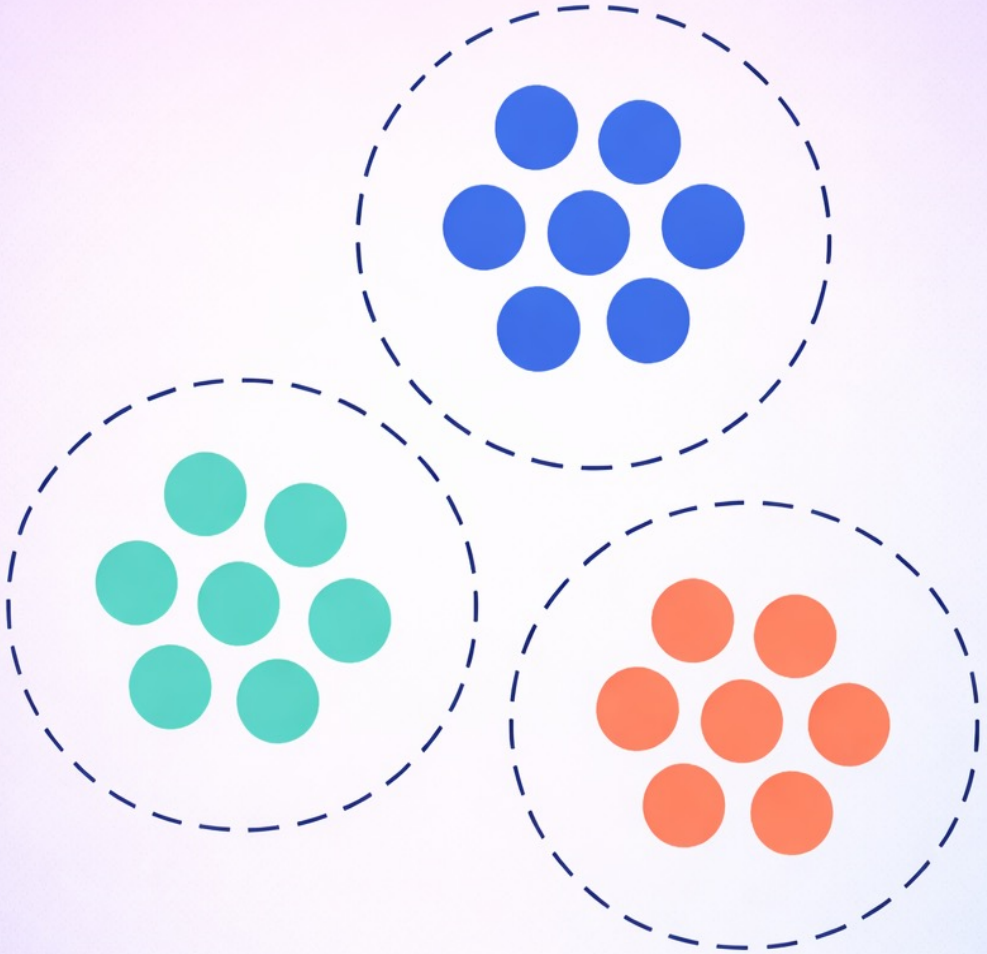
Georgia Tech

Clustering



Mohsen Moghaddam, Ph.D.

Gary C. Butler Family Associate Professor
H. Milton Stewart School of Industrial and Systems Engineering
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

SAIL
Symbiotic & Augmented Intelligence Laboratory

Spring 2026

Learning Outcomes

- Explain clustering as an unsupervised learning problem and distinguish it from supervised learning, including the role of distance and similarity choices
- Represent diverse data types (images, objects, documents, text) as feature vectors and analyze how representations affect clustering results
- Formulate and implement the k-means algorithm, including its objective, iterative assign–update steps, and convergence to local optima
- Compare standard and generalized k-means by varying distance metrics and cluster prototypes (e.g., mean, median)
- Analyze and apply different distance and similarity functions and evaluate how these choices influence clustering behavior and outcomes
- Interpret hierarchical clustering using linkage methods and dendrograms to select meaningful cluster structures

Motivation & Problem Setup

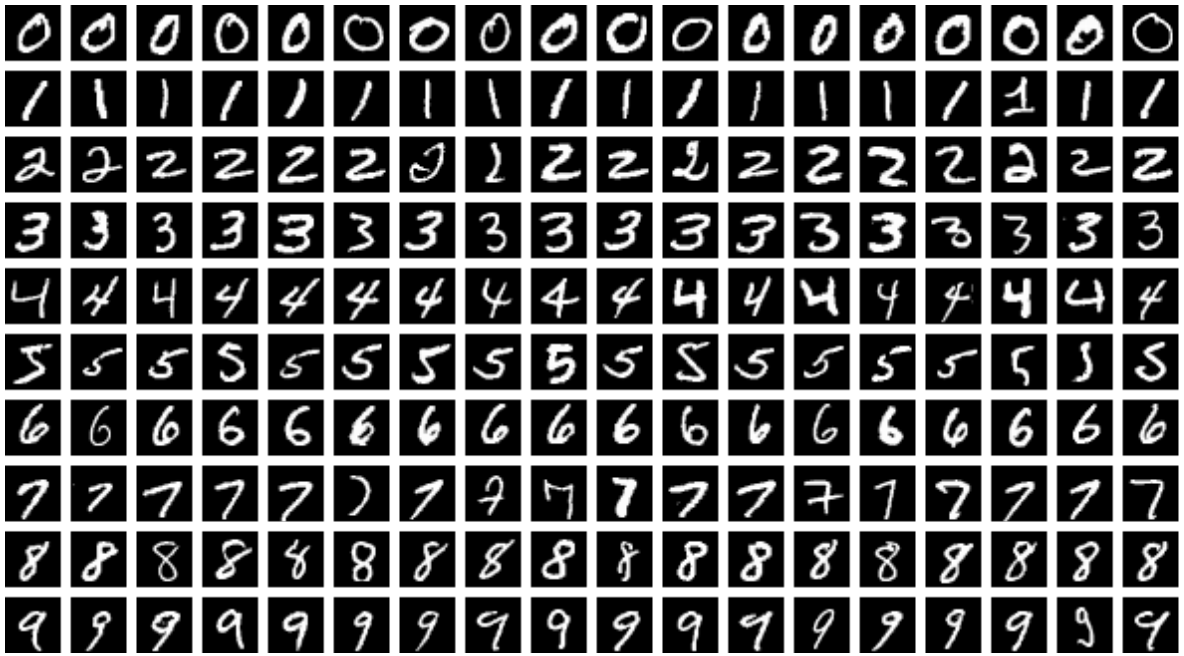
Goal of Clustering

Partition data into groups such that objects within a group are more **similar** to each other than to objects in other groups

Unlike classification or regression, clustering has no ground truth labels

Example: Handwritten Digits

Even without labels, digit images often cluster by visual similarity—clustering can approximate **latent** classes or sub-classes



Clusters may align with labels but are not given them (“latent classes”)

Core Concepts & Methods

- **Similarity/Distance:** Clustering relies on a measure of similarity (or distance) between data points.

  → **Common measures:** Euclidean distance, Manhattan distance, cosine similarity—the choice of measure affects how clusters are formed

- **Unsupervised Nature:** No training labels are provided—the algorithm must discover hidden patterns in the data

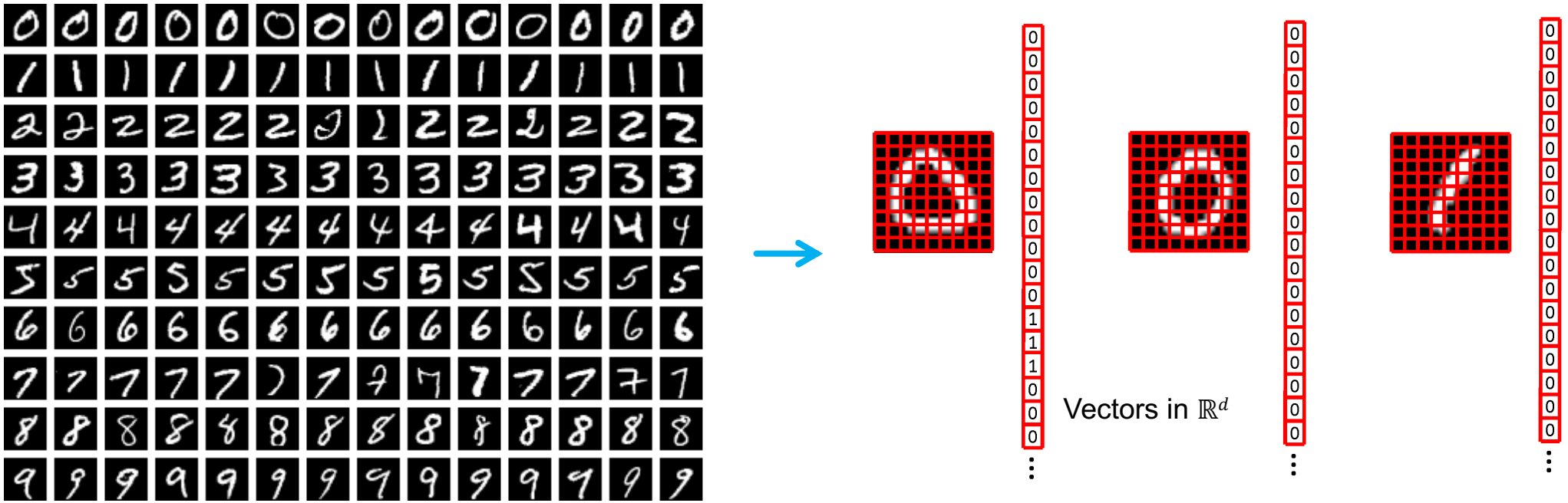
- **Cluster Quality:** Good clusters have **high intra-cluster similarity** (points close within a cluster) and **low inter-cluster similarity** (clusters well-separated)

- **Clustering Methods:** Two common clustering methods include partitioning methods, which split the data into groups (e.g., K-means), and hierarchical methods, which build a tree of clusters (e.g., agglomerative clustering)

Data Representation & Similarity

How to Represent Data?

