Clustering

Zhiyao Duan

Associate Professor of ECE and CS

University of Rochester

Some figures are copied from the following book

- **GBC** - Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press.
- **LWLS** - Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön, *Machine Learning: A First Course for Engineers and Scientists*, Cambridge University Press, 2022.

Machine Learning Paradigms

- Supervised learning
  - Given examples $(\boldsymbol{X}, \boldsymbol{Y})$, learn $f: \boldsymbol{x} \mapsto \boldsymbol{y}$

- Unsupervised learning
  - Given examples $\boldsymbol{X}$, discover structures of data

- Semi-supervised learning
  - Given examples $(\boldsymbol{X}^l, \boldsymbol{Y}^l)$ and $\boldsymbol{X}^u$, learn $f: \boldsymbol{x} \mapsto \boldsymbol{y}$

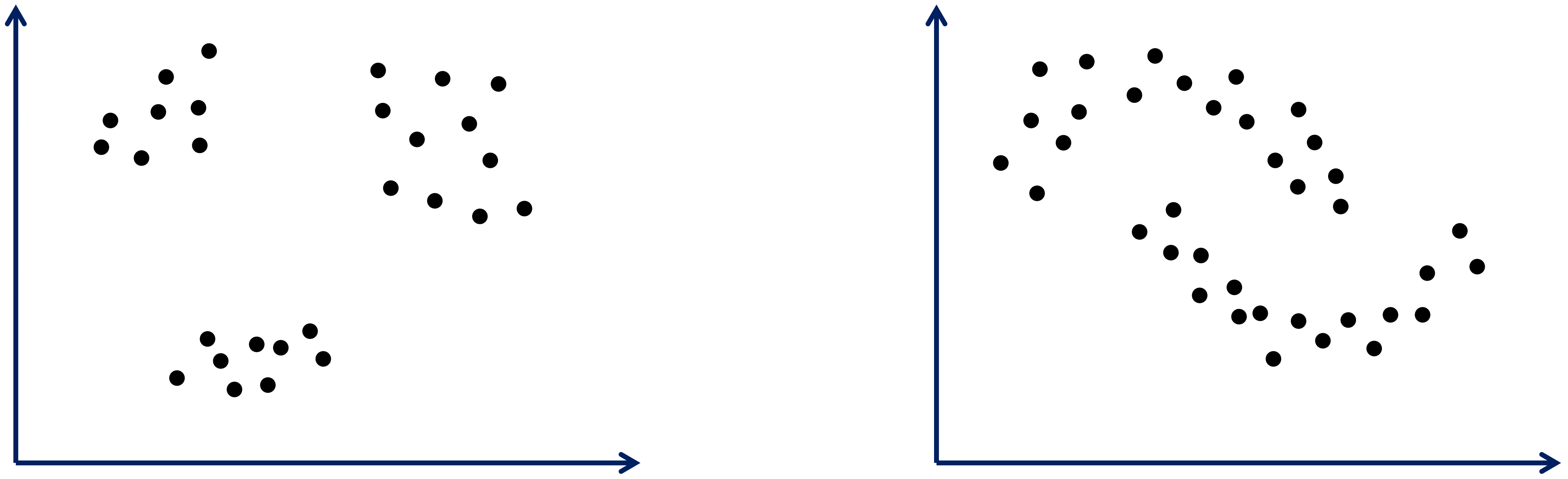
- Reinforcement learning
  - Given sequences of (state, action, immediate reward): $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{r})$
  - Learn optimal behavior $f: \boldsymbol{s} \mapsto \boldsymbol{a}$ that is good in the long run

Unsupervised Learning

- Given examples $\boldsymbol{X}$, discover structures of data
- Density estimation: how is data distributed in the data space?
  - E.g., finding out the distribution of SAT scores in college applications
- Clustering: which data examples form a cluster?
  - E.g., sorting out types of insects
- Dimensionality reduction: find the lower dimensional subspace or manifold where the data resides
  - E.g., reducing a 4K image (8.3M pixels) to a 100-d vector for scene classification
- Data generation: sample from data distribution
  - E.g., https://thispersondoesnotexist.com/

Clustering – Grouping Data Points

- How would you cluster these data points, and why?



- Idea 1: points in the same cluster should be close or "connected" to each other?
- Idea 2: different clusters should be far or separated from each other?

