SME3006 Machine Learning – 2025 Fall

Clustering: K-means and more

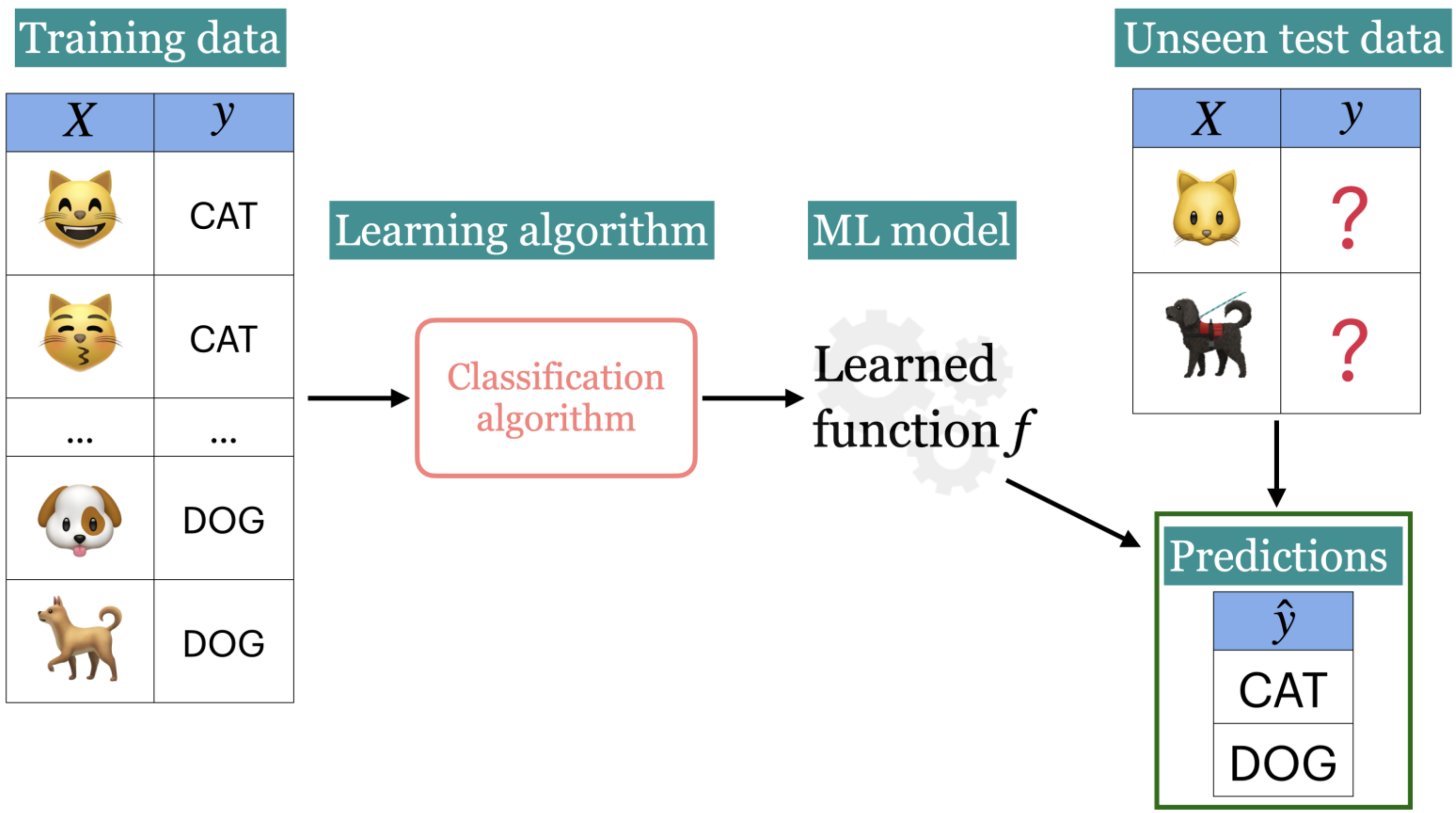
INHA UNIVERSITY

Overview

- Introduction to clustering problem
- How close is the data? distance/similarity
- K-means and extensions
- Other clustering methods
  - DBSCAN
  - Hierarchical Clustering
  - (Spectral Clustering)

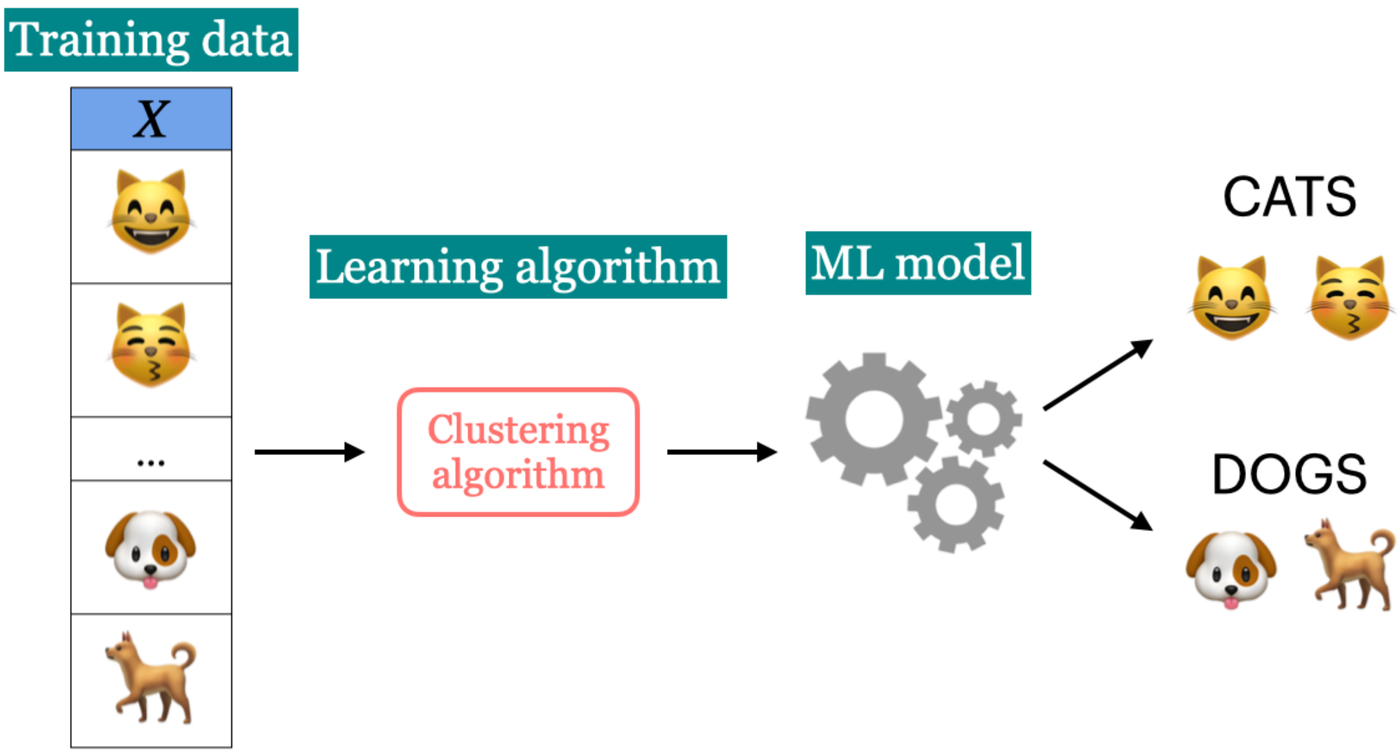
Types of machine learning

- Supervised learning
  - training a model from input data and its corresponding targets



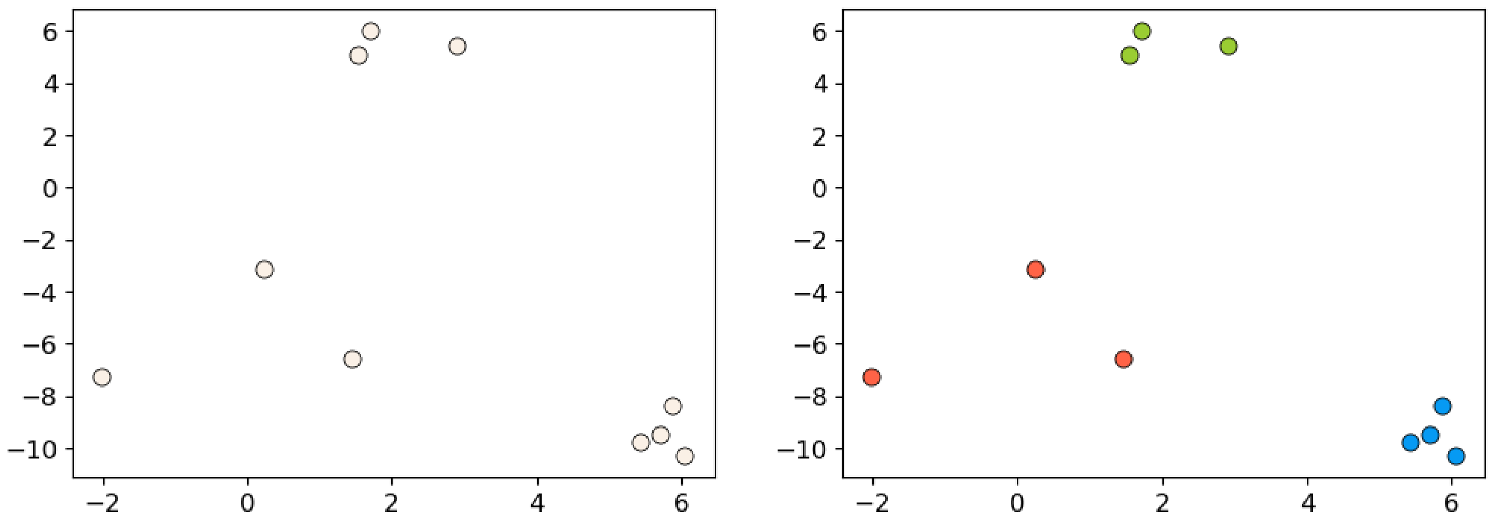
Types of machine learning

> Unsupervised learning
  - only training input data (since labeling is expensive)
  - learning will be focused on finding the underlying structures of the inputs



