literature clustering



Customer segmentation

Anime recommendation

Definition of "groupness"

- Q: What is a natural grouping among these animals?

- Clustering is subjective!

What is similarity?

It is hard to define similarity, but...
"*We know it when we see it.*"

- The real meaning of similarity is a philosophical question. We will take a more mathematical approach.
- Depends on representation and algorithm. For many rep./alg., easier to think in terms of a distance between vectors (rather than similarity).

Desirable properties for distance measure

A good **distance** measure should have the following properties:

- Symmetry: $D(A, B) = D(B, A)$

  Otherwise you could claim "Amy looks like John, but John looks nothing like Amy"

- Constancy of Self-Similarity: $D(A, A) = 0$

  Otherwise you could claim "Amy looks more like John, than John does"

- Positivity Separation: $D(A, B) = 0$, if $A = B$

  Otherwise there are objects in your world that are different, but you cannot tell apart.

- Triangular Inequality: $D(A, B) \leq D(A, C) + D(B, C)$

  Otherwise you could claim "Amy is very like John, and Amy is very like Carl, but John is very unlike Carl."

Distance Measures: Minkowski Metric

- Suppose two feature vectors $\mathbf{x}$ and $\mathbf{y}$ both have p features

$$\mathbf{x} = (x_1, x_2, \ldots, x_p)^T$$
$$\mathbf{y} = (y_1, y_2, \ldots, y_p)^T$$

- The Minkowski metric:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{p} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

- Most commonly used Minkowski metrics:

Manhattan distance (r=1)	$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p} \|x_i - y_i\|$
Euclidean distance (r=2)	$d(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^{p} \|x_i - y_i\|^2)^{\frac{1}{2}}$
"sup" distance (r=∞)	$d(\mathbf{x}, \mathbf{y}) = \max_{1 \le i \le p} \|x_i - y_i\|$

Similarity measures: correlation coefficient

- Pearson correlation coefficient:

$$d(\mathbf{x}, \mathbf{y}) = \frac{Cov(\mathbf{x}, \mathbf{y})}{\sqrt{Var(\mathbf{x})}\sqrt{Var(\mathbf{y})}} = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{p}(y_i - \bar{y})^2}}$$

  where $\bar{x} = \frac{1}{p}\sum_{i=1}^{p} x_i, \quad \bar{y} = \frac{1}{p}\sum_{i=1}^{p} y_i.$

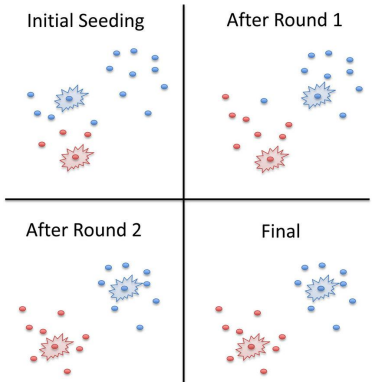
- Cosine distance:

$$d(\mathbf{x}, \mathbf{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} == \frac{\mathbf{x}^T\mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$$

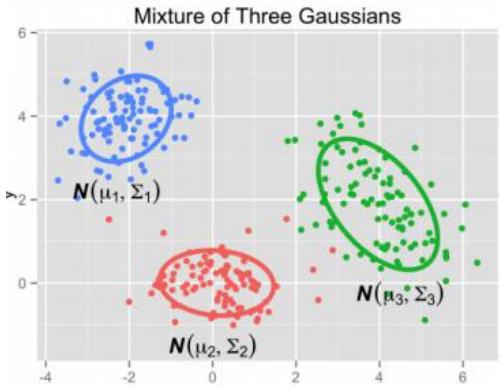
  Note: Cosine distance is in the range of [-1,1]

Clustering Methods

- Partitional methods
  - K-means clustering
  - Partitioning Around Medoids (PAM)
  - Self-Organizing Maps (SOM)
  - Fuzzy c-means ...
- Hierarchical methods
  - Bottom-up: Agglomerative Hierarchical Clustering
  - Top-down: Divisive Hierarchical Clustering