Designing Clustering Algorithms

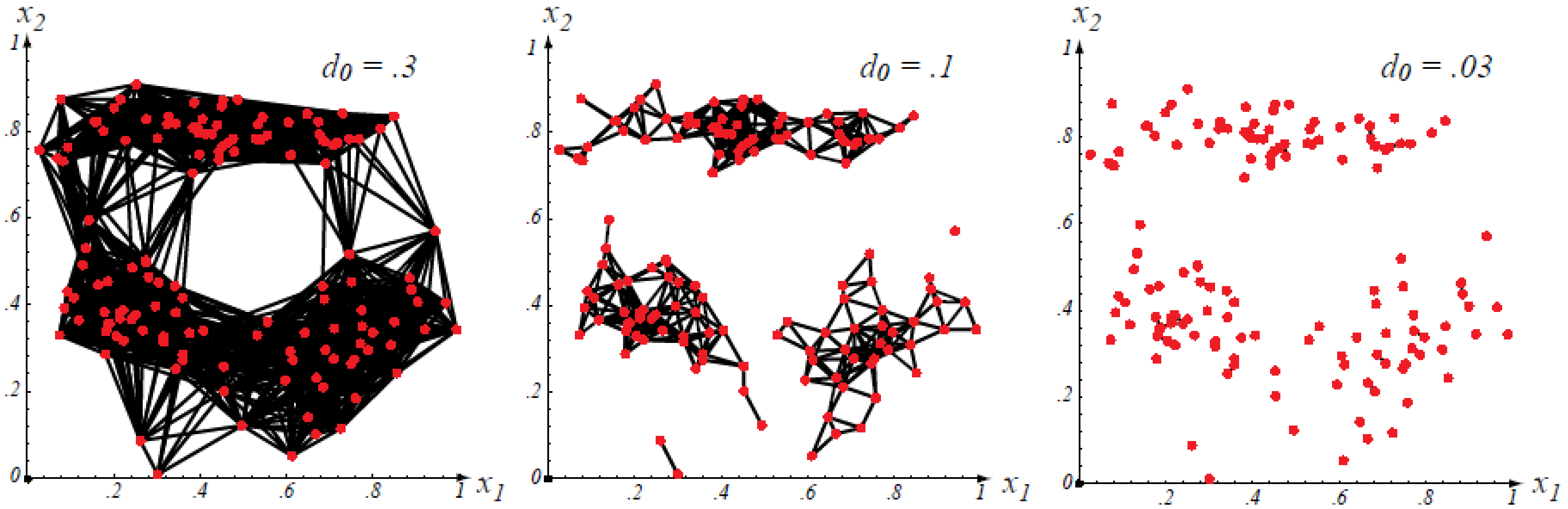
- We need some measure for "distance", "similarity" or "proximity" between data points
  - E.g., Euclidean distance, L^p distance, cosine similarity, K-L divergence, Mahalanobis distance, geodesic distance
  - How to weigh different dimensions in distance calculation?

- We need an efficient algorithm to cluster data points such that points in the same cluster are close and/or points in different clusters are far away
  - For N data points forming K clusters, how many possible clustering results (i.e., partitions of data)?

- We often need to decide how many clusters to output
  - Two trivial extremes: All data in one cluster, each point is one cluster

A Threshold-Based Algorithm

- Algorithm: put two points into the same cluster if their Euclidean distance is smaller than a threshold d_0



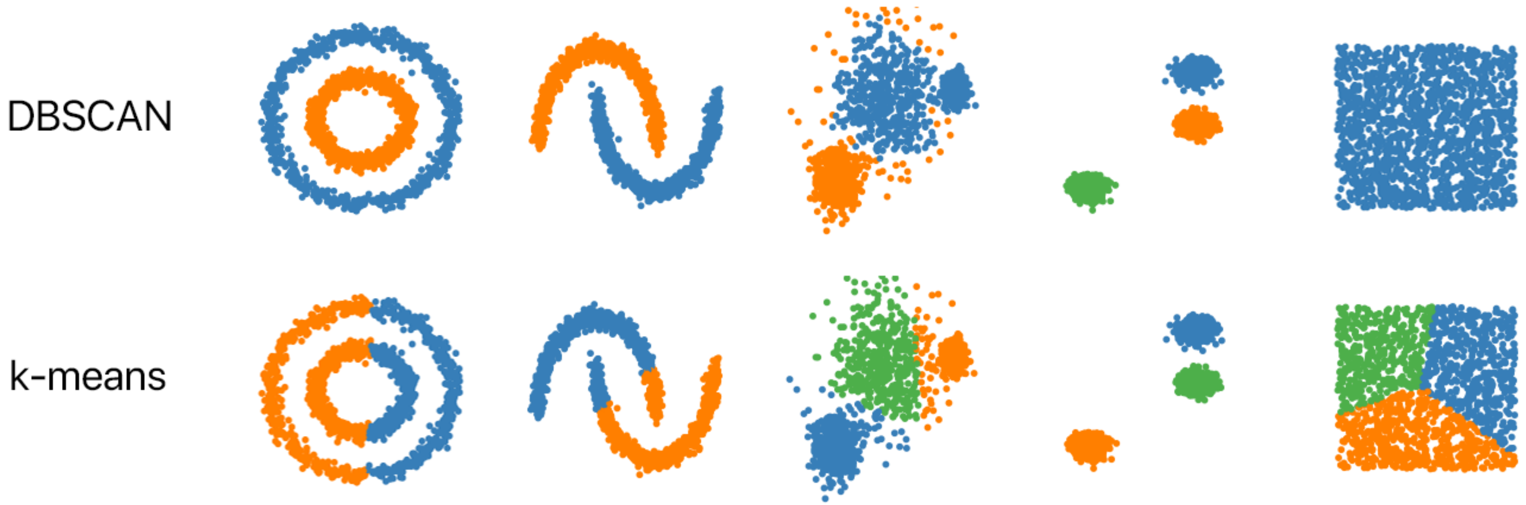
(Fig. 10.7 from Duda, Hart & Stork, Pattern Classification, 2001)