Clustering

> Group similar examples together to get some insight into the data



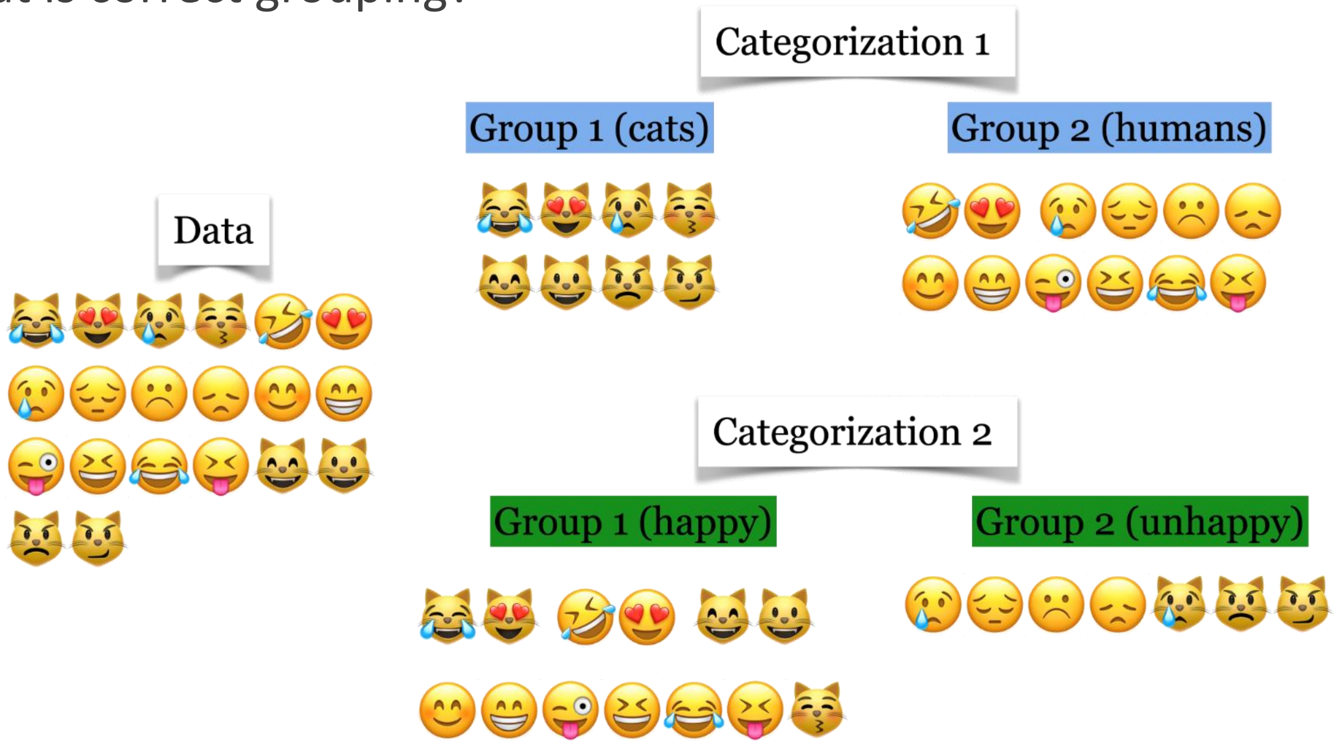
- clusters are identified by a cluster label
- label is only for separating the clusters (knowing the different group)
- In real-world data, we often do not know how many clusters are there

Clustering

> What is correct grouping?

Clustering

> What is correct grouping?



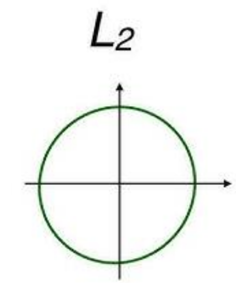
- Both seem reasonable
- This makes it hard to for us to measure the quality of a clustering algorithm

Clustering

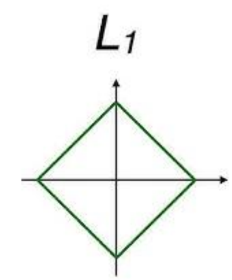
> How can we define the similarity (or closeness) between data points?

- Euclidean distance (L2 norm)

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\textstyle\sum_i (x_i - y_i)^2}$$



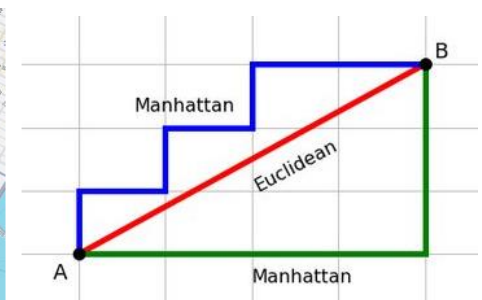
  - measure straight-line distance in continuous feature space
  - assumes all features are equally scaled and uncorrelated



- Manhattan distance (L1 norm)

$$d(\boldsymbol{x}, \boldsymbol{y}) = |\textstyle\sum_i (x_i - y_i)|$$

  - less sensitive to outliers
  - intuitively, we can't go diagonally

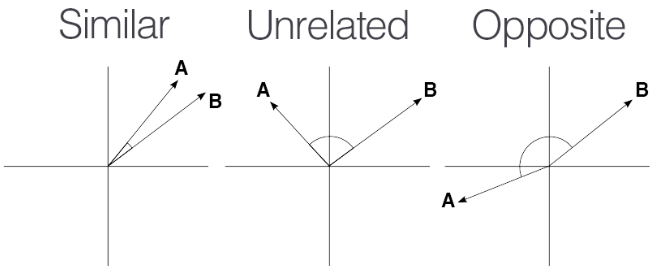


Clustering

> How can we define the similarity (or closeness) between data points?

- Cosine similarity

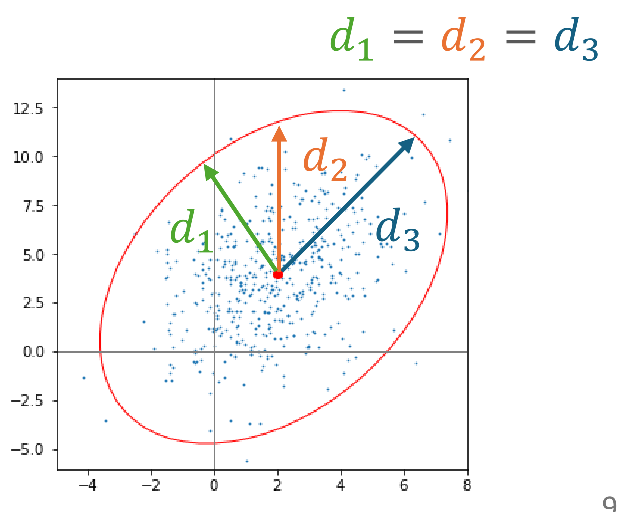
$$sim(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$$



  - measures only the angle between two vectors (ignoring the magnitude)
  - effective in high-dimensional, sparse data

- Mahalanobis distance

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^\top \Sigma (\boldsymbol{x} - \boldsymbol{y})}$$



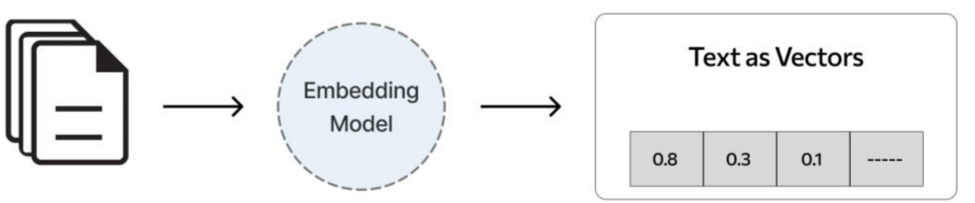
  - accounts for correlations between features using the covariance matrix Σ
  - useful when variables have different scales

Clustering

- How can we define the similarity (or closeness) between data points?
  - Categorical data
    - Hamming distance:
      count the number of mismatch
    - Jaccard similarity:
      measure the size of intersection
      $$sim(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

	Blood Type	Hair Color
student 1	A	Black
student 2	B	Brown
student 3	B	Black
student 4	AB	Red

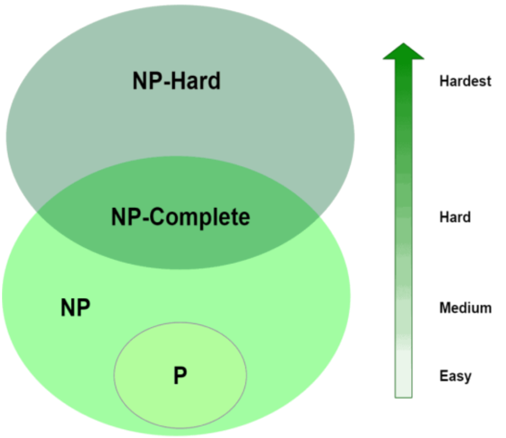
  - In practice, text and image data
    - embed to high-dimensional vector (dimension reduction)
    - compute the geometric distance in the feature space



K-means

> Assumptions
  - the data lives in a Euclidean space
  - the data belongs to K classes
  - the data points from same class are similar (close in Euclidean distance)
  - K-means assumes there are k clusters and each point is close to cluster center

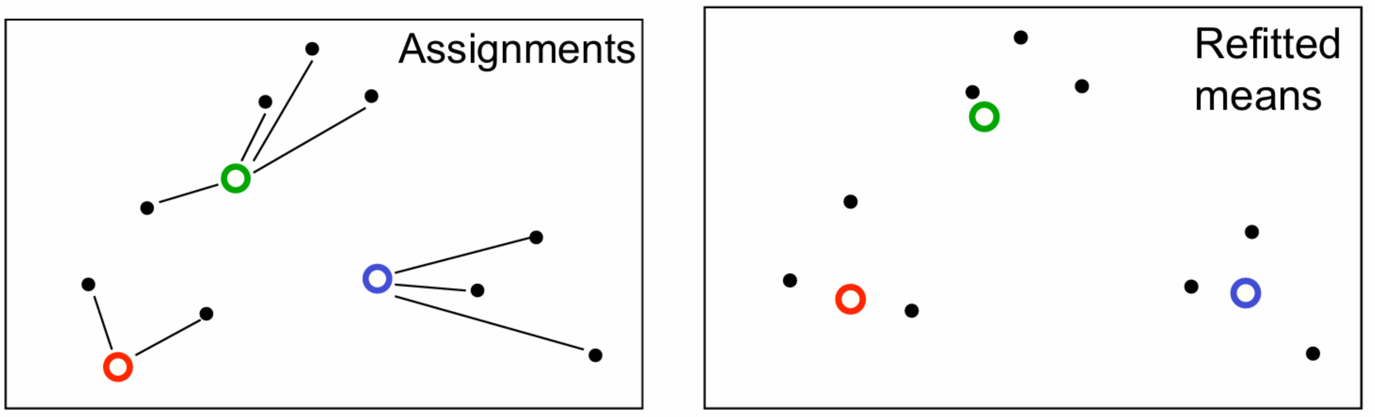
> Chicken and egg problem
  - if we knew the cluster assignment, we could easily compute means
  - if we know the means, we could easily compute cluster assignment
  - it is a NP hard