K-means clustering



Mixture-model clustering

Top-down and bottom-up clustering

Partitional method: K-means clustering

- Given n data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^p$.
- Let K be the number of clusters.
- n_k denotes the number of points in cluster k.
- **Objective**: learn the clustering function $C()$ to minimize the within-cluster variation (distance measures). $C(i) = k$ means that $\mathbf{x}_i$ is assigned to group k.
- Let us take the Euclidean distance as example:

$$\min \sum_{k=1}^{K} \sum_{C(i)=k} \|\mathbf{x}_i - c_k\|_2 \tag{1}$$

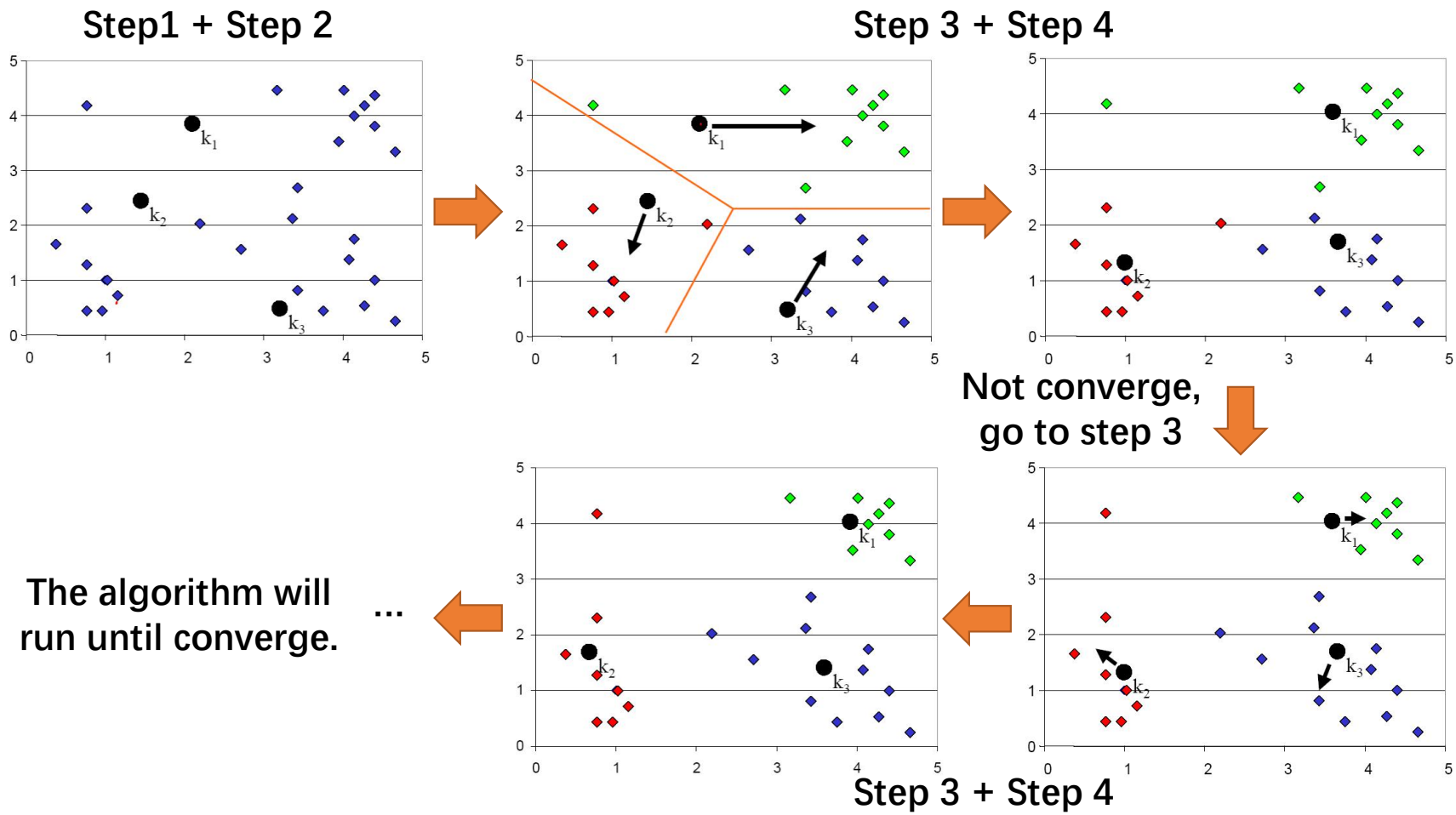
where c_k is the centroid of group k:

$$c_k = \frac{1}{n_k} \sum_{C(i)=k} \mathbf{x}_i \tag{2}$$

K-means clustering: Algorithm

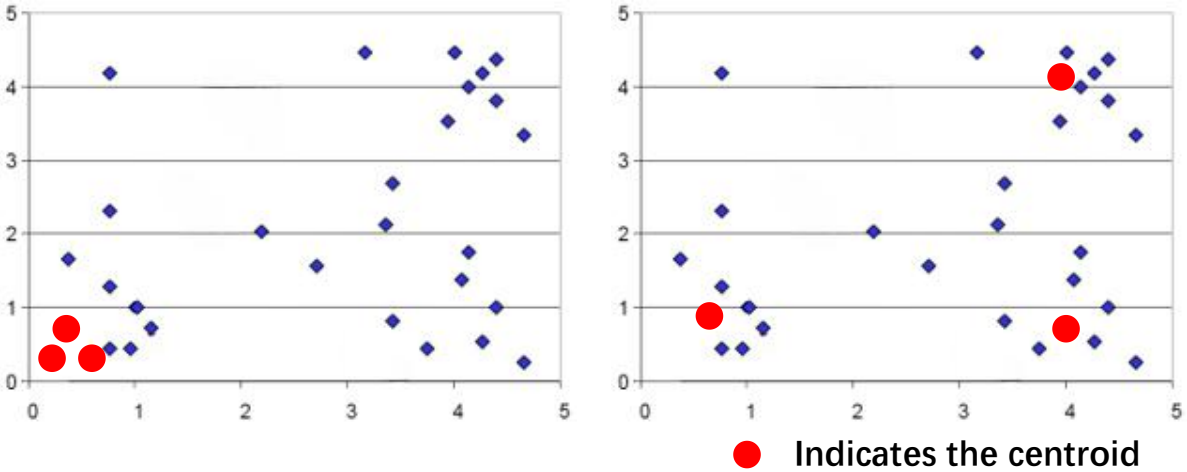
- **Algorithm procedure:**
  1. Decide on a value for K. (Q: how to choose a good K?)
  2. Randomly initialize k cluster centroids c_k.
  3. Fix c_k, calculate distance between data point $\mathbf{x}_i$ and c_k.
  4. Update class memberships $C(i)$ by assigning each point to the **nearest** cluster centroids (aka the center of gravity or mean)
  5. Fix $C(i)$, update the cluster centroids c_k using Eq. (2).
  6. If none of the n points changed membership in the last iteration (aka converge), exit. Otherwise go to **step 3**.
- A lazy solution: from sklearn.cluster import kMeans

K-means clustering: step-by-step visualization



K-means clustering: Selection of initial seeds c_k

- Clustering results may vary according to the random seed selection.



- Some seeds can result in poor convergence rate, or convergence to sub-optimal result.
- Some solutions to avoid this problem:
  - Select good seeds using some prior knowledge.
  - Try out multiple starting points (very important!!!).
  - Initialize seeds with the results of another method.

K-means clustering: Selection of Cluster number K

Q: How to choose the number of Clusters K?

- Sometimes, the number of clusters K is given on the context.
- Sometimes, we have to search for the "right" number of clusters, *e.g.*, grid search $K = 1, 2, \ldots, 10$
- Solve an optimization problem: penalize having lots of clusters
  - Application-dependent
  - Information theoretic approaches: model-based approach
- Tradeoff between the clustering error and the number of clusters

K-means clustering: good v.s. bad

What Is A Good Clustering? How to quantify it?

- Internal criterion: A good clustering will produce high-quality clusters in which
  - The intra-class (that is, intra-cluster) similarity is high;
  - The inter-class similarity is low;
  - The measured quality of a clustering depends on both the obj representation and the similarity measure used.
- External criteria for clustering goodness:
  - Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data.
  - Assesses a clustering with respect to ground truth.
    - Purity: The ratio between the dominant class in the cluster and the size of cluster.
    - Entropy of classes in clusters (or mutual information between classes and clusters).