The quality of clustering depends critically on how data are represented; i.e., what **mathematical objects** the similarity measures are computed on

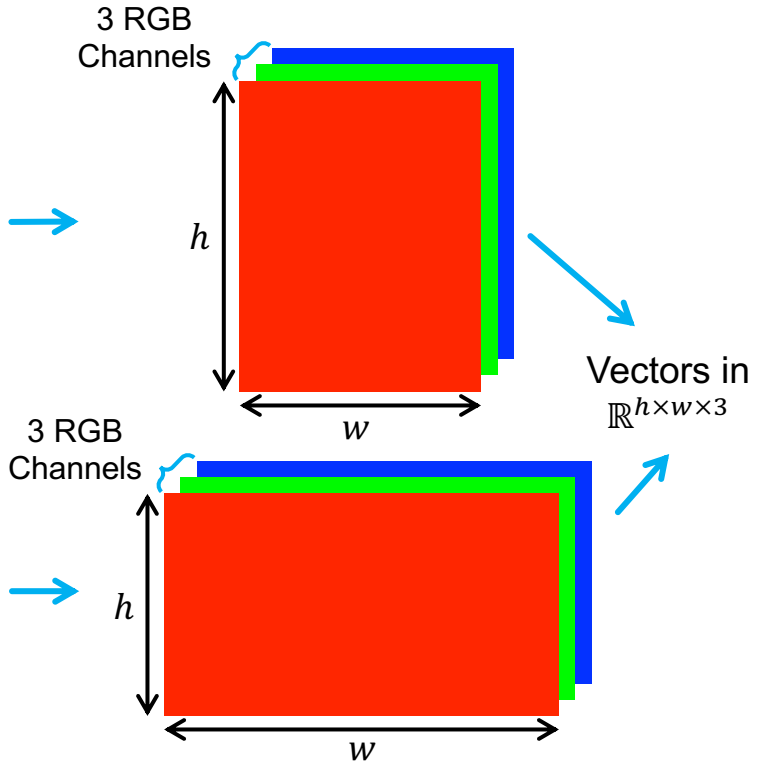


Raw data are often transformed into feature vectors before clustering

Images of Different Sizes

Real-world data often require preprocessing, as clustering algorithms typically assume fixed-length vectors:

- **Resizing or normalization:** Putting all data on the same scale
- **Feature extraction:** Pulling out meaningful numeric descriptors
- **Dimensionality reduction:** Keeping only the most informative features



What could go wrong if we cluster images without resizing them?

Objects in Real Life

Objects may be complex, structured, or heterogeneous—**feature engineering** is often the most important step in practical clustering

Horsepower	450 hp
Weight	3,800 lbs
MPG	18
Price	$43,000

Horsepower	510 hp
Weight	4,065 lbs
MPG	118
Price	$53,000

Horsepower	382 hp
Weight	3,400 lbs
MPG	27
Price	$46,000

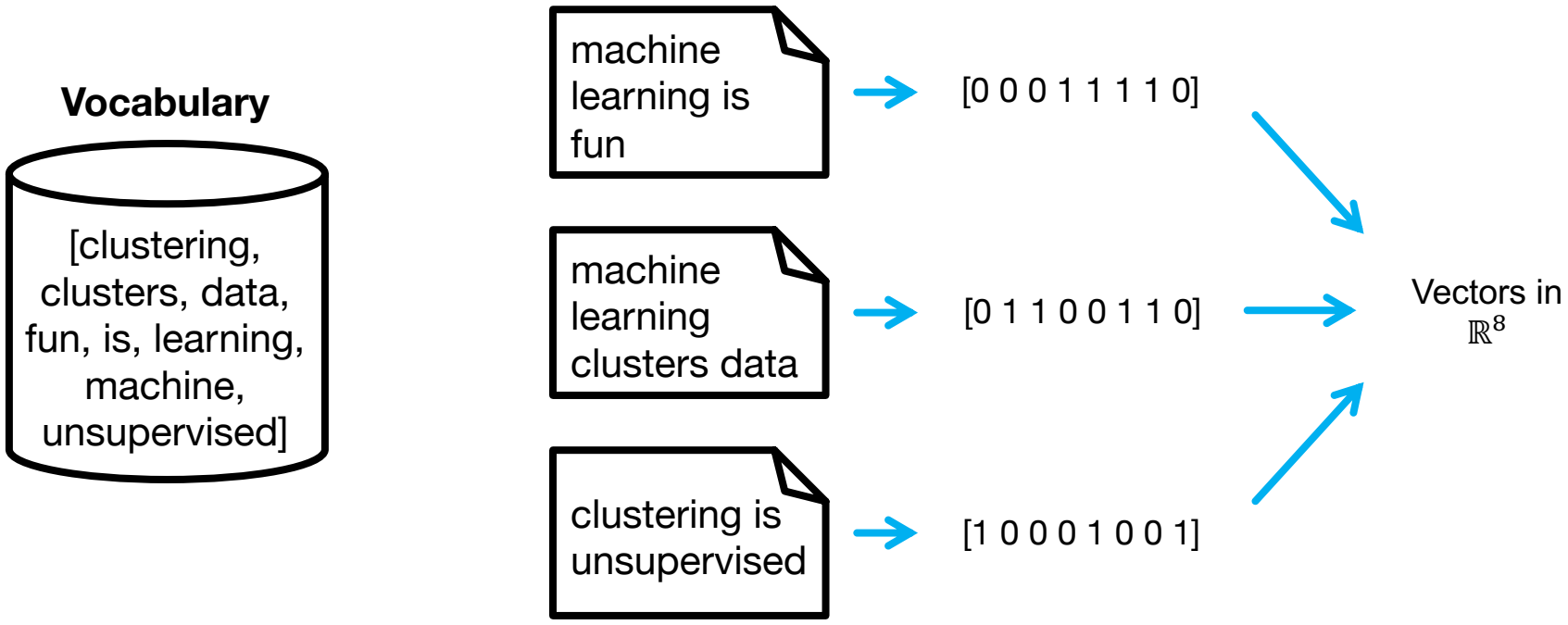
How do we select features? How do we scale them? Which feature might dominate?

Vectors in $\mathbb{R}^4$

More features don't always help—noisy or redundant ones yield poor clusters

Document Collections

Text documents lack a natural numeric representation—they must be converted into vectors before clustering

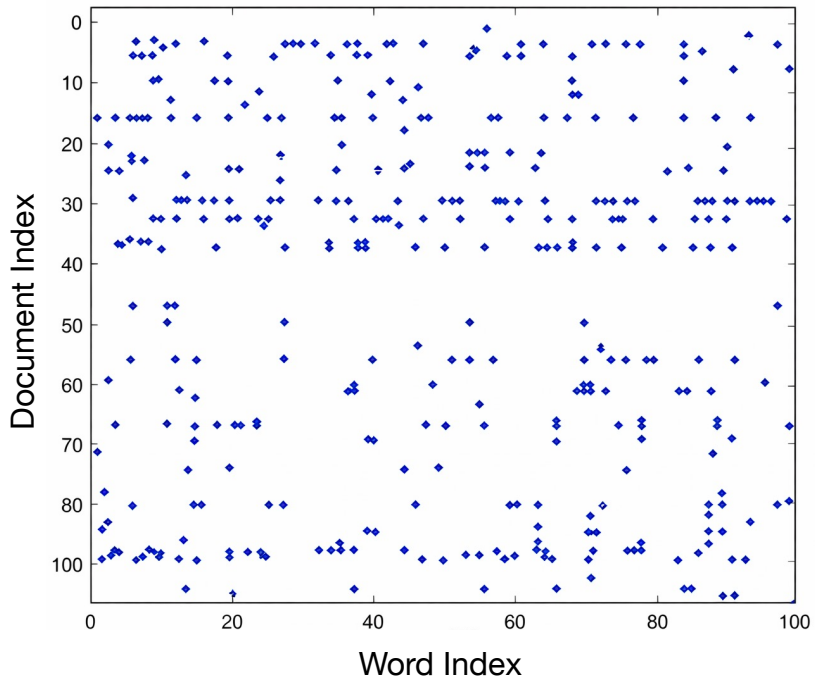


This representation emphasizes "word count" but loses "word order"

Bag-of-Words Matrix

Document clustering is often used for topic discovery, corpus organization, or exploratory data analysis