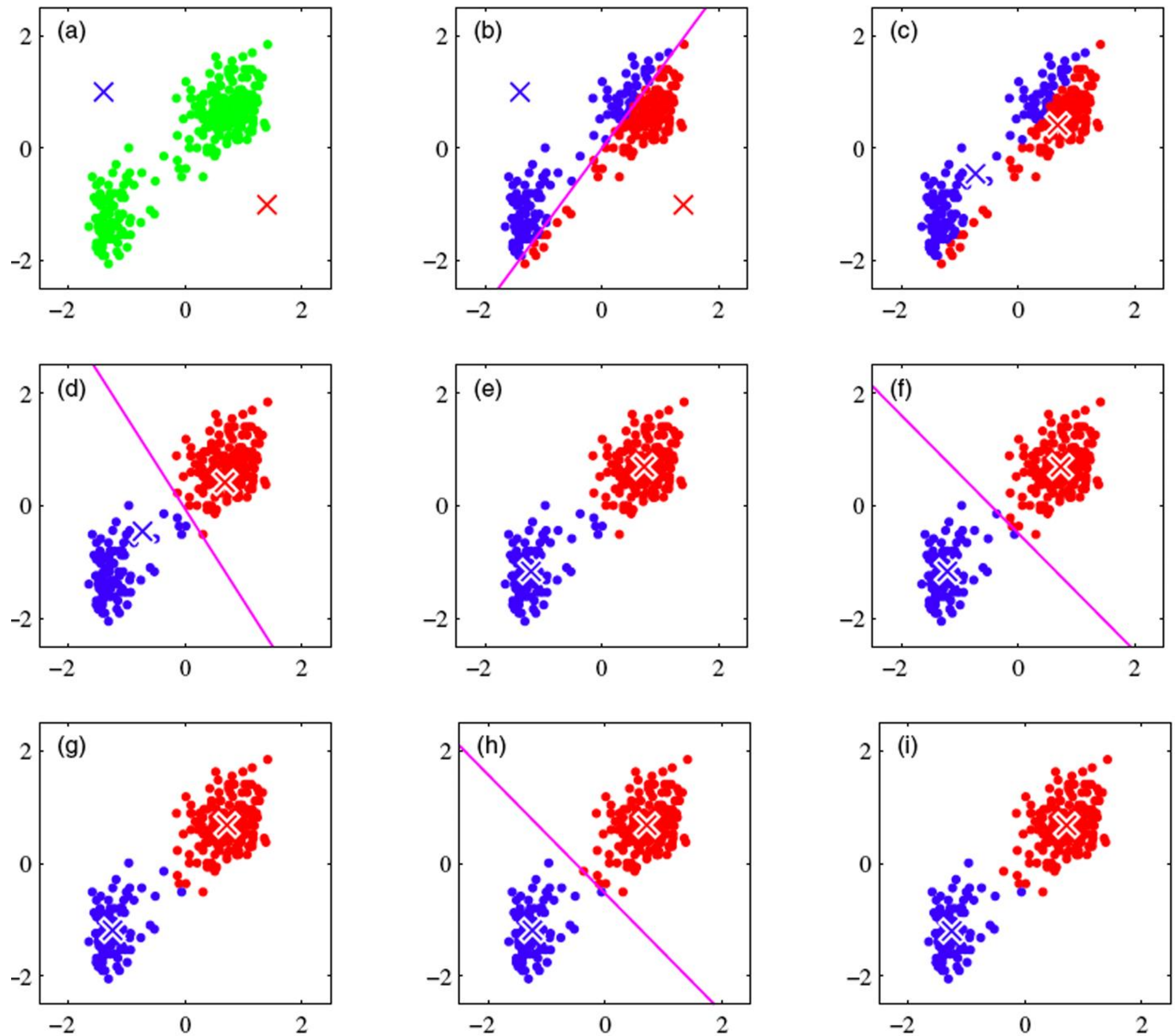
> Simple heuristics: start randomly and alternate between the two

K-means

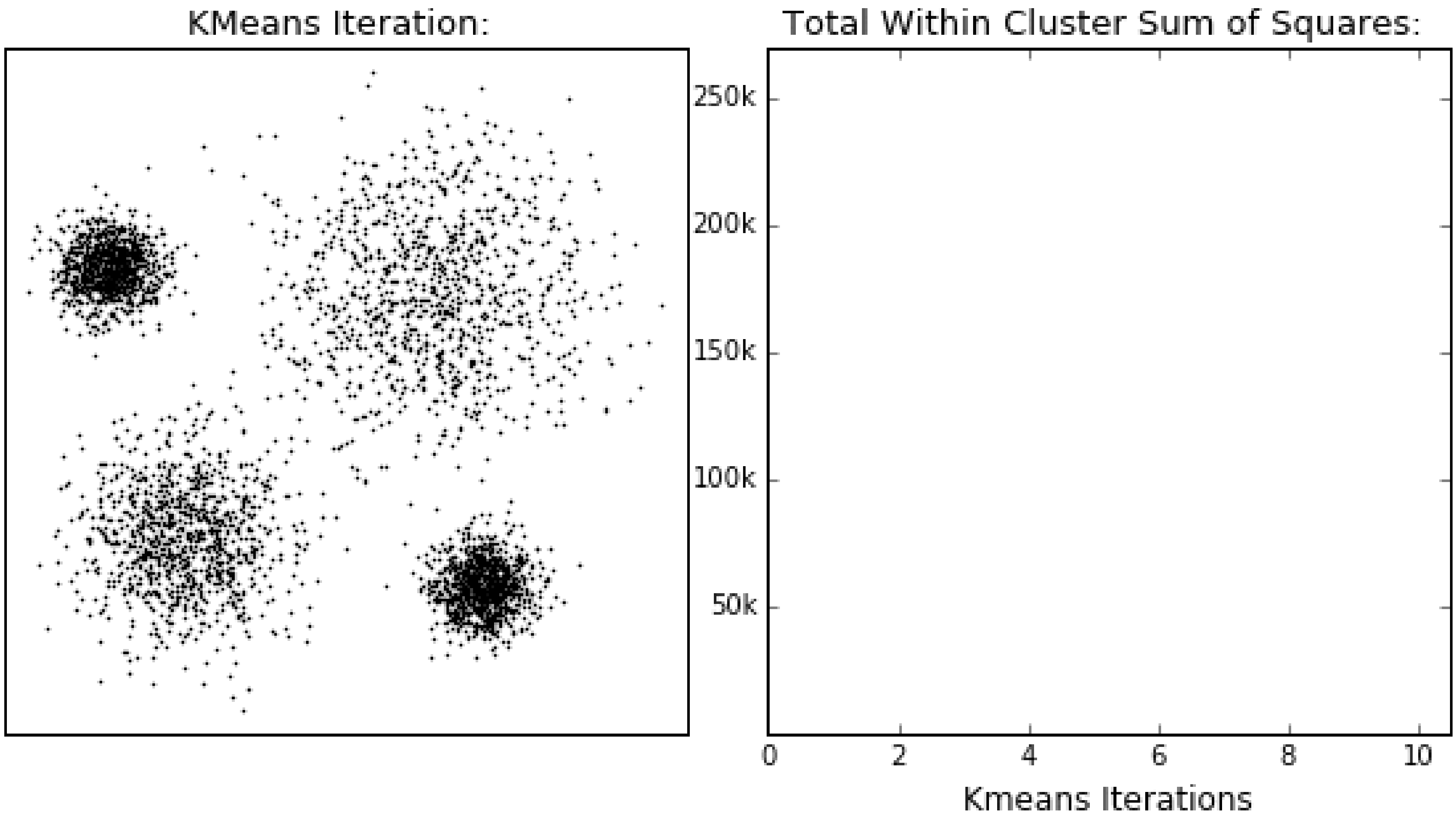
- Initialization: randomly initialize cluster centers
- Iteratively alternates between two steps:
  - assignment step: assign each data point to the closest cluster
  - refitting step: move each cluster center to the center of gravity of the data assigned to it



K-means



K-means

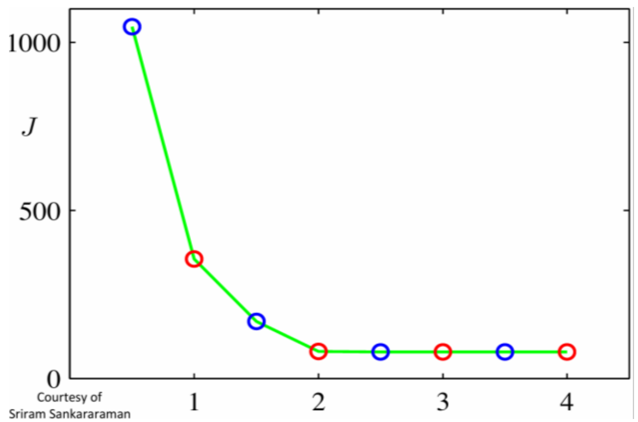

KMeans Iteration:
Total Within Cluster Sum of Squares:
250k
200k
150k
100k
50k
0
2
4
6
8
10
Kmeans Iterations