Partitional methods: Others

- Partitioning around medioids (PAM): Instead of averages, use multidim medians as centroids. (cluster "prototypes").
- Self-organizing maps (SOM): Add an underlying "topology" (neighboring structure on a lattice) that relates cluster centroids to one another.
- Fuzzy c-means: Allow for a "gradation" of points between clusters; soft partitions.
- Mixture-based clustering: Implemented through an EM (Expectation-Maximization) algorithm. This provides soft partitioning, and allows for modeling of cluster centroids and shapes.

Hierarchical methods

There are two types of hierarchical clustering algorithms: agglomerative, and divisive.
In agglomerative, or bottom-up hierarhical clustering, the procedure is as follows:

- Start with all points in their own group.
- Until there is only one cluster, repeatedly: merge the two groups that have the smallest dissimilarity.

In divisive, or top-down hierarchical clustering, the procedure is as follows:

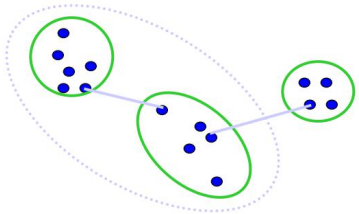
- Start with all points in one cluster.
- Until all points are in their own cluster, repeatedly: split the group into two resulting in the biggest dissimilarity.

Agglomerative strategies are generally simpler than divisive ones, so we'll focus on them.

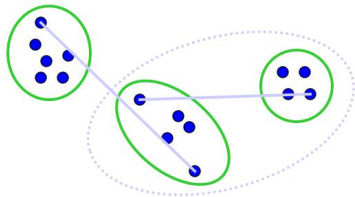
Measurements of dissimilarity between clusters

The dissimilarity between two clusters is defined as the distance between them:

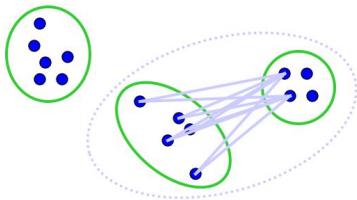
- Single link: cluster distance = distance of two closest members between two clusters
- Complete link: cluster distance = distance of two farthest members between two clusters
- Average link: cluster distance = average distance of all pairs



Single link



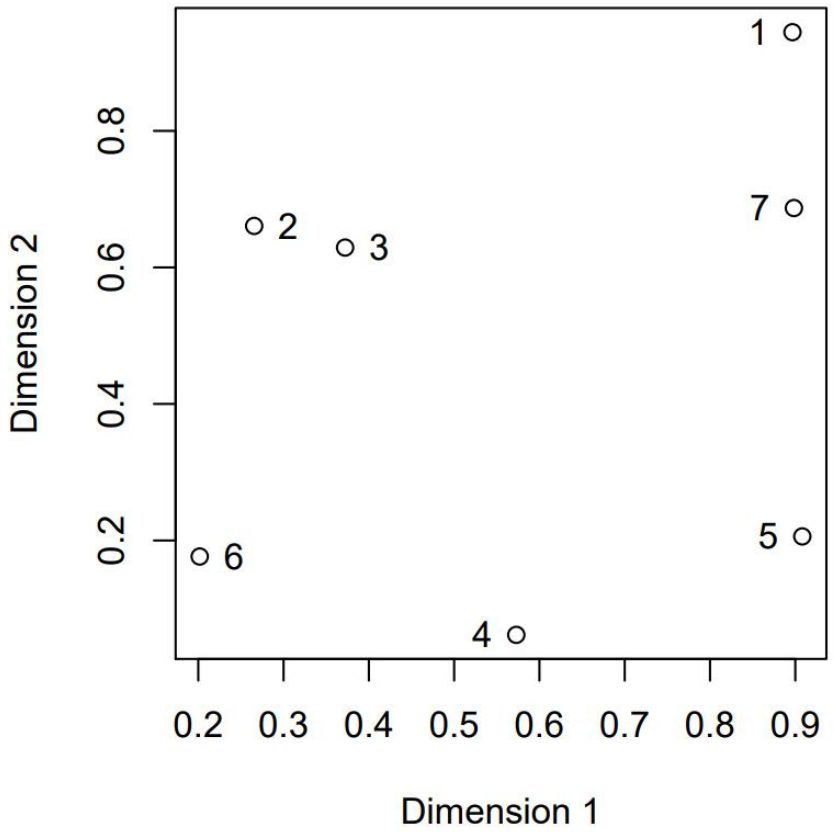
Complete link



Average link

Hierarchical methods: Agglomerative

Given the data points on the left, an agglomerative hierarhical clustering algorithm might decide on the clustering sequence described on the right.