Bag-of-words or term-document matrix → $X =$



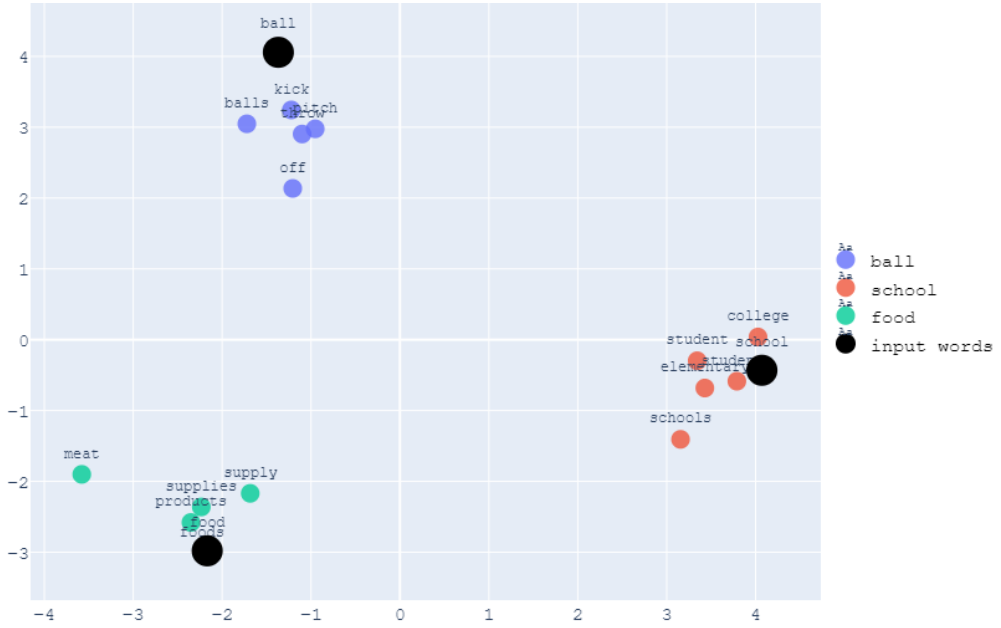
$$X_{ij} = \begin{cases} 1, & \text{word } j \text{ appears in document } i \\ 0, & \text{otherwise} \end{cases}$$

This is a sparsity plot (aka, "spy plot" in numerical linear algebra)

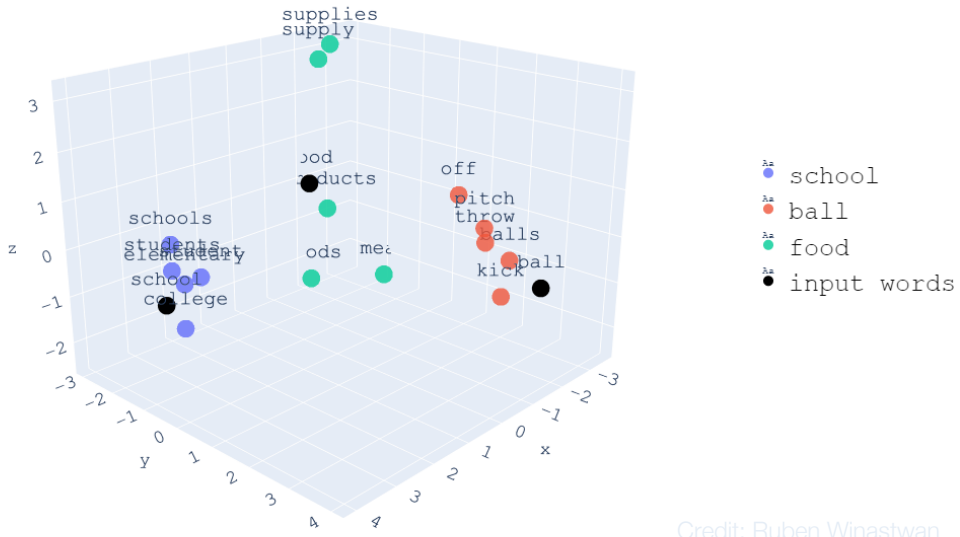
Bag-of-Words Representation

Each document is represented as a vector of word counts or frequencies

2D Word Embedding with PCA



3D Word Embedding with PCA

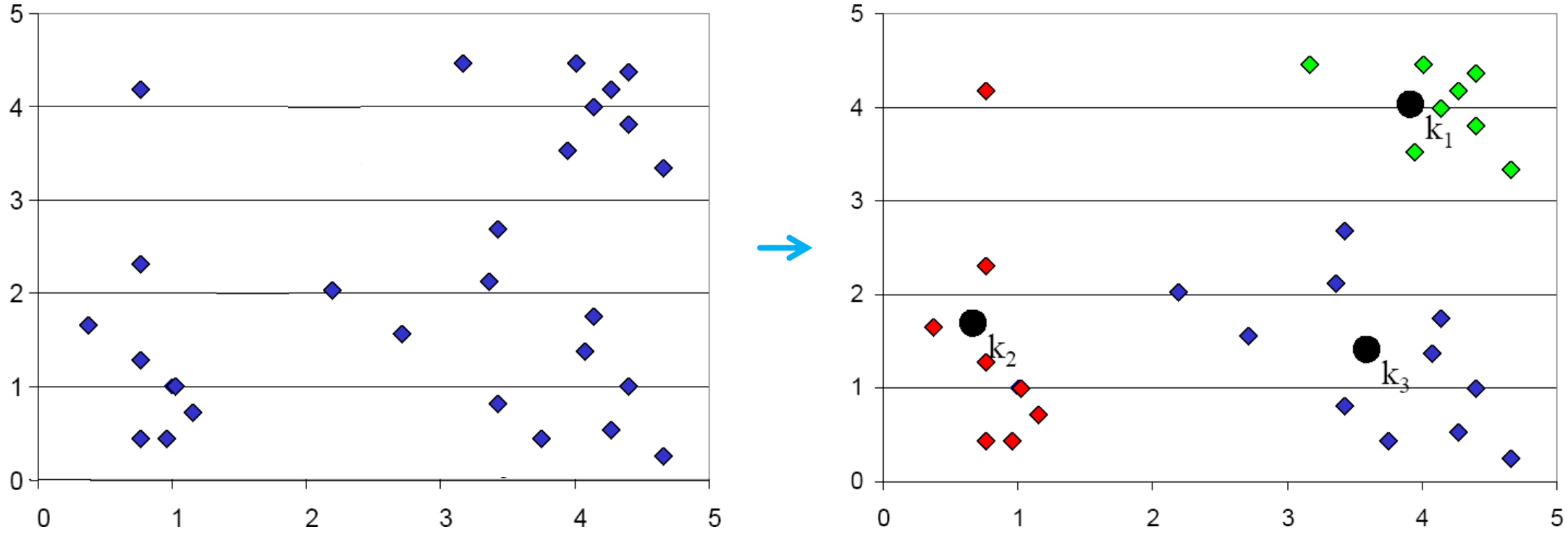


Credit: Ruben Winastwan

High dimensionality and sparsity challenge distance-based clustering

Before/After Clustering

Clustering assigns each data point to a group based on similarity, producing an **interpretable structure** from unorganized data



K-means Clustering

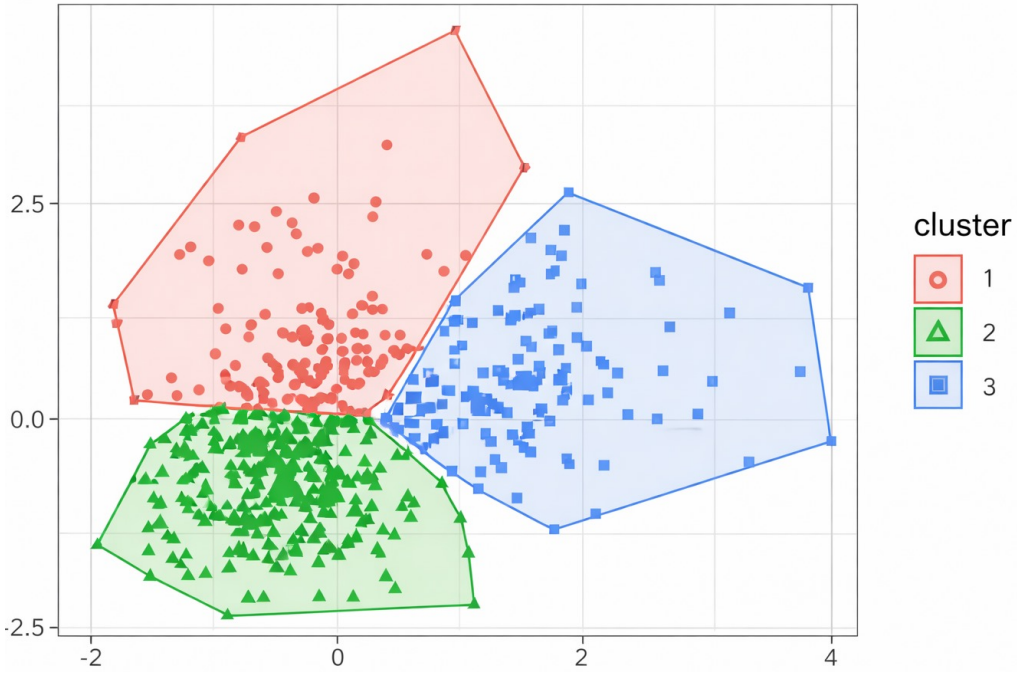
What is K-means Clustering?