K-means

> Objective: find cluster center μ_i that minimizes the following cost

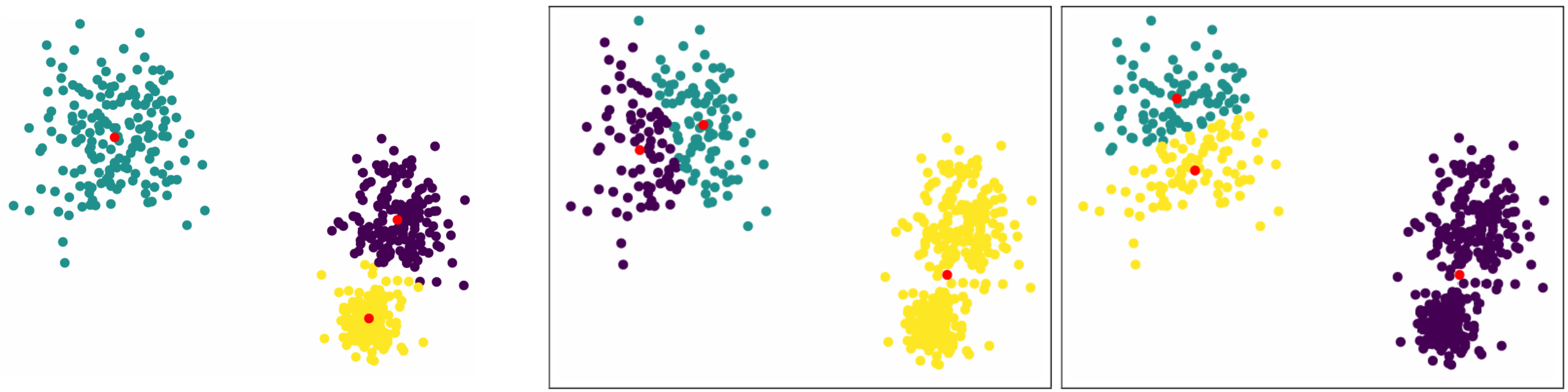
$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$ where C_i is the set of points assigned to cluster i

- assignment step: assign each data point to the closest cluster
- refitting step : move each cluster center to the center of gravity of the data assigned to it
- cost J cannot increase
  - assignment step: for each point, distance decreases
  - refitting step: for fixed cluster, the best centroid is the mean of the points



K-means

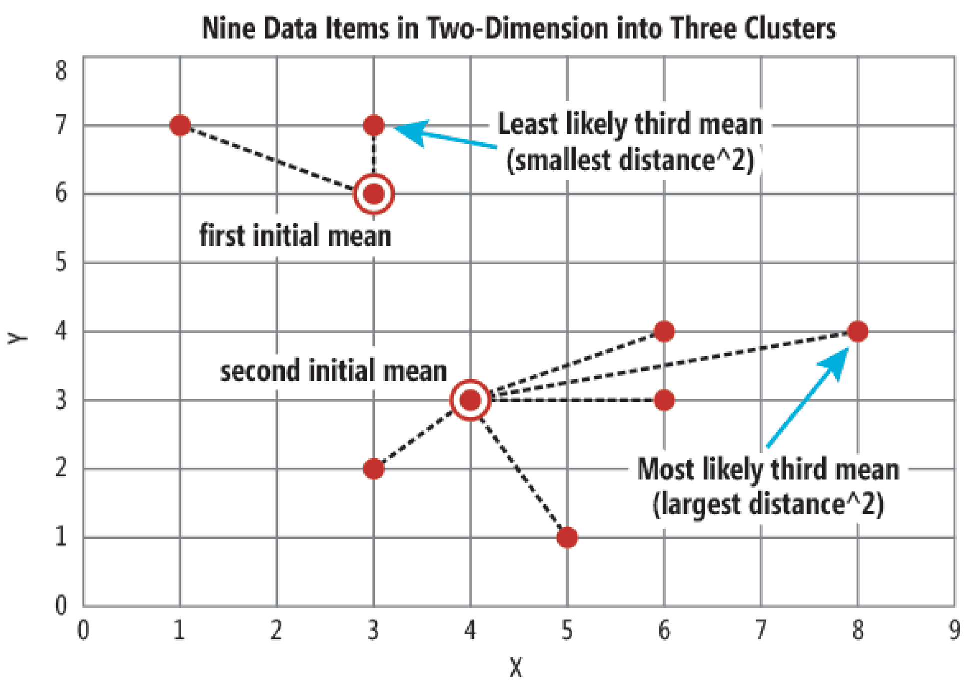
> Poor clustering results
  - it depends on the random initialization (since it a non-convex problem)
  - there is nothing to prevent k-means getting stuck at local minima



  - try many random starting points
  - try non-local split-and-merge moves
    - simultaneously merge two nearby clusters
    - and split a big cluster into two

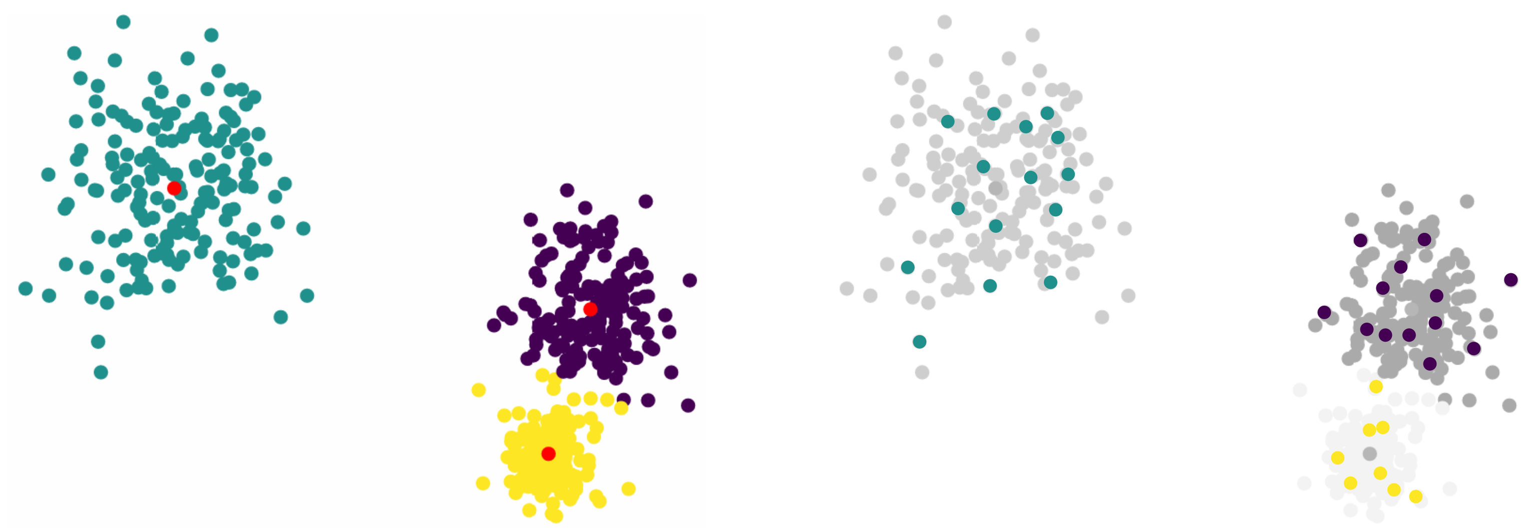
K-means extensions

- K-means++: improves initialization
  - choose the first center uniformly at random from data points
  - For each new center, pick a point with probability proportional to the square of its distance from the nearest chosen center
  - once all centers are selected, run K-means
- In practice: more accurate and faster than k-means



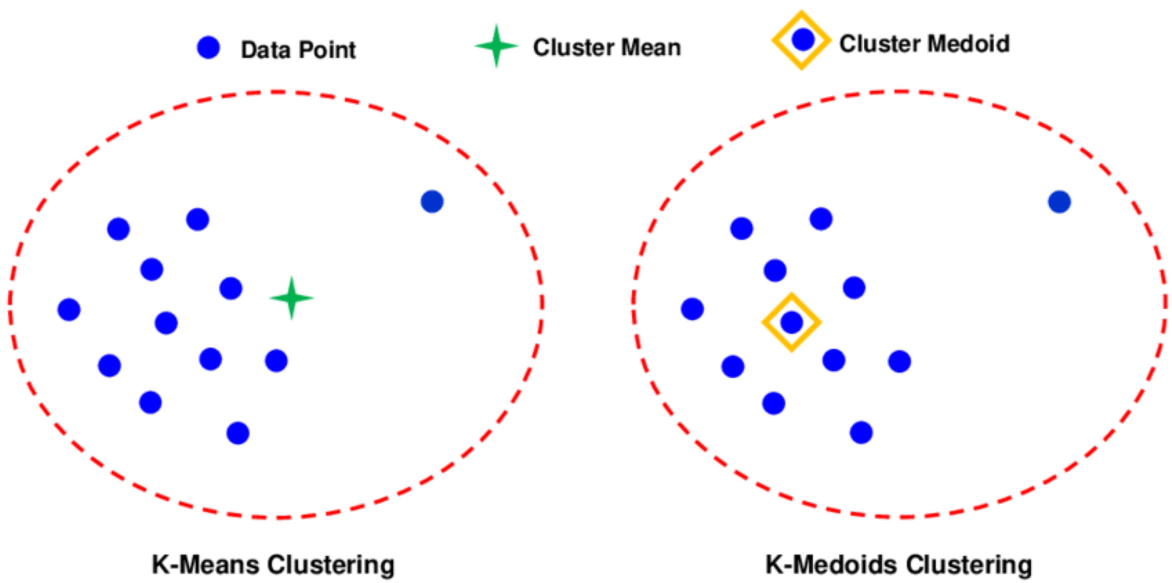
K-means extensions

- mini-batch K-means: improves the memory and computation
  - randomly sample a mini-batch of data points
  - update cluster centroids using only the sampled data, then resample (like gradient descent → stochastic gradient)



K-means extensions

> K-medoids: improves robustness
  - instead of using means, we chose the central data point (medoid)
  - we can use other distance (dissimilarity) measures (e.g., categorical)
  - it is more robust to outliers than K-means