- Inductive bias: clusters are connected subgraphs
- Obviously, the result is very sensitive to d_0

Density-Based Clustering

- Key assumptions
  - Each cluster is a contiguous region with high data density
  - Clusters are separated by contiguous regions with low data density

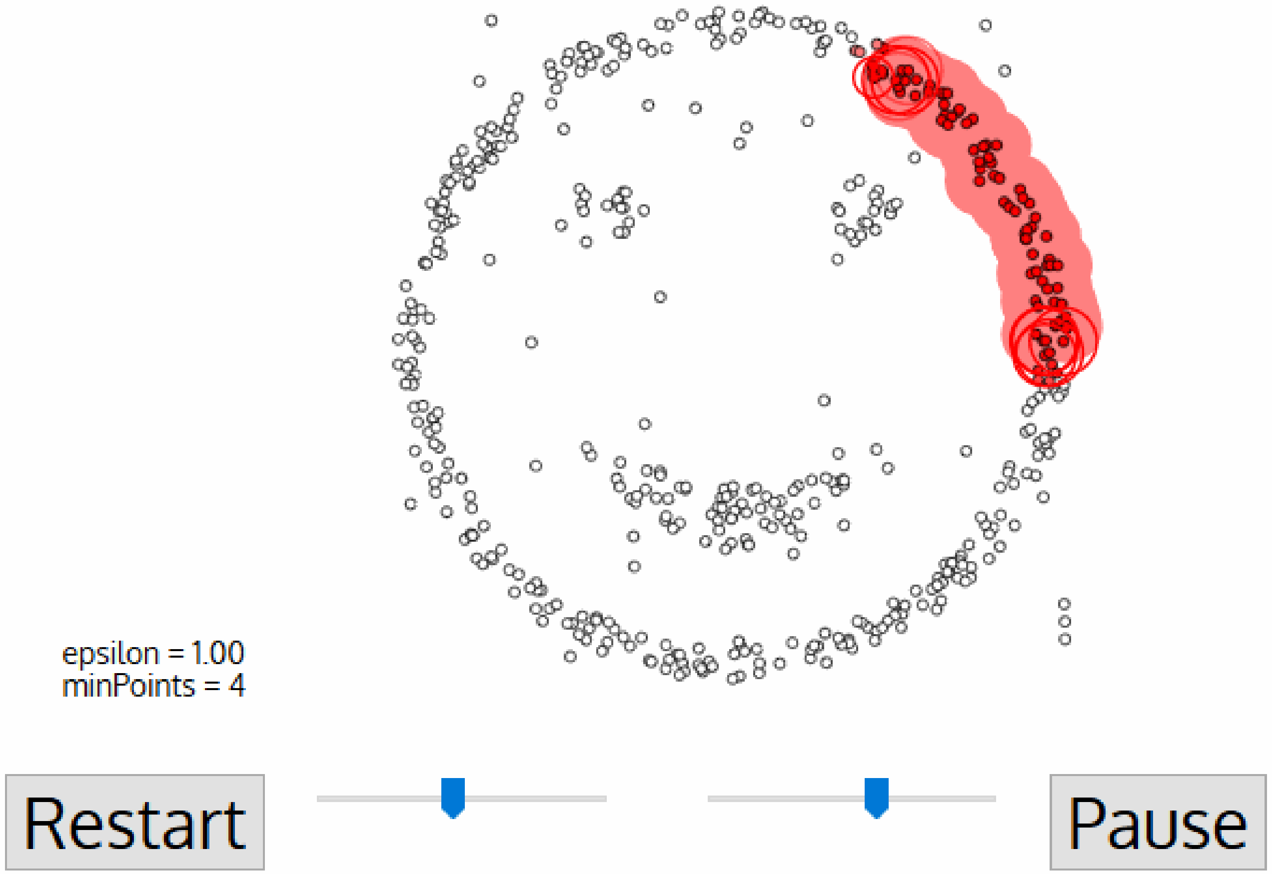
- **DBSCAN**: Density-Based Spatial Clustering of Applications with Noise