K-means Clustering: Given m data points $\{x^1, x^2, \ldots, x^m\} \in \mathbb{R}^n$

- Find k cluster centers $\{c^1, c^2, \ldots, c^k\} \in \mathbb{R}^n$
- Assign each data point i to **one cluster**

$$\pi(i) = \arg\min_{j=1,\ldots,k} \left\| x^i - c^j \right\|^2, \qquad \forall i$$

Squared Euclidean distance: $\sum_{d=1}^{n} \left(x_d^i - c_d^j\right)^2$



Assumptions: Clusters are compact, roughly spherical, and similar in size

K-means partitions data into k clusters by minimizing within-cluster variance

K-means Algorithm

Input: m data points $\{x^i\}_{i=1}^m$, $x^i \in \mathbb{R}^n$, number of clusters k

Initialize: cluster centers $\{c^1, c^2, \ldots, c^k\} \in \mathbb{R}^n$ randomly

Repeat

1. Assign each data point to the nearest cluster center:

$$\pi(i) = \arg\min_{j=1,\ldots,k} \left\| x^i - c^j \right\|^2, \qquad \forall i = 1, \ldots, m$$

2. Update each cluster center as the mean of the assigned points:

$$c^j := \frac{1}{|\{i : \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

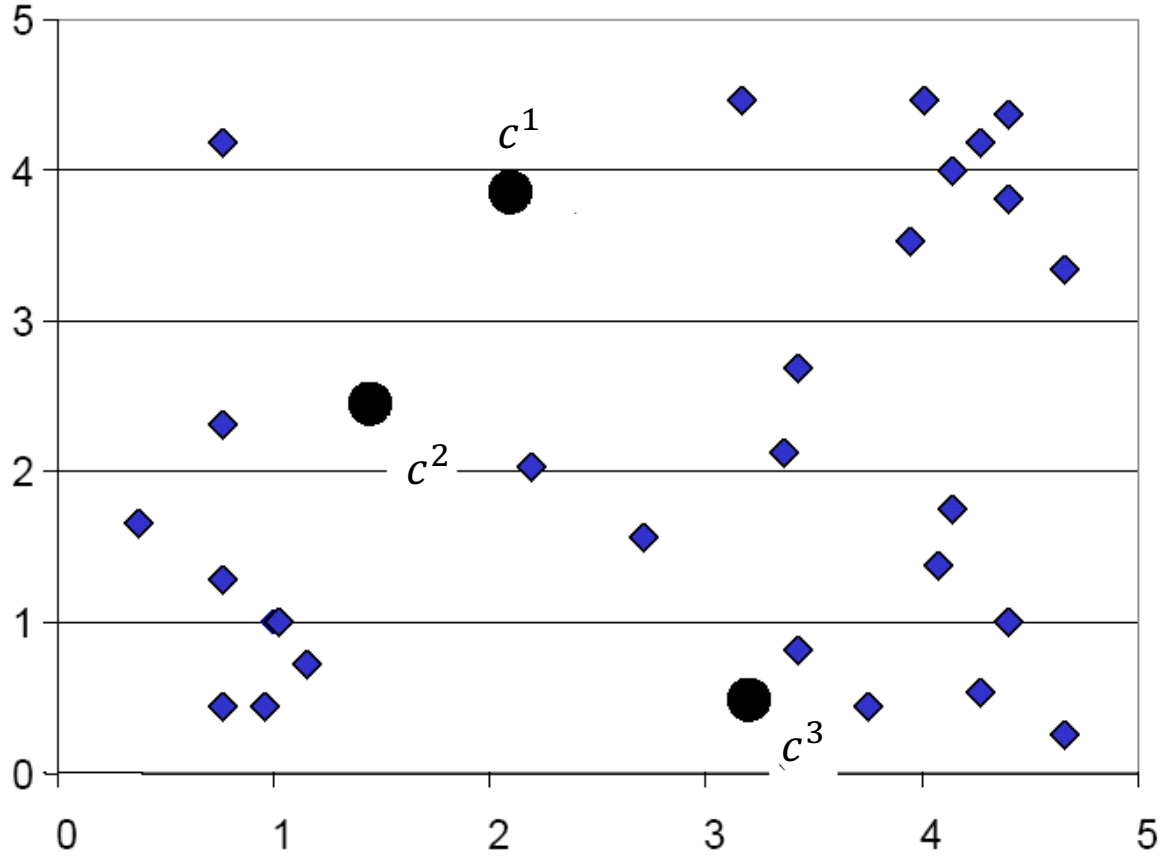
Until: no cluster center (or assignment) changes

Output: final clusters centers $\{c^1, \ldots, c^k\}$ and assignments $\{\pi(i)\}_{i=1}^m$

K-means Algorithm: Initialization

Initialization is critical and can affect convergence and solution quality

- Initialization precedes the assignment–update loop
- Different initialization can lead to different local minima (final clusters)
- Randomness is the source of variability in outcomes

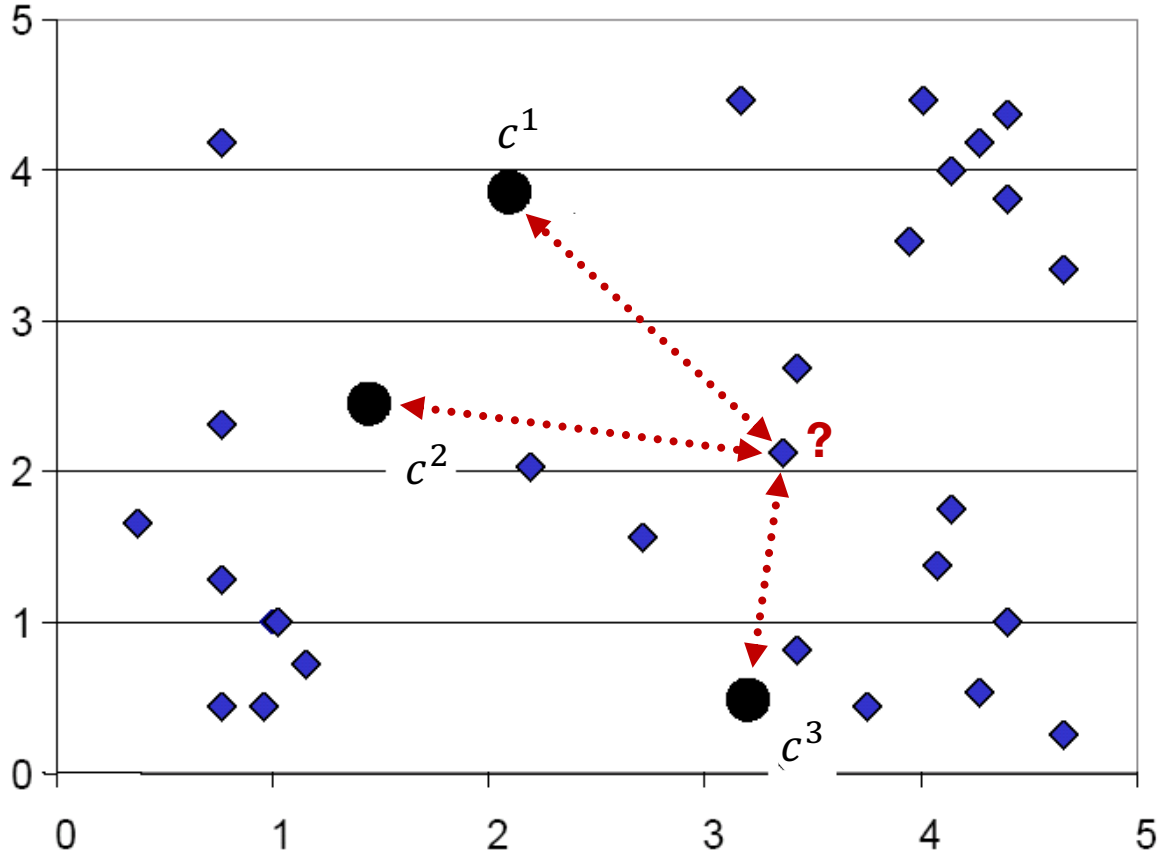


If initialization matters so much, is K-means unreliable?

K-means Algorithm: Assignment

Assign data points to the nearest cluster center based on some distance function

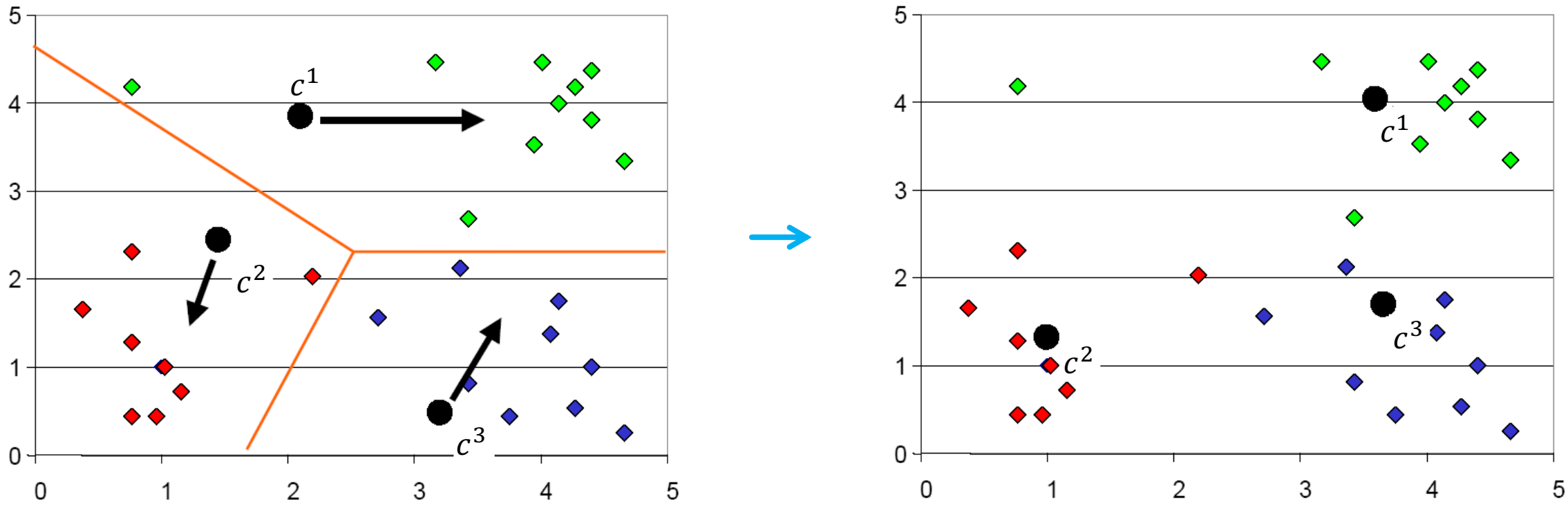
- Assignments minimize distance given fixed centers
- Distance function plays a key role in assignment
- **Geometric intuition:** Space is partitioned into Voronoi cells



What happens to assignments if two centers are very close?