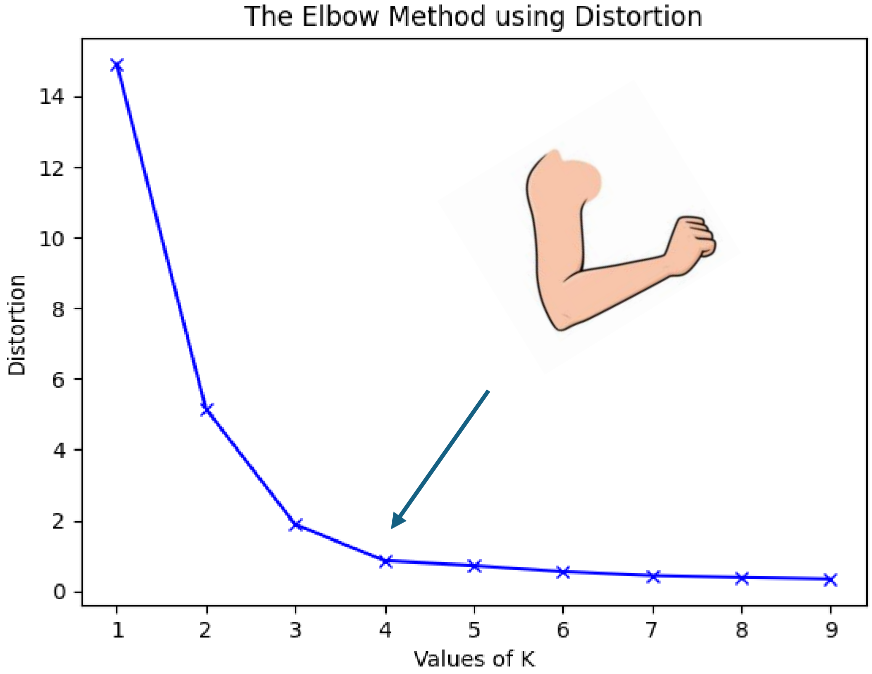


$$X = \{1,2,3,4,100\}$$

$$mean: 22$$
$$medoid: 3$$

K-means extensions

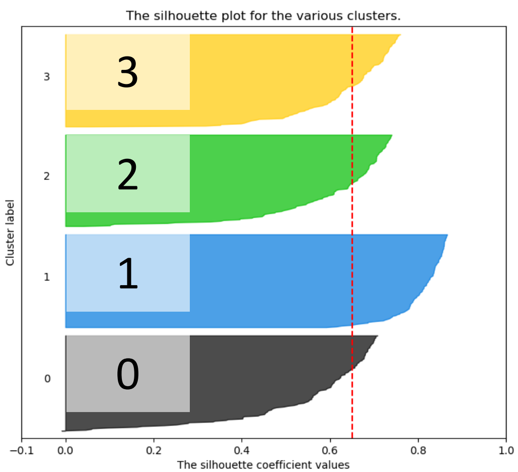
> how to choose the number of clusters k
  - elbow method
    - for each k, compute the cost
    - plot the cost vs. k and look for the elbow point
    - cost is monotonically decreasing (why?)

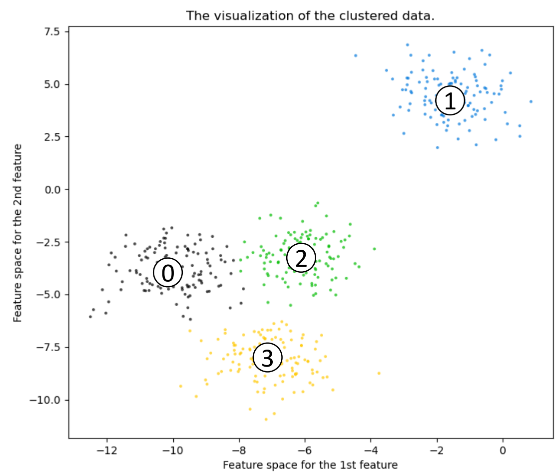


K-means extensions

> how to choose the number of clusters k
  - Silhouette analysis

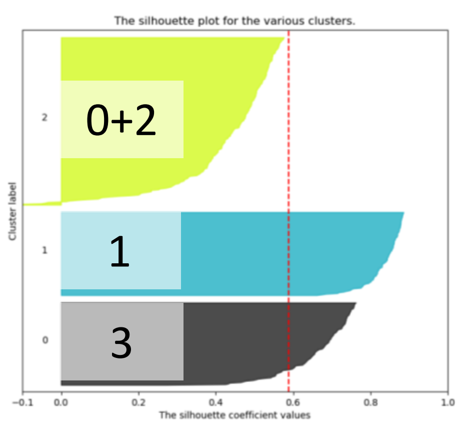
$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

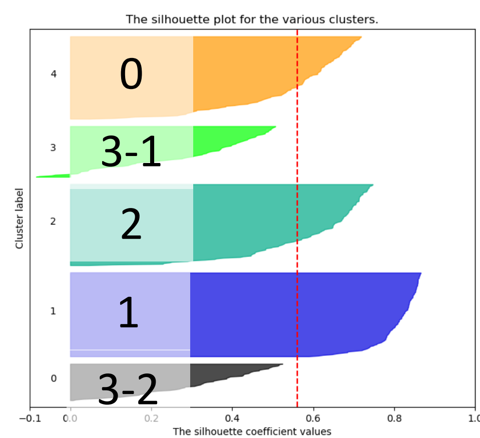


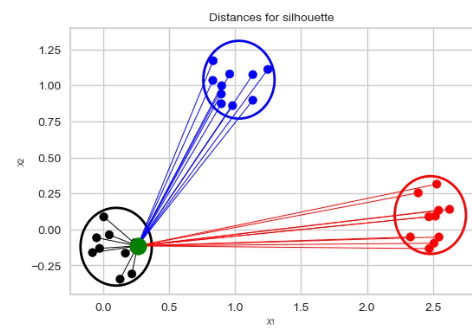


a_i: avg. distance to other points in the same cluster

b_i: avg. distance to points in the nearest different cluster

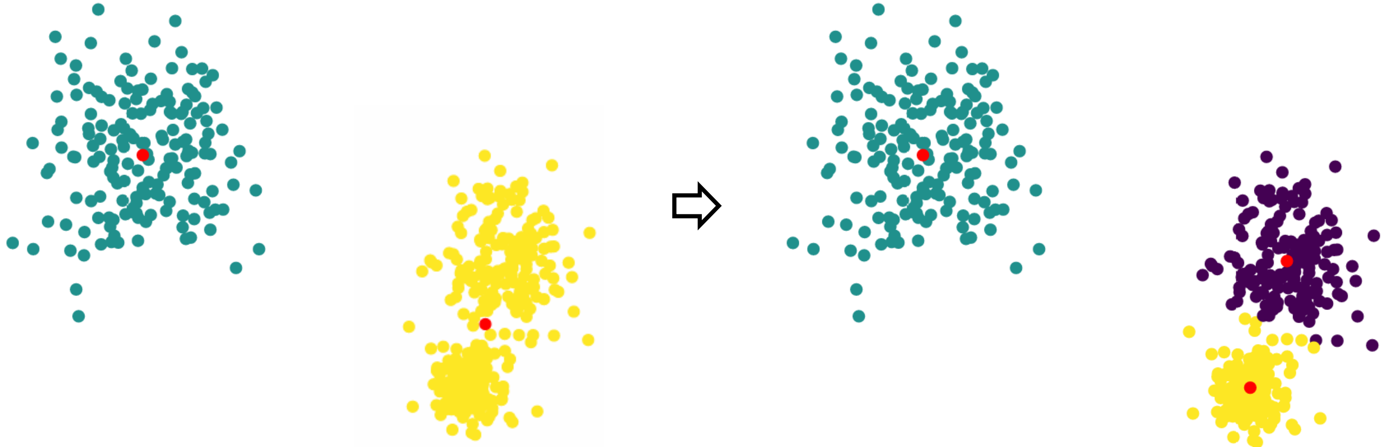






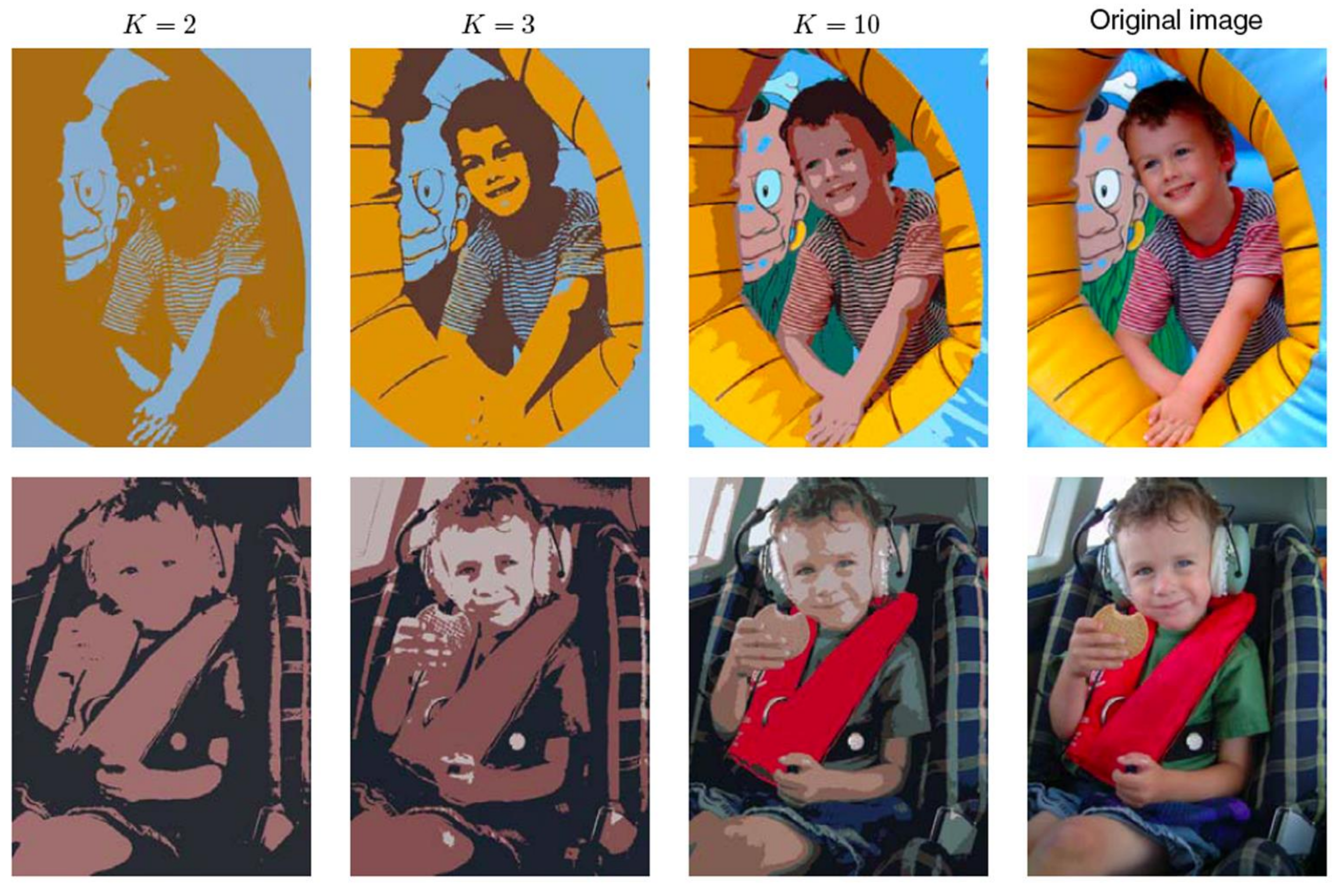
K-means extensions

- how to choose the number of clusters k
  - X-means: if splitting a cluster into two improves the model, then split it
    - based on Bayesian information criterion
  - G-means: if the data within a cluster doesn't look Gaussian, then split it
    - assumes each true cluster should follow a Gaussian distribution
  - Both start with small number of clusters (e.g., 1 or 2)



