Distance & Similarity

d is a distance function if and only if:

- d(i, j) = 0 if and only if i = j
- d(i, j) = d(j, i)
- d(i, j) ≤ d(i, k) + d(k, j)

We don't need a distance function to compare data points

- In order to uncover interesting structure from our data, we need a way to compare data points.
- A dissimilarity function is a function that takes two objects (data points) and returns a large value if these objects are dissimilar.
- A special type of dissimilarity function is a distance function

<u>Minkowski Difference</u>

For x, y points in d-dimensional real space
I.e. x = [x1 , … , xd] and y = [y1 , … , yd]

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

p ≥ 1
When p = 2 -> Euclidean Distance
When p = 1 -> Manhattan Distance

<u>Cosine Similarity</u>

A similarity function is a function that takes two objects (data points) and returns a large value if these objects are similar.

s(x, y) = cos(θ)

where θ is the angle between x and y

To get a corresponding dissimilarity function, we can usually try

d(x, y) = 1 / s(x, y)

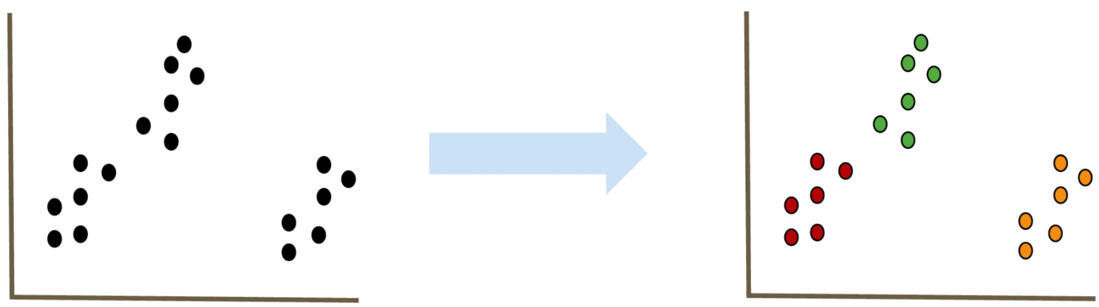
or

d(x, y) = k - s(x, y) for some k

- use cosine (dis)similarity over euclidean distance when direction matters more than magnitude

Clustering

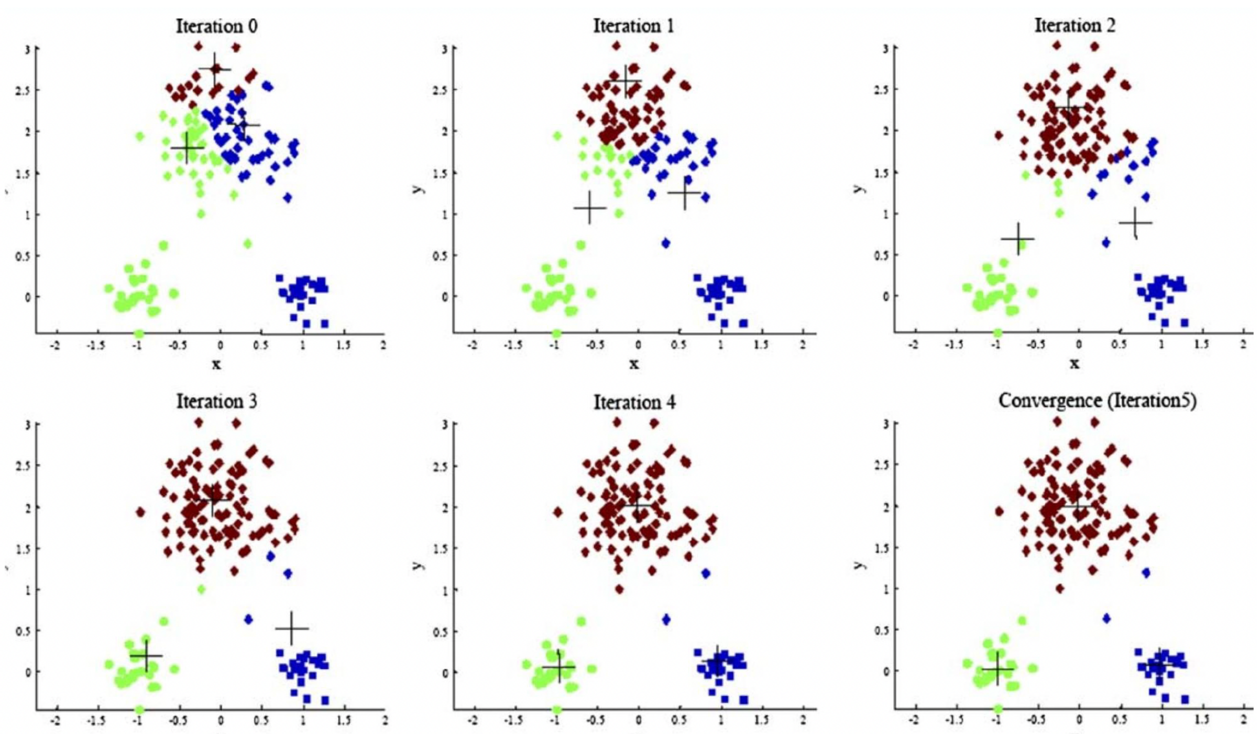
A clustering is a grouping / assignment of objects (data points) such that objects in the same group / cluster are:
- similar to one another
- dissimilar to objects in other groups