K-means Algorithm: Update

New cluster centers are computed as the mean of assigned points

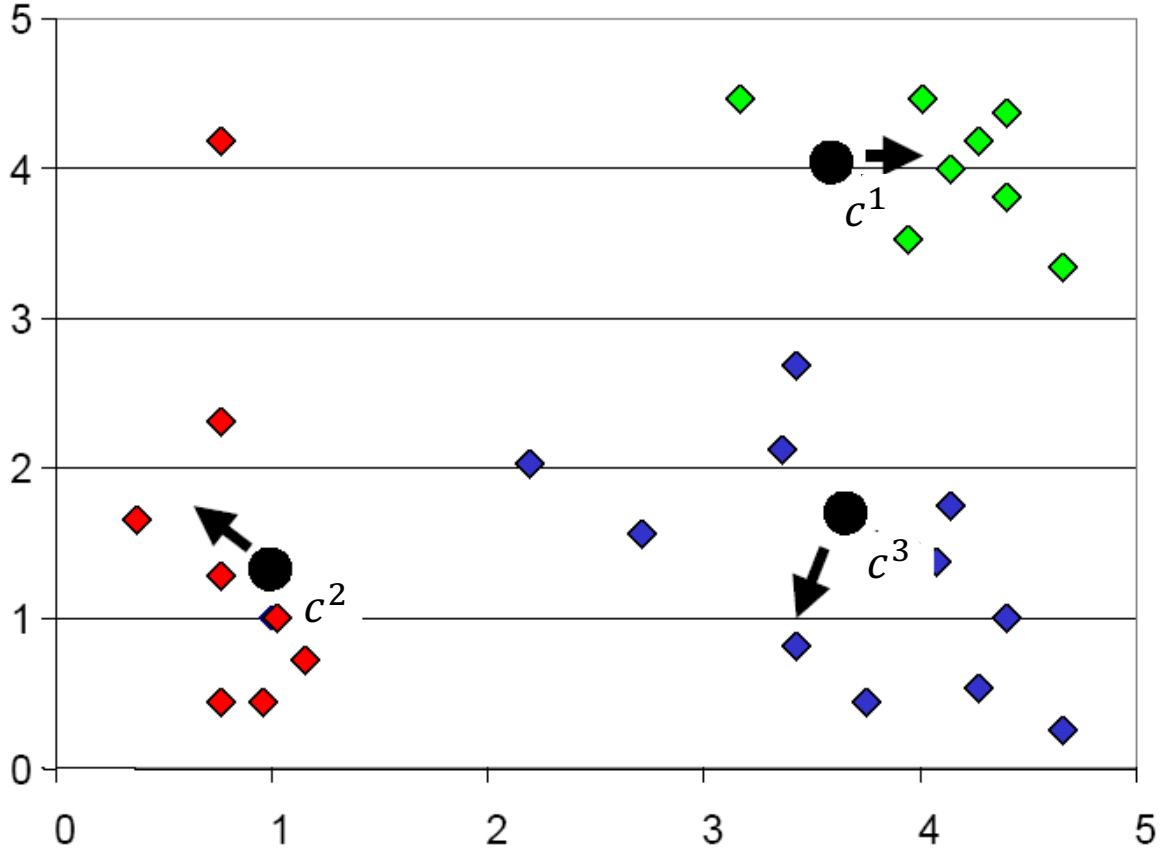


Move each center to the "middle" (mean) of its assigned points

K-means Algorithm: Repeat

Repeat the previous two steps: assignment → cluster center update

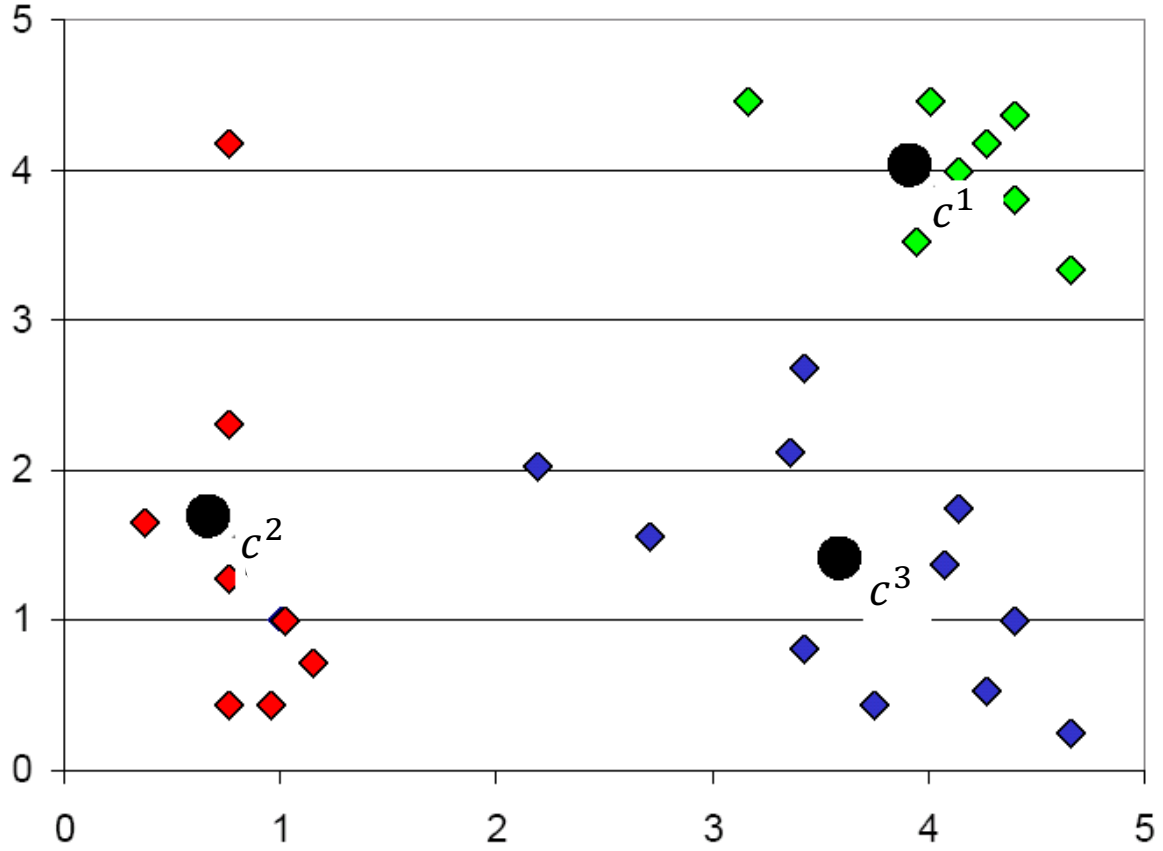
- Assignment and update steps are repeated until stabilization
- **Monotonic improvement:** the objective never increases
- Convergence does not imply optimality



Could this loop run forever?

K-means Algorithm: Stop

The algorithm converges to a **local minimum** due to discrete assignments and continuous center updates



How could we improve our chances of finding a better minimum?

K-means Objective: Formal Statement

Given m data points $\{x^1, x^2, \ldots, x^m\} \in \mathbb{R}^n$

- Find k clusters $\{c^1, c^2, \ldots, c^k\} \in \mathbb{R}^n$
- Assign each data point i to one cluster

$$\pi(i) \in \{1, \ldots, k\}$$

Such that the **average squared Euclidean distance** from each data point x^i to its cluster center $c^{\pi(i)}$ is small

$$\min_{\{c^j\}_{j=1}^k, \{\pi(i)\}_{i=1}^m} \frac{1}{m} \sum_{i=1}^{m} \left\| x^i - c^{\pi(i)} \right\|^2$$

Nonconvex due to discrete assignments $\pi(i)$

What happens when we solve the objective with fixed $\pi(i)$ or fixed c^j?