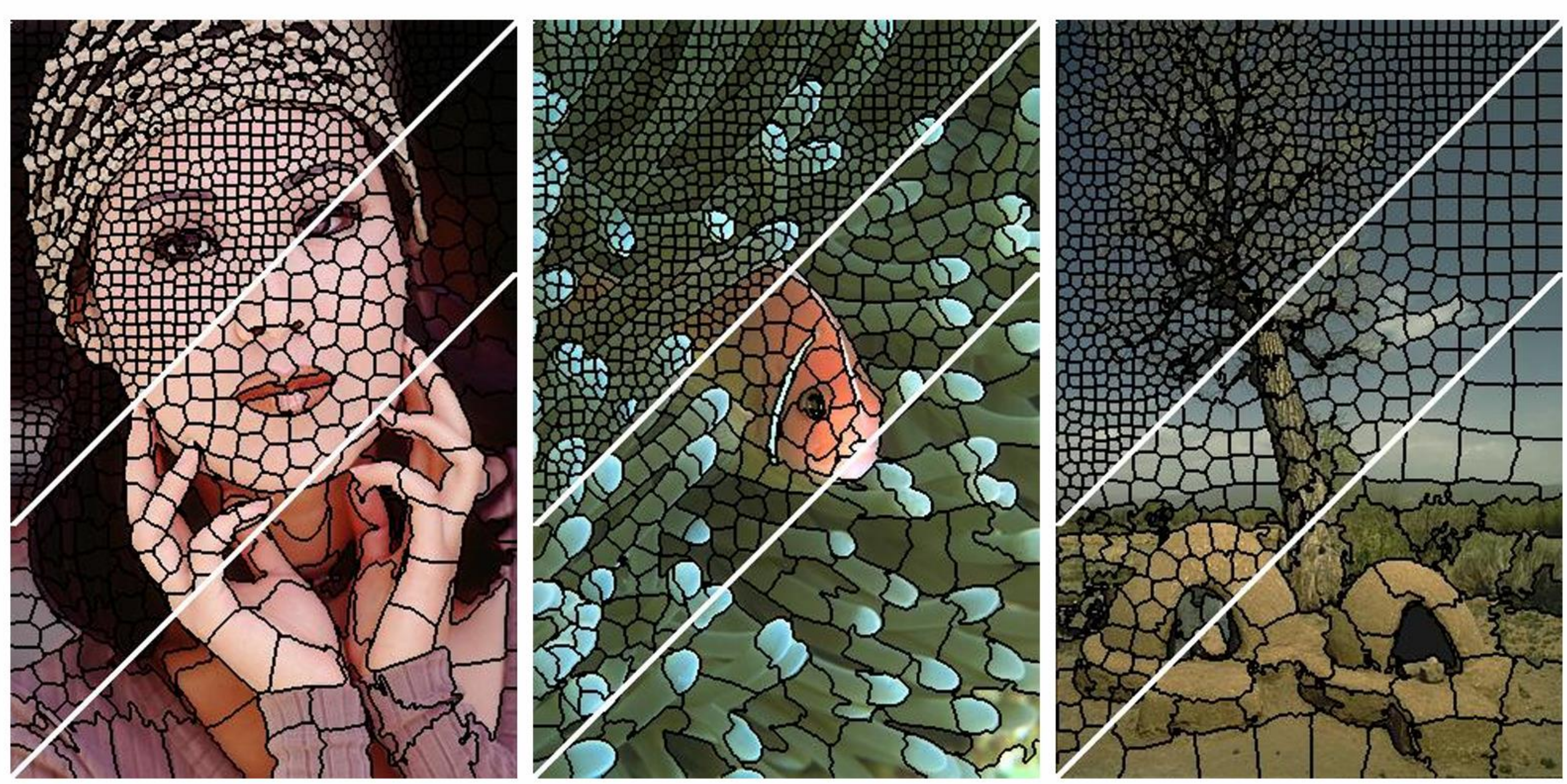
K-means applications

> Vector quantization
  - replace each image pixel with the closest representative color



K-means applications

- Image segmentation



K-means summary

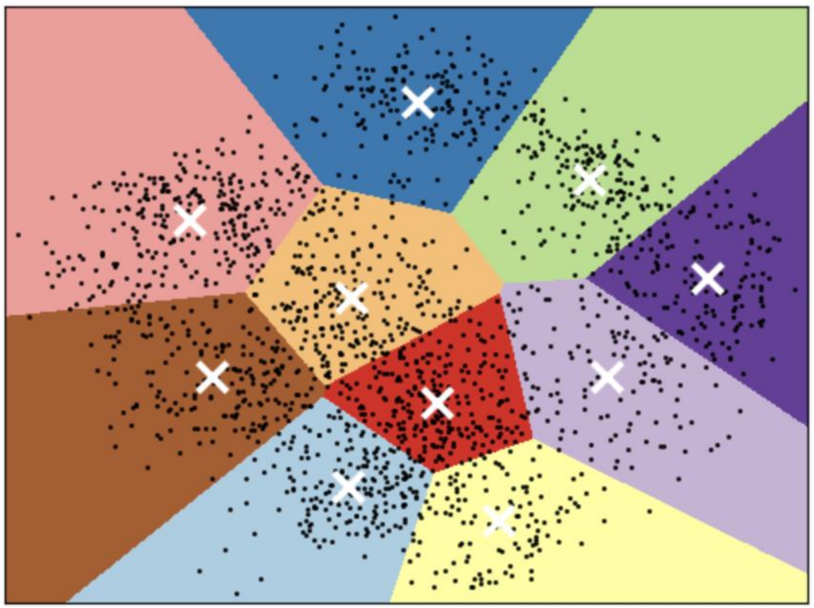
> To sum up,
  - clustering goal: segment datapoints into similar groups
  - k-means: iteratively optimize cluster assignments and centroids

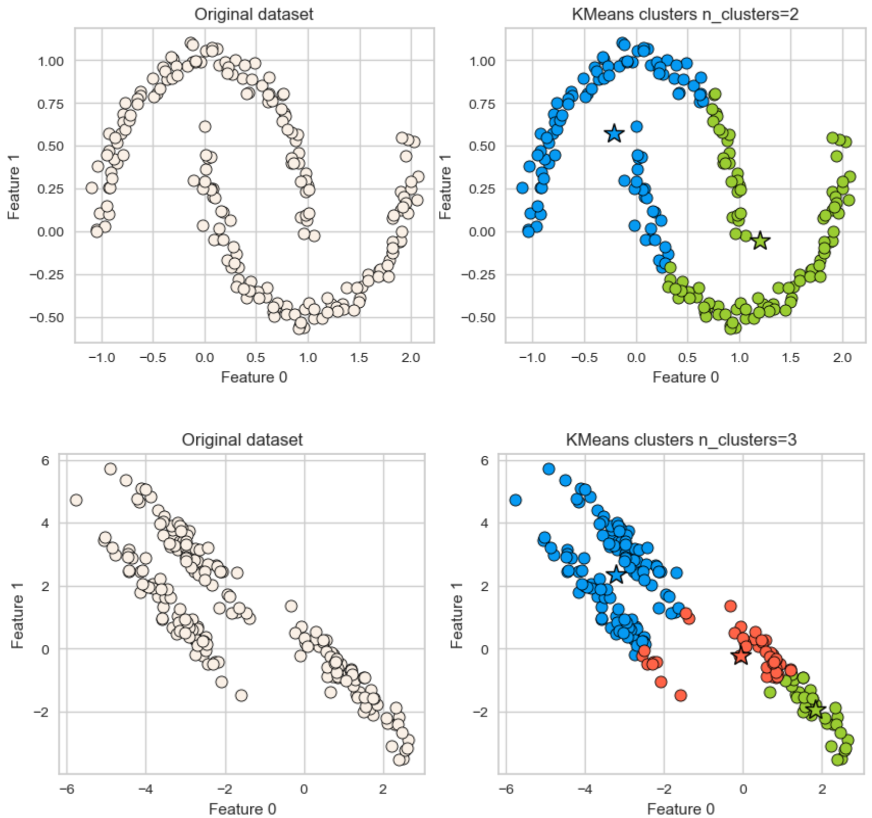
> Issues
  - squared Euclidean objective restrictive → K-medoids
  - non-convex optimization problem → K-means++
  - running time → mini-batch K-means
  - must choose k → elbow method, Silhouette analysis, X-means, G-means

> More issues
  - hard assignments are unstable under small perturbations of data
  - gives equal weight to each coordinates and clusters
  - clusters change arbitrarily with different k
  - works poorly on non-convex / non-spherical clusters
  - different cluster density and size

K-means limitations

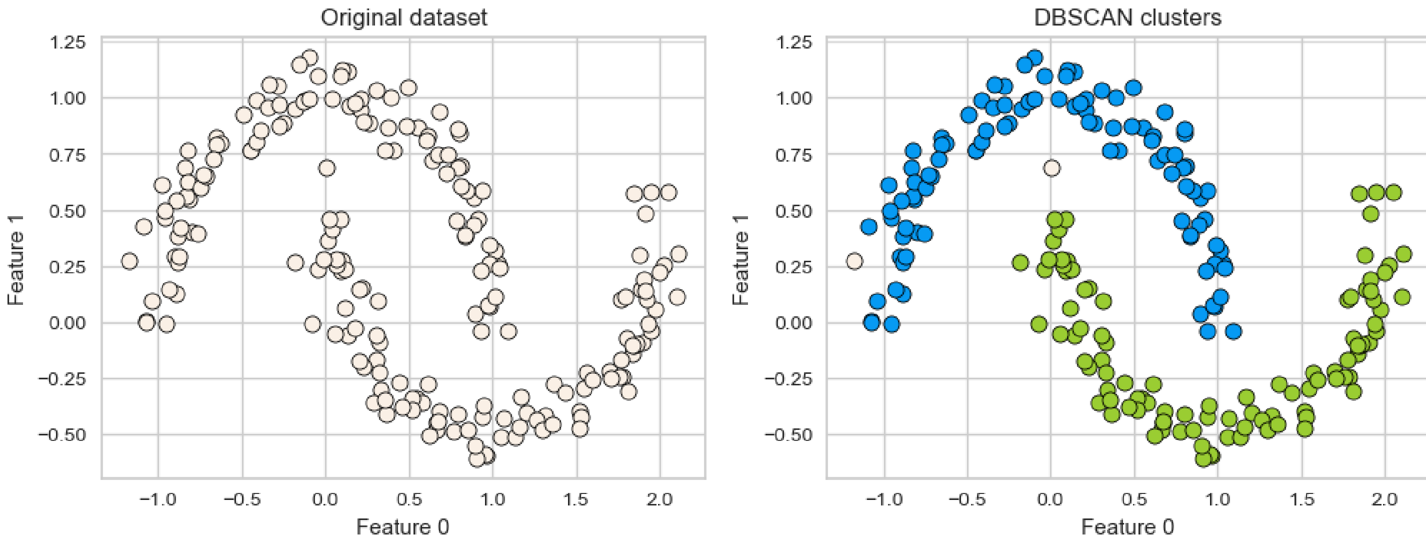
- Shape of K-means clusters
  - boundaries between clusters are linear