Step 1: {1}, {2}, {3}, {4}, {5}, {6}, {7};

Step 2: {1}, {2, 3}, {4}, {5}, {6}, {7};

Step 3: {1, 7}, {2, 3}, {4}, {5}, {6};

Step 4: {1, 7}, {2, 3}, {4, 5}, {6};

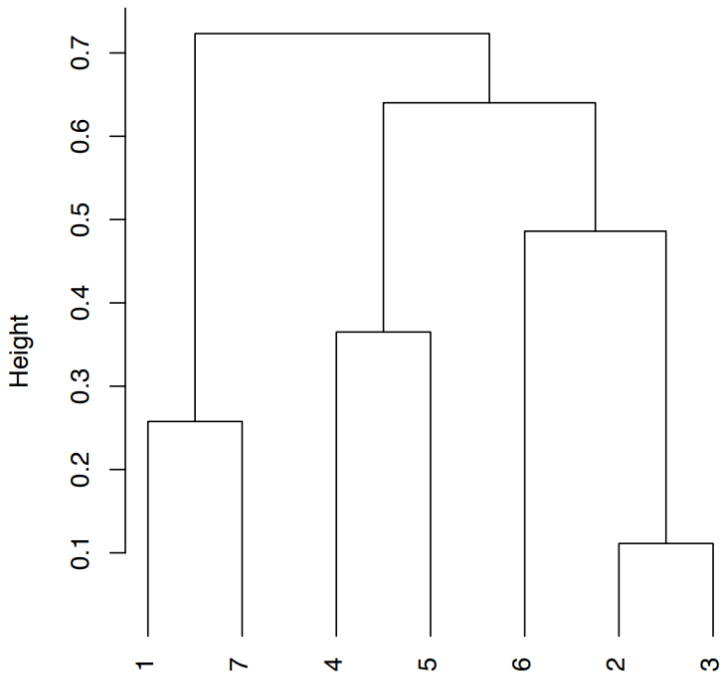
Step 5: {1, 7}, {2, 3, 6}, {4, 5};

Step 6: {1, 7}, {2, 3, 4, 5, 6};

Step 7: {1, 2, 3, 4, 5, 6, 7}.

Hierarchical methods: Dendrogram

We can also represent the sequence of clustering assignments using what is called a *dendrogram*:



In general, a dendrogram is a convenient graphic to display a hierarchical sequence of clustering assignments. It is simply a tree where:

- Each node represents a group
- Each leaf node is a singleton (i.e., a group containing a single data point)
- Root node is the group containing the whole data set
- Each internal node has two daughter nodes (children), representing the the groups that were merged to form it

Hierarchical methods vs Partitional methods

Hierarchical clustering vs. K-means:

- Hierarchical clustering: The algorithm can produces a consistent result, without the need to choose initial centroids (number of clusters). It also fits a sequence of clustering assignments, one for each possible number of underlying clusters $K = 1, \ldots, n$.
- K-means clustering: The algorithm fits exactly K clusters. The final clustering assignment depends on the chosen initial centroids (as specified).

Summary of Clustering

Clustering is the task of dividing up data points into groups or clusters, so that points in any one group are more "similar" to each other than to points outside the group.

Two types of clustering:

- Partitional algorithms
  - K-means clustering algorithm
  - PAM / SOM / Fuzzy-c means / Mixture-model based clustering ...
  - Common procedure:
    - Usually start with a random partitioning.
    - Refine it iteratively.
- Hierarchical algorithms
  - Bottom-up: Agglomerative
  - Top-down: Divisive

Watch from 8min: `https://www.bilibili.com/video/BV1Bg411Z77N/`