Clustering is NP-hard

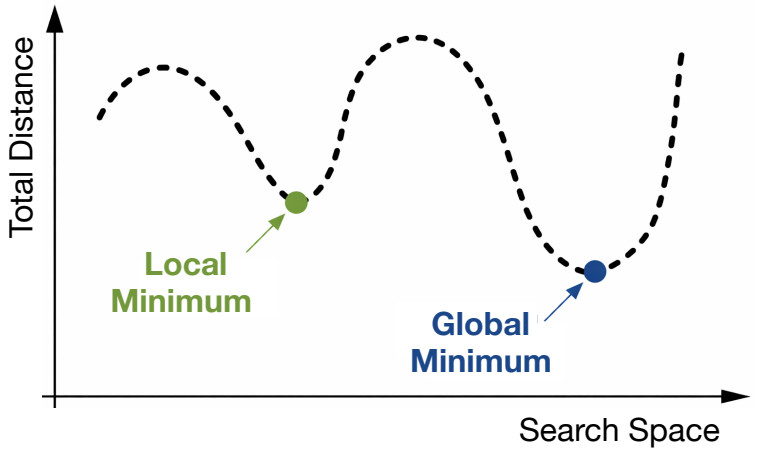
Find k clusters $\{c^1, c^2, \ldots, c^k\} \in \mathbb{R}^n$ and assign each data point i to one cluster $\pi(i) \in \{1, \ldots, k\}$ to minimize

$$\min_{\{c^j\}_{j=1}^k, \{\pi(i)\}_{i=1}^m} \frac{1}{m} \sum_{i=1}^{m} \left\| x^i - c^{\pi(i)} \right\|^2$$

NP-hard

A search over the space of discrete assignments

- Combinatorial explosion—for all m data points, there are $\boldsymbol{k^m}$ **possibilities**
- The cluster assignment determines cluster centers, and vice versa



NP-hardness rules out global optimization, not finding good local solutions

Convergence of K-means

Will the objective oscillate?

$$\min_{\{c^j\}_{j=1}^k, \{\pi(i)\}_{i=1}^m} \frac{1}{m} \sum_{i=1}^{m} \left\| x^i - c^{\pi(i)} \right\|^2$$

The minimum value of the objective is finite, and each iteration of the K-means algorithm **decreases the objective**:

- Cluster assignment (fixed centers): $\pi(i) = \arg\min_{j=1,\ldots,k} \left\| x^i - c^j \right\|^2$
- Center assignment (fixed assignments): $c^j = \frac{1}{|\{i : \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$

Convergence is guaranteed in finite steps, but not to the global optimum

Beyond Standard K-means

Vanilla K-means Assumptions

Assumption 1: Euclidean distance is meaningful

$$d\left(x^i, c^{\pi(i)}\right) = \left\|x^i - c^{\pi(i)}\right\|^2$$

→ What if features aren't numeric or Euclidean distance ≠ semantic similarity?

Assumption 2: Clusters are spherical and isotropic

→ What if clusters are elongated, skewed, or lie on manifolds?

Assumption 3: The mean is a valid representative

→ What if the data is binary, sparse, categorical, or directional?

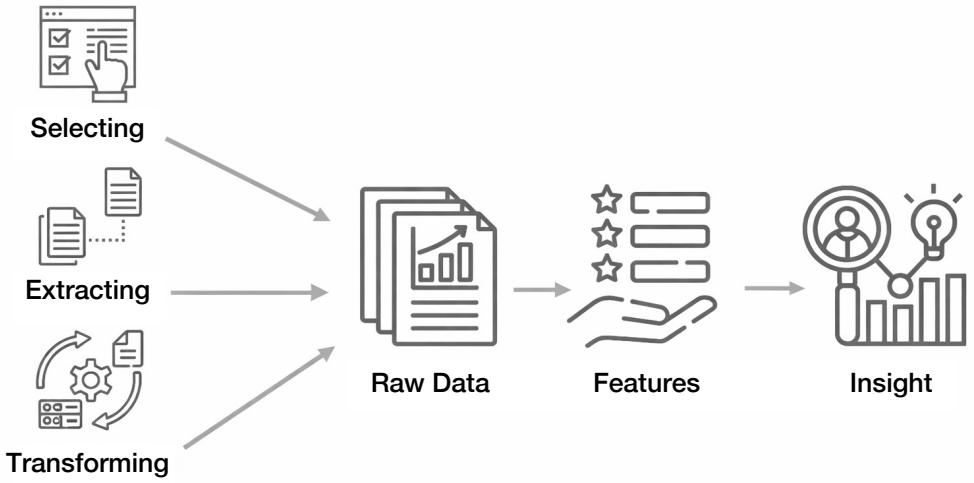
What changes when we change the distance function?

Features

A feature is a **measurable property** or **characteristic** of the data that is used as input to a model—features are not given, but designed

Examples of features

- Numerical (age, salary, temperature)
- Categorical (gender, country, product type)
- Text-based (word counts, TF-IDF scores)
- Image-based (pixel intensity, edges, colors)
- Time-series (moving averages, trends, seasonality)



Implications for Clustering

- Clustering pipelines consist of **features** + **distance** + **prototype update**
- Clustering behavior is largely determined by how we represent data and measure similarity
- In K-means, **features and distance choices** often matter more than the algorithm itself

Distance (Similarity) Functions

A mathematical measure that quantifies **dissimilarity between two data points**

Desired properties:

- **Symmetry:** The distance between two points x and y is the same in both directions:

$$d(x, y) = d(y, x)$$

- **Positive separability:** The distance between two points is non-negative and is zero only when the points are identical:

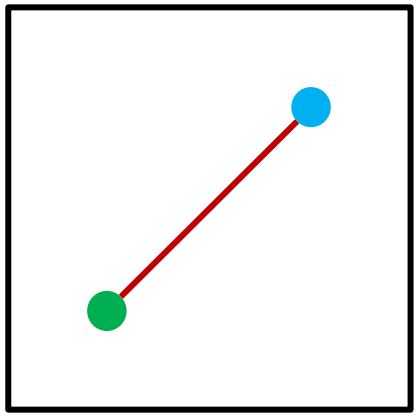
$$d(x, y) = 0 \quad \text{iff} \quad x = y$$

- **Triangular inequality:** The direct distance between two points x and y is no greater than going through a third point z

$$d(x, y) \leq d(x, z) + d(y, z)$$

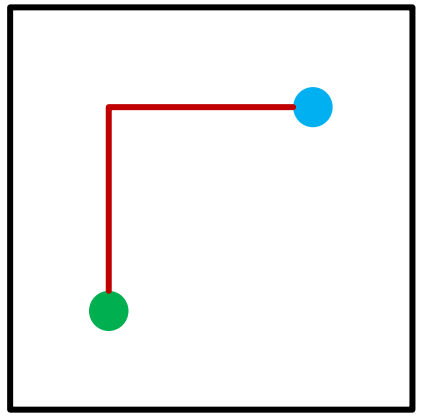
Distance Functions for Vectors

Suppose two data points $x = (x_1, x_2, \ldots, x_n)^\top$ and $x = (y_1, y_2, \ldots, y_n)^\top$ in $\mathbb{R}^n$



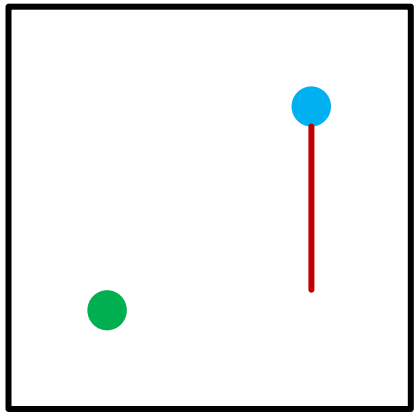
Euclidean Distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



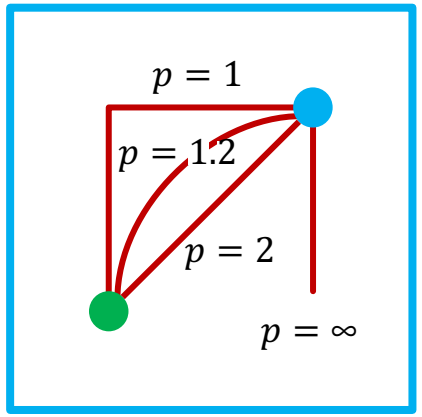
Manhattan Distance

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$



Chebyshev Distance

$$d(x, y) = \max_i |x_i - y_i|$$

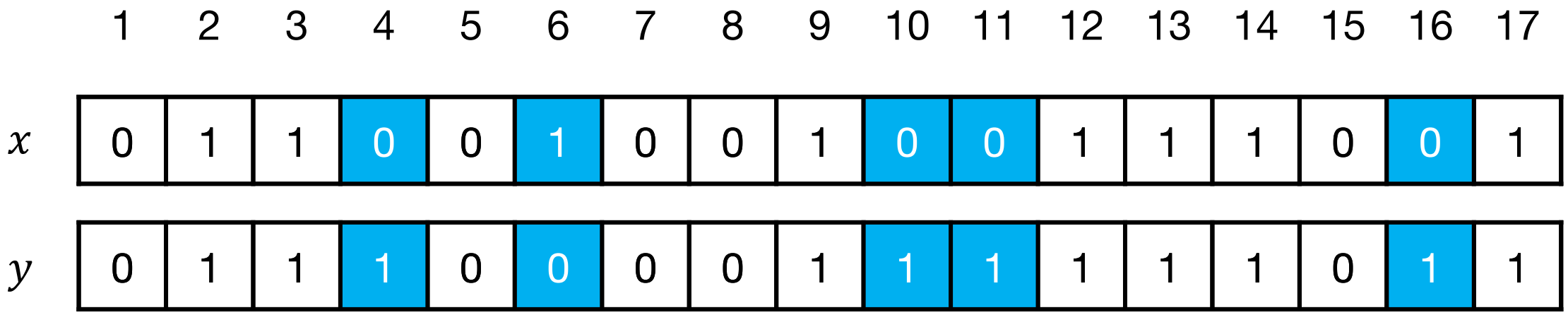


Minkowski p-Norm Distance

$$d(x, y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}, \quad p \geq 1$$

Hamming Distance

Manhattan distance when all features are binary (i.e., magnitude is meaningless)—**counting the number of mismatches** between two binary vectors