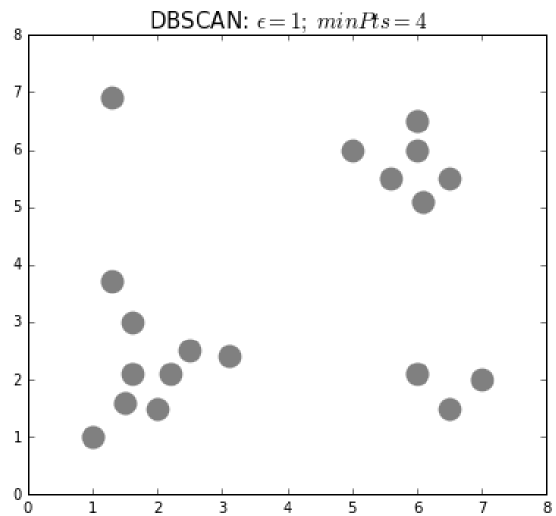
DBSCAN

- Density-based spatial clustering of applications with noise
  - idea: clusters are dense regions in the feature space, so identify them
  - it does not require to specify the number of clusters
  - it can identify points that are not part of any clusters
  - it can capture clusters of complex shapes

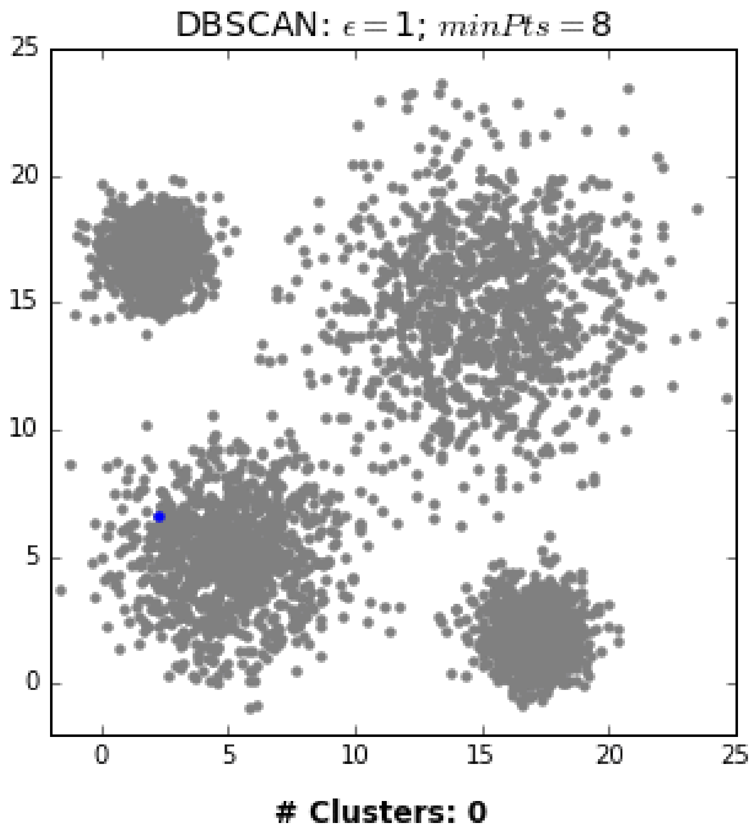


DBSCAN

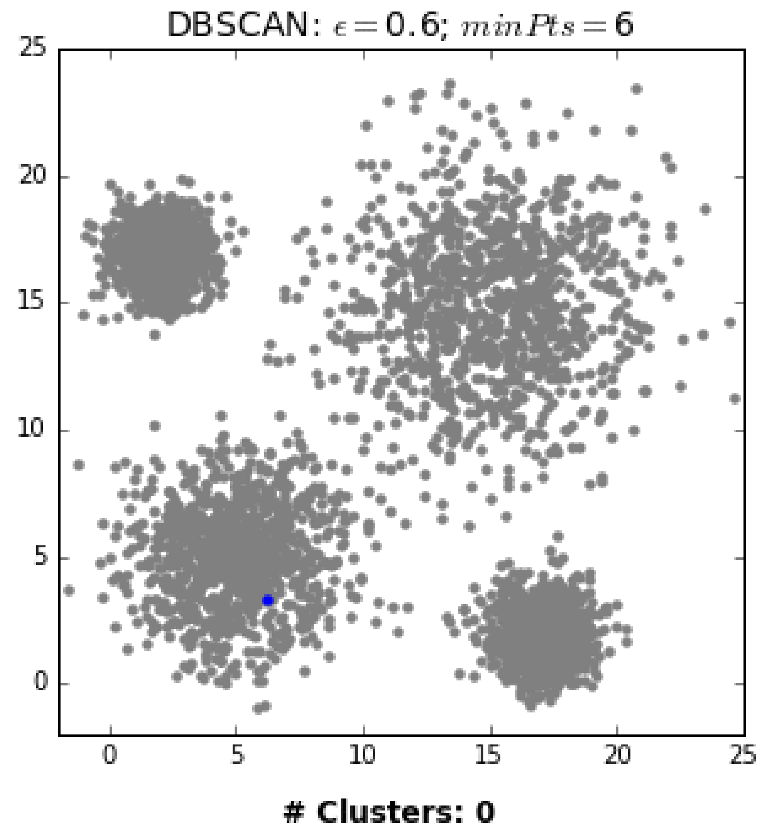
> Algorithm
  - pick a point at random
  - check whether the point is a core point
    - core point: that have at least n points within a distance of d
  - if the point is a core point, give it a color (label)
  - spread the color to all of its neighbors
  - check if any of the neighbors is a core point, if yes, spread the color
  - once there is no more core point left to spread, pick a new unlabeled point



DBSCAN



DBSCAN: $\epsilon = 1$; $minPts = 8$
25
20
15
10
5
0
0
5
10
15
20
25
Clusters: 0



DBSCAN: $\epsilon = 0.6$; $minPts = 6$
25
20
15
10
5
0
0
5
10
15
20
25
Clusters: 0

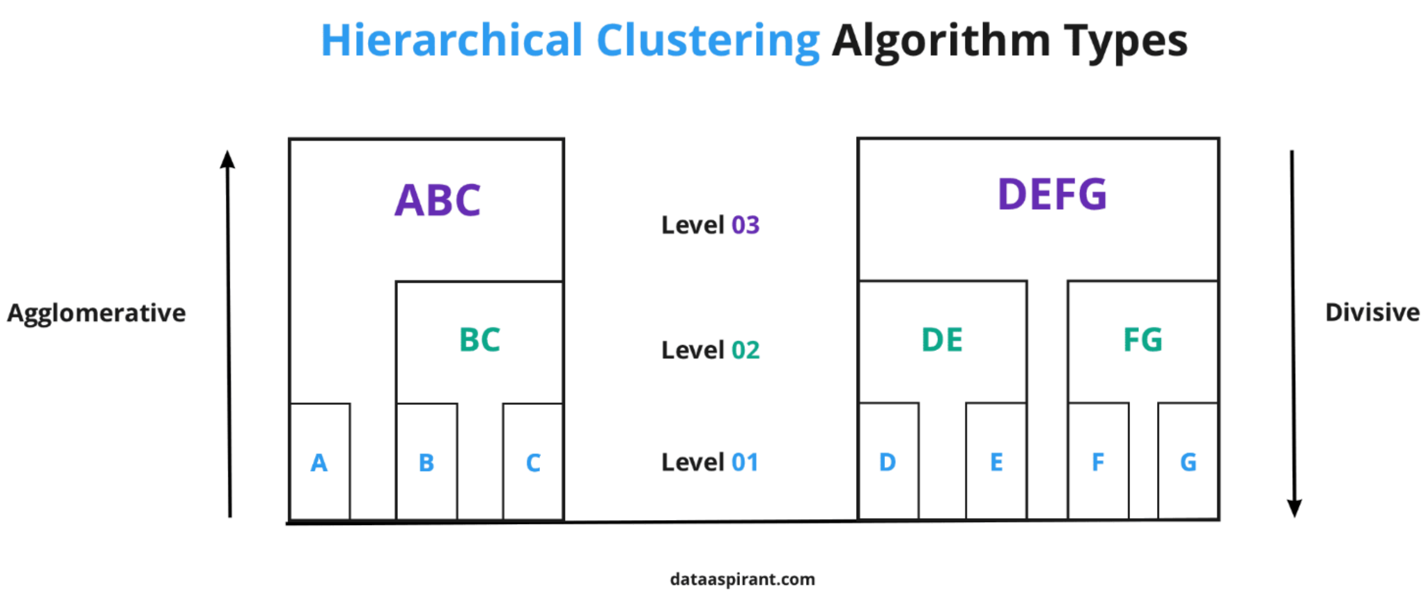
DBSCAN

> Pros and cons
  - can learn arbitrary cluster shapes
  - can detect outliers

  - cannot predict on new examples
  - needs tuning of two non-obvious hyperparameters (n, d)
  - doesn’t do well when we have clusters with different densities