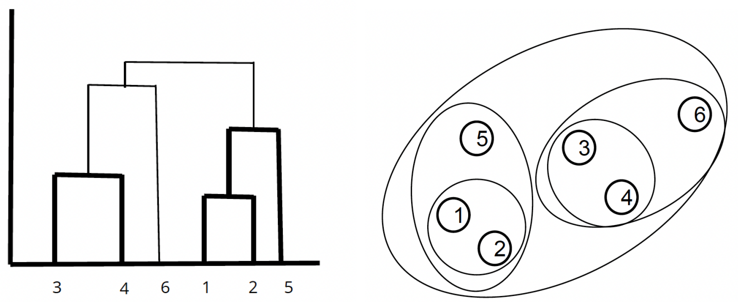
- can be ambiguous

Types:

- Partional
    - Goal: partition dataset into k partitions
    - Each object belongs to exactly one cluster
    - Eg K-means



- Hierarchical
    - A set of nested clusters organized in a tree
    - At every step, we record which clusters were merged in order to produce a Dendrogram:



    - 2 types:
        - Agglomerative
        - Divisive
- Density-Based
    - Defined based on the local density of points
    - **ε** and min_pts given:
    - 1. Find the **ε**-neighborhood of each point
    - 2. Label the point as core if it contains at least min_pts
    - 3. For each core point, assign to the same cluster all core points in its
    - neighborhood (crux of the algorithm)
    - 4. Label points in its neighborhood that are not core as border
    - 5. Label points as noise if they are neither core nor border
    - 6. Assign border points to nearby clusters
- Soft Clustering
    - Each point is assigned to every cluster with a certain probability

Clustering Aggregation

- Clustering: A group of clusters output by a clustering algorithm
- Cluster: A group of points

Goals:
1. Compare clusterings
2. Combine the information from multiple clusterings to create a new clustering

Same clustering, different assignments/labels

we cannot know this conversion upfront unless there is a known set of conventions

- A good question to determine the conventions: “Do P and C agree or disagree oN whether x and y should be clustered together?”

<u>Disagreement Distance</u>

Given 2 clusterings P and C:

$$D(P, C) = \sum_{x,y} \mathbb{I}_{P,C}(x, y)$$

Where

$$\mathbb{I}_{P,C}(x, y) = \begin{cases} 1 & \text{if P \& C disagree on which clusters x \& y belong to} \\ 0 & \end{cases}$$

	P	**C**
$\mathbf{x_1}$	1	1
$\mathbf{x_2}$	1	2
$\mathbf{x_3}$	2	1
$\mathbf{x_4}$	3	3
$\mathbf{x_5}$	3	4

What’s the disagreement distance between P and C?

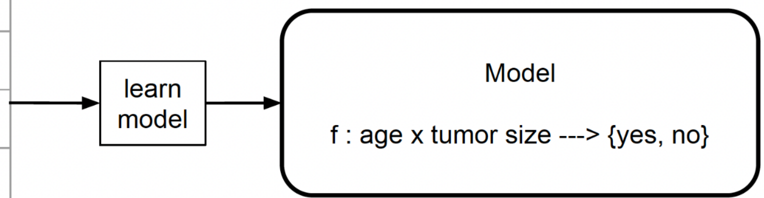
x_2	x_1	1
x_3	x_1	1
x_4	x_1	0
x_5	x_1	0
x_3	x_2	0
x_4	x_2	0
x_5	x_2	0
x_4	x_3	0
x_5	x_3	0
x_4	x_5	1

Benefits:
1. Can identify the best number of clusters (optimization function does not make any assumptions on the number of clusters)
2. Can handle / detect outliers (points where there is no consensus)
3. Improve robustness of the clustering algorithms - combining clusterings can produce a better result
4. Privacy preserving clustering (can compute aggregate clustering without sharing the data, need only share the assignments)

Classification

● Given a training set where data is labeled with a special attribute called a class (a discrete value)
● We want to find a model describing the class attribute as a function of the values of the other attributes
● Goal: use this model on unlabeled data to assign a class as accurately as Possible

age	Tumor size	malignant?
20	10	no
30	15	yes
40	20	no
50	25	yes



Tasks:
- Predicting tumor cells as benign or malignant
- Classifying images
- Classifying credit card transactions as being legitimate or fraudulent

Techniques:
- Instance-Based Classifiers
- Decision Trees
- Naive Bayes
- Support Vector Machines
- Neural Networks

<u>K Nearest Neighbor Classifier</u>
Requires:
- Training set
- Distance function
- Value for k

How to classify an unseen record:
1. Compute distance of unseen record to all training records
2. Identify the k nearest neighbors
3. Aggregate the labels of these k neighbors to predict the unseen record class (ex: majority rule)

Pros:
- Simple to understand why a given unseen record was given a particularclass
- Adapts to new attributes

Cons:
- Expensive to classify new points
- KNN can be problematic in high dimensions (curse of dimensionality)