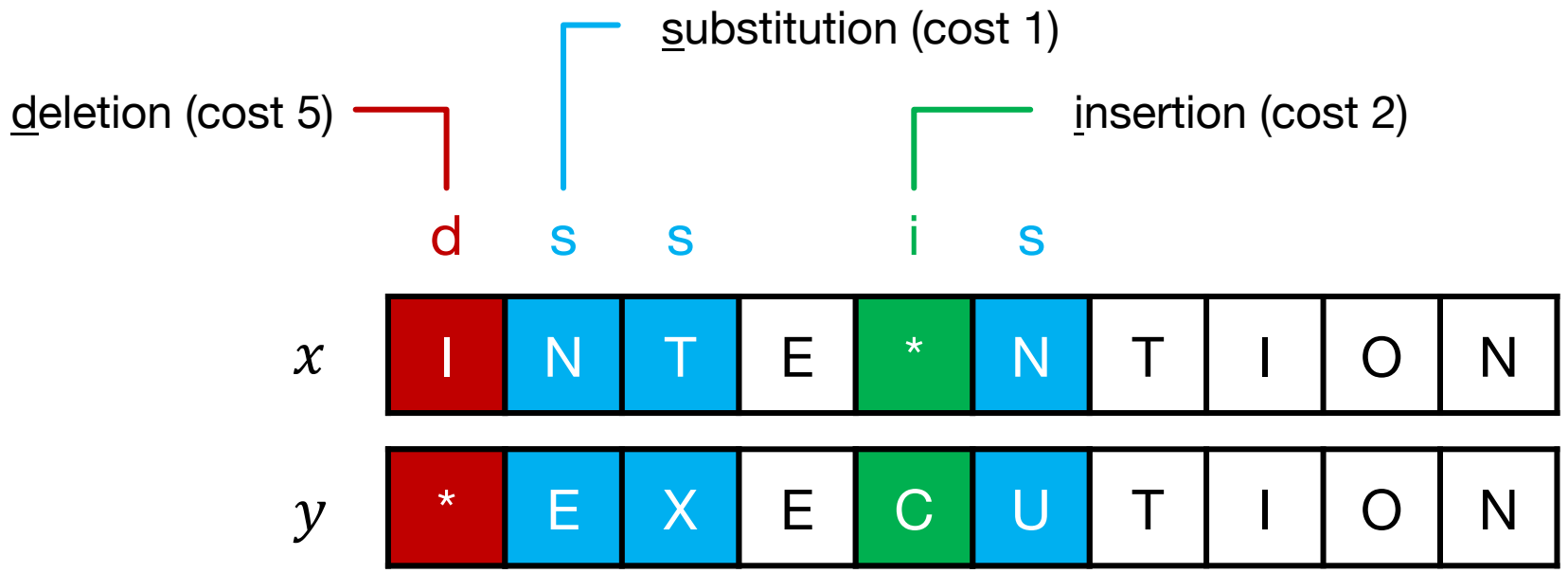


$$d(x, y) = 5$$

Why would Euclidean distance be misleading for one-hot encoded data?

Edit Distance

Transform one of the objects into the other and measure **how much effort** it takes



$$d(x, y) = 1{\times}5 + 3{\times}1 + 1{\times}2 = 10$$

Commonly used in string and sequence data (e.g., DNA, spelling correction)

Generalization of K-means

Given m data points $\{x^1, x^2, \ldots, x^m\} \in \mathbb{R}^n$

- Find k clusters $\{c^1, c^2, \ldots, c^k\} \in \mathbb{R}^n$
- Assign each data point i to one cluster

$$\pi(i) \in \{1, \ldots, k\}$$

Such that the **average general distance** from each data point x^i to its cluster center $c^{\pi(i)}$ is small

$$\min_{\{c^j\}_{j=1}^k, \{\pi(i)\}_{i=1}^m} \frac{1}{m} \sum_{i=1}^{m} d\left(x^i, c^{\pi(i)}\right)$$

General distance function d

Only need a notion of "closeness," not necessarily Euclidean geometry

Generalized K-means Algorithm

Input: data $\{x^i\}_{i=1}^m$, $x^i \in \mathbb{R}^n$, number of clusters k

Initialize: cluster centers $\{c^1, c^2, \ldots, c^k\} \in \mathbb{R}^n$ randomly

Repeat

1. Assign each data point to the nearest cluster center:

$$\pi(i) = \arg\min_{j=1,\ldots,k} d\left(x^i, c^{\pi(i)}\right), \qquad \forall i = 1, \ldots, m$$

2. Update each cluster center s.t. **within-cluster distortion** is minimized:

$$c^j := \arg\min_{v \in \mathbb{R}^n} \sum_{i:\pi(i)=j} d\left(x^i, v\right)$$

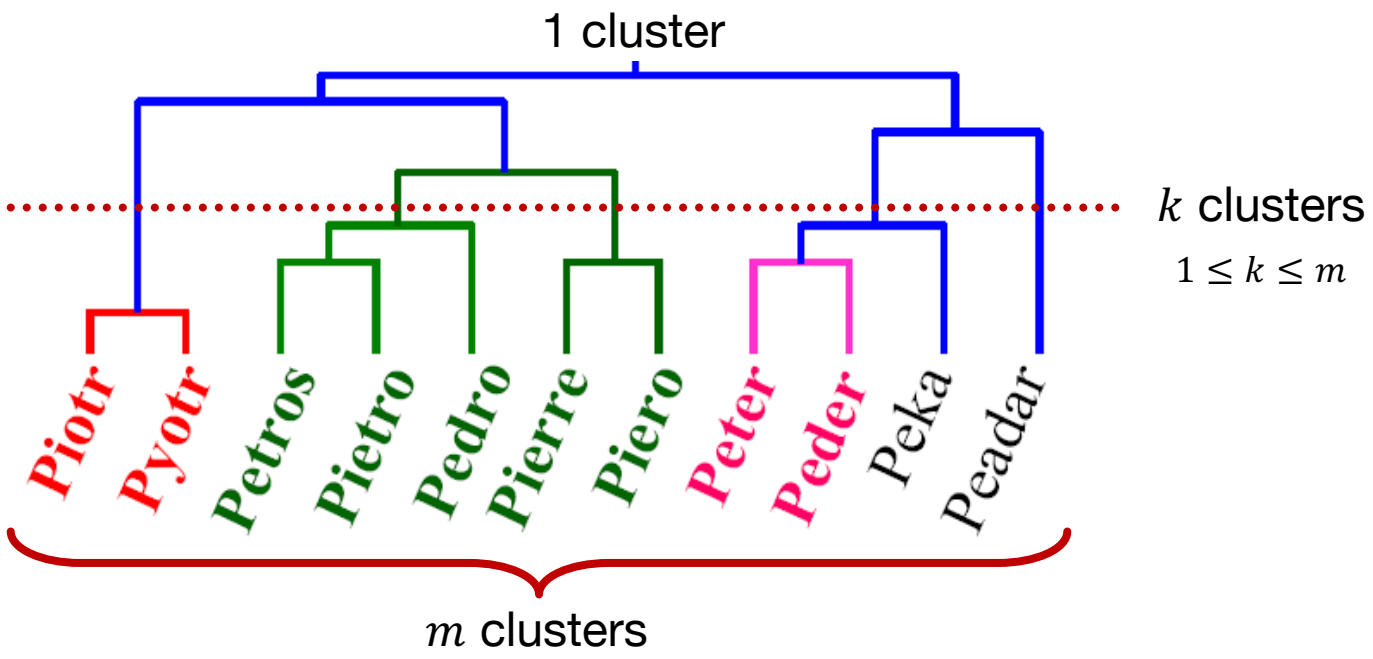
Until: no cluster center (or assignment) changes

Output: final clusters centers $\{c^1, \ldots, c^k\}$ and assignments $\{\pi(i)\}_{i=1}^m$

Hierarchical Clustering

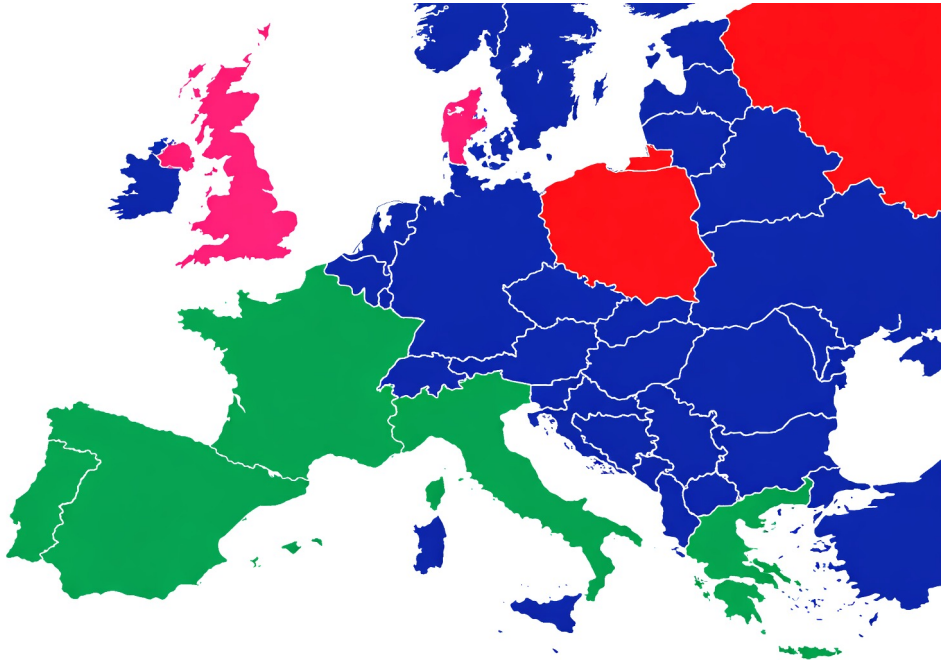
What is Hierarchical Clustering?