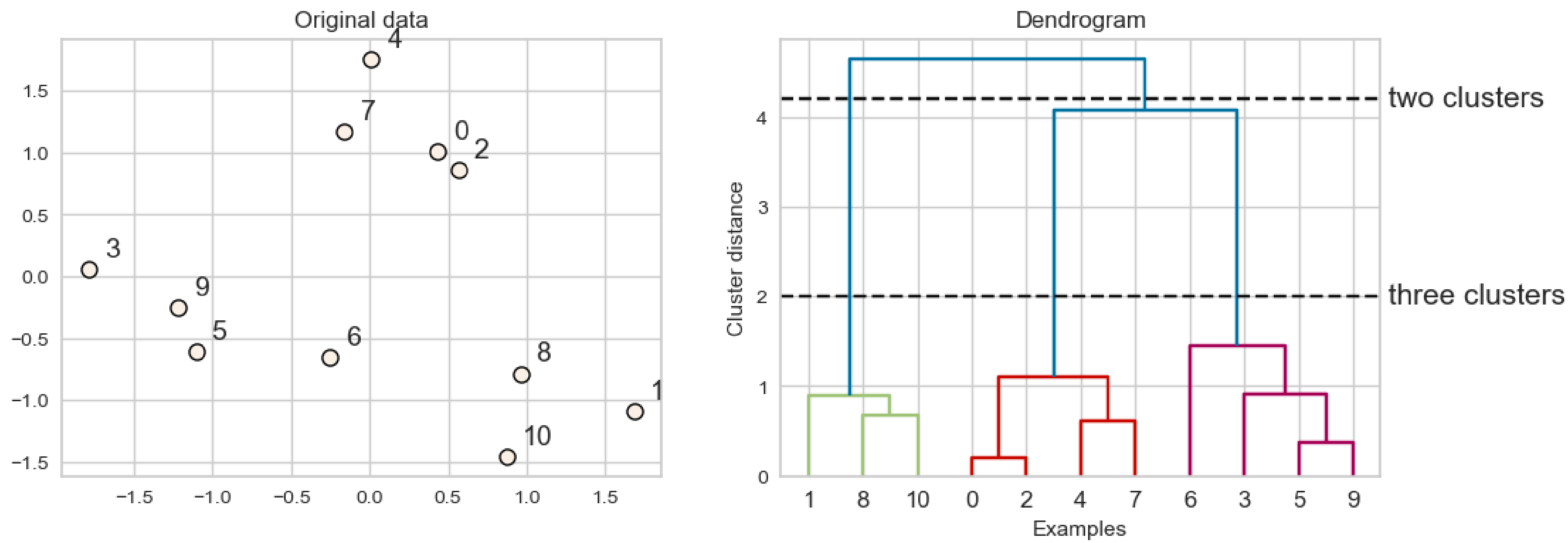
Hierarchical clustering

- Construct a hierarchy of clusters based on proximity
  - Divisive: start with the one cluster (entire dataset) and split
  - Agglomerative: start with the all individual points (all clusters) and combine



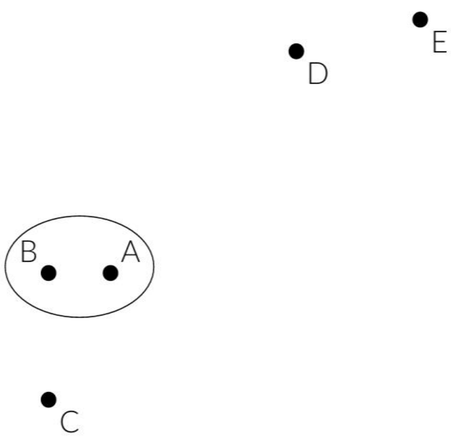
Hierarchical clustering

> Dendrogram
  - a tree that shows how clusters are merged/split hierarchically
  - a clustering is obtained by cutting the dendrogram at the desired level



Hierarchical clustering

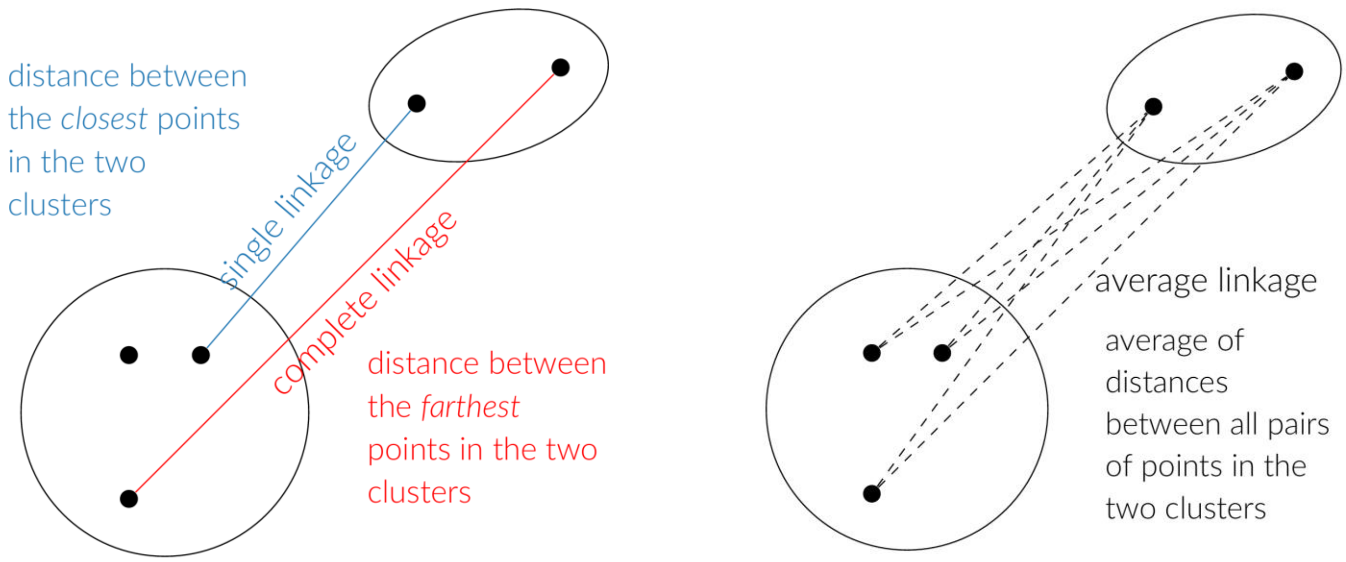
- Agglomerative
  - first, we merge the two closest points into a cluster
  - next, we want to merge the next closest into a cluster



- How do we measure the distance between a cluster and a point?
- How do we measure distance between two clusters?

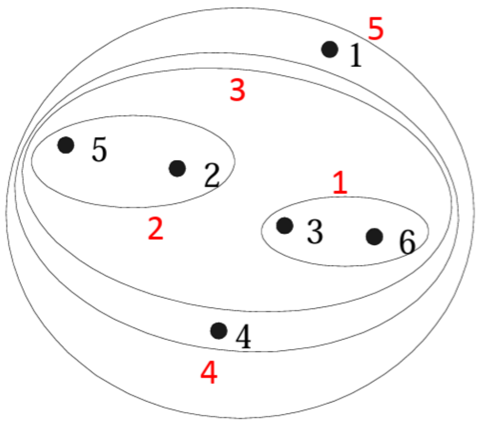
Hierarchical clustering

> How to measure distances between clusters is called the linkage
  - simple linkage: minimal distance
  - complete linkage: maximum distance
  - average linkage: average distance between all pairs
  - ward linkage: increase in within-cluster variance



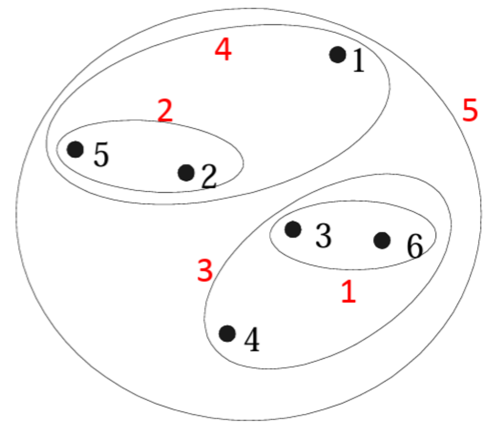
Hierarchical clustering

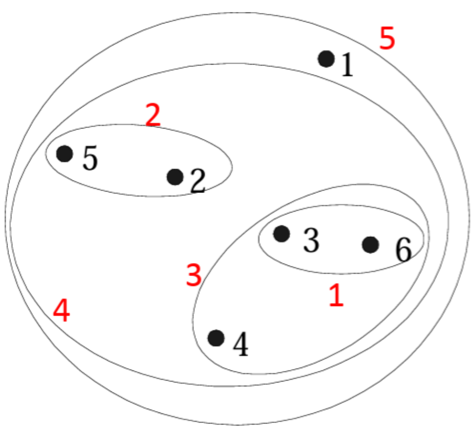
> Example



MIN

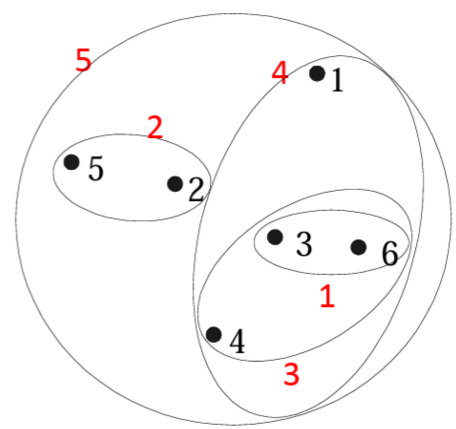
MAX





Group Average

Ward's Method



Hierarchical clustering

- Pros and cons
  - do not have to assume any particular number of clusters
  - easy to decide the number of clusters by merely looking at the Dendrogram
  - they may correspond to meaningful taxonomies

  - once a decision is made to combine two clusters, it cannot be undone
  - no objective function is directly minimized
  - does not work well on vast amounts of data
  - different measures have problems with one ore more:
    - sensitive to noise and outliers
    - breaking large clusters
    - difficulty handling different sized clusters and irregular shapes

Reference

- K-means
  - https://web.stanford.edu/~lmackey/stats306b/doc/stats306b-spring14-lecture1_slides.pdf
  - https://www-users.cse.umn.edu/~jwcalder/Clustering.pdf
  - https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/slides/lec15-slides.pdf
- Other clustering
  - http://pajarito.materials.cmu.edu/Data_Analytics-lectures/27737-Clustering-L11-24Mar21.pdf
  - https://ubc-cs.github.io/cpsc330-2024W2/lectures/notes/16_DBSCAN-hierarchical.html
  - https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/
  - https://cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_hierarchical.pdf
  - https://dlsun.github.io/stats112/slides/hierarchical_clustering.pdf