(Figure from https://github.com/NSHipster/DBSCAN)

DBSCAN

- Algorithm
  - Randomly pick an unvisited point
  - Check neighborhood with distance ϵ
    - If #neighbors > minPoints, then expand cluster to these neighbors
  - Repeat till all points are visited
- Pros
  - Works with arbitrary cluster shapes and sizes
  - Robust to noise and outliers
- Cons
  - Hard to deal with clusters with different densities
  - Hard to pick parameters in high-dimensional space

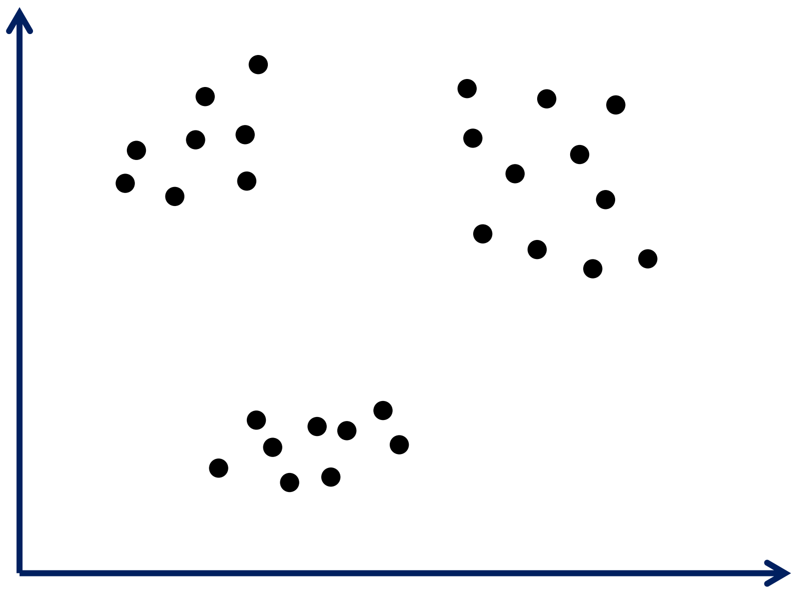


(Figure from https://www.digitalvidya.com/blog/the-top-5-clustering-algorithms-data-scientists-should-know/)

Iterative Algorithm

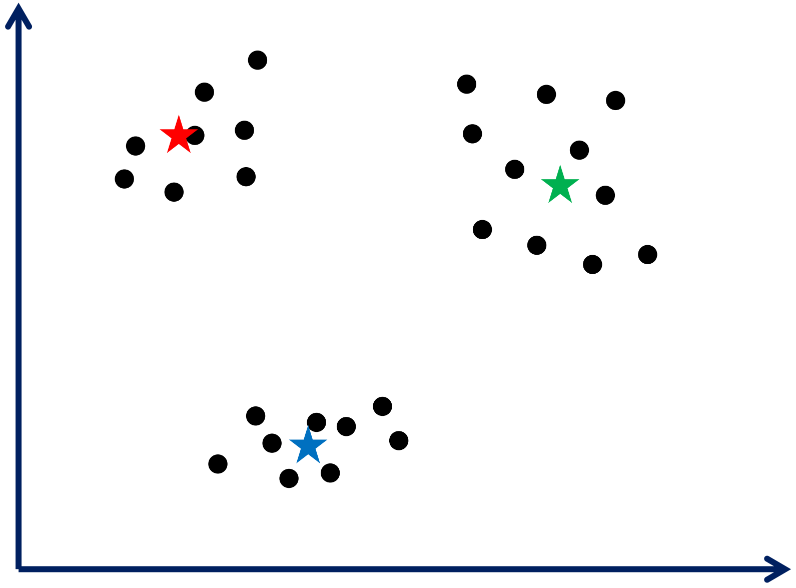
- Idea: start from some initial clustering, and then iteratively update it

- Where to start? Assuming there are K groups of data
  - Random partition of the dataset into K groups?
- How to update clustering?
  - Updating clustering means that the assignment of some data points needs to be changed
  - Assign to a closer cluster? Closer in what sense?
- When to stop?
  - Stop till convergence?
  - Stop after certain iterations?