Organizes data through a hierarchical "tree"—clusters are obtained by cutting the **dendrogram** at a desired level: each connected component forms a cluster



k clusters

$1 \leq k \leq m$



What does the height at which two branches merge represent?

Bottom-Up Hierarchical Clustering

Input: data $\{x^i\}_{i=1}^m$, $x^i \in \mathbb{R}^n$

Initialize: assign each data point to its own cluster $g_i = \{x^i\}$, $\forall i$, and let the set of clusters be $G = \{g_1, g_2, \ldots, g_m\}$

Repeat

1. Find the closest pair of clusters:

$$(i,j) = \arg \min_{1 \le i,j \le |G|,\, i \ne j} D(g_i, g_j), \qquad \forall i,j$$

2. Merge the selected clusters: $g \coloneqq g_i \cup g_j$
3. Update the cluster set: $G \coloneqq G \setminus \{g_i, g_j\}$, $G \coloneqq G \cup \{g\}$

Until: $|G| = 1$

Output: a hierarchy of clusters and a dendrogram

$$D_{\text{single}}(g_i, g_j) = \min_{x \in g_i, y \in g_j} d(x,y)$$

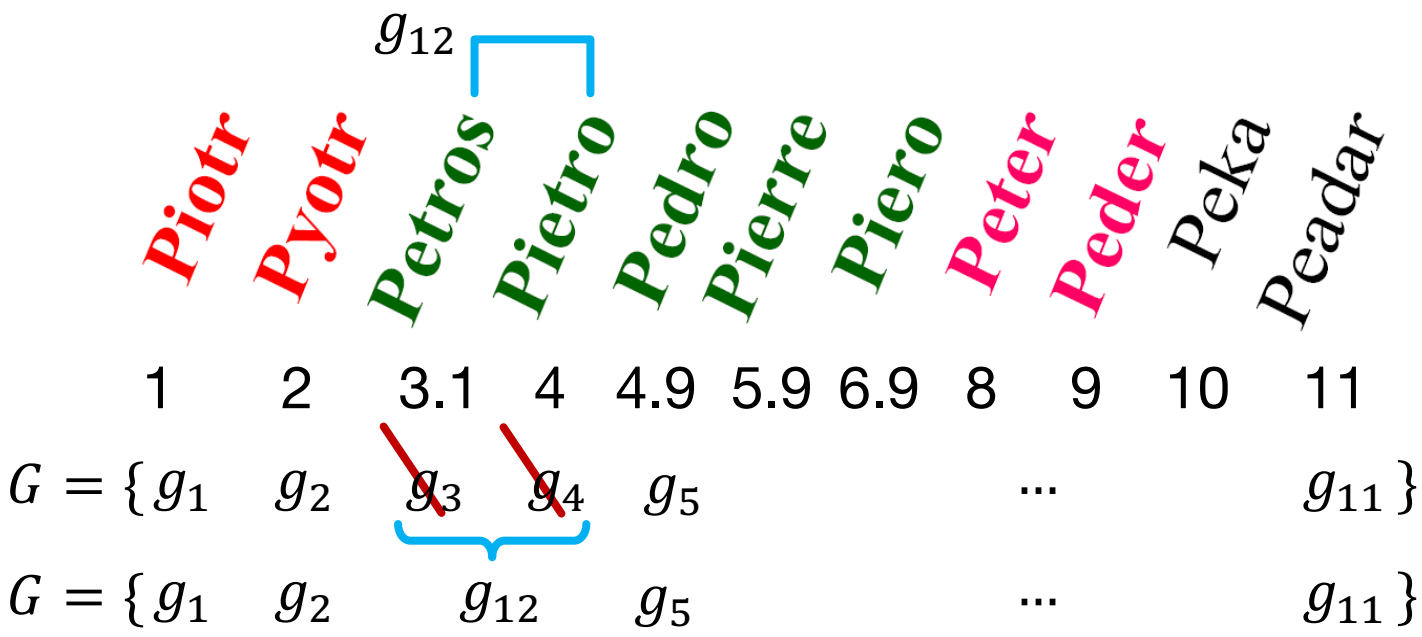
$$D_{\text{complete}}(g_i, g_j) = \max_{x \in g_i, y \in g_j} d(x,y)$$

$$D_{\text{avg}}(g_i, g_j) = \frac{1}{|g_i||g_j|} \sum_{x \in g_i} \sum_{y \in g_j} d(x,y)$$

Hierarchical Clustering Step 2

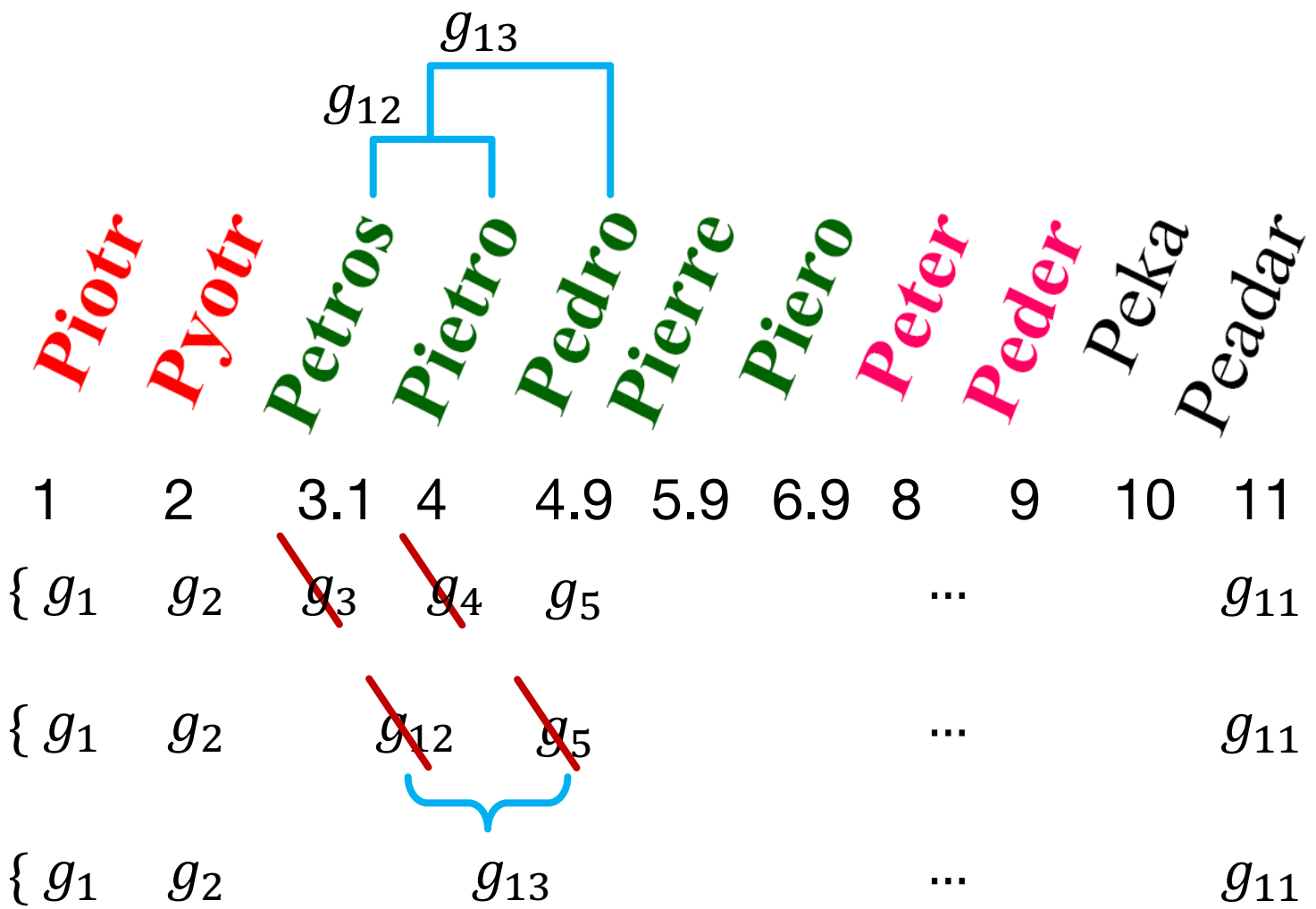
Linkage criteria determine cluster merging, measuring how close two clusters are

- **Single:** If any one pair of points—one from each cluster—is close
- **Complete:** Only if all points in one cluster are close to all points in the other
- **Average:** If the average distance between all cross-cluster point pairs is small



Hierarchical Clustering Step 3

Cutting the dendrogram yields different clusters at different resolutions



Key Takeaways

What We Learned This Week

- Clustering is an unsupervised learning task that discovers structure without labels, and results depend strongly on representation and similarity choices
- Data must be represented as feature vectors; scaling and distance metrics directly shape cluster geometry and outcomes
- K-means alternates between assignment and center updates, converges to a local optimum, and works best for compact, roughly spherical clusters
- Generalized K-means extends the framework by changing distance functions and cluster prototypes (e.g., mean vs. median)
- Hierarchical clustering builds a multiscale view of data via dendrograms and linkage rules, allowing flexible choices of the number of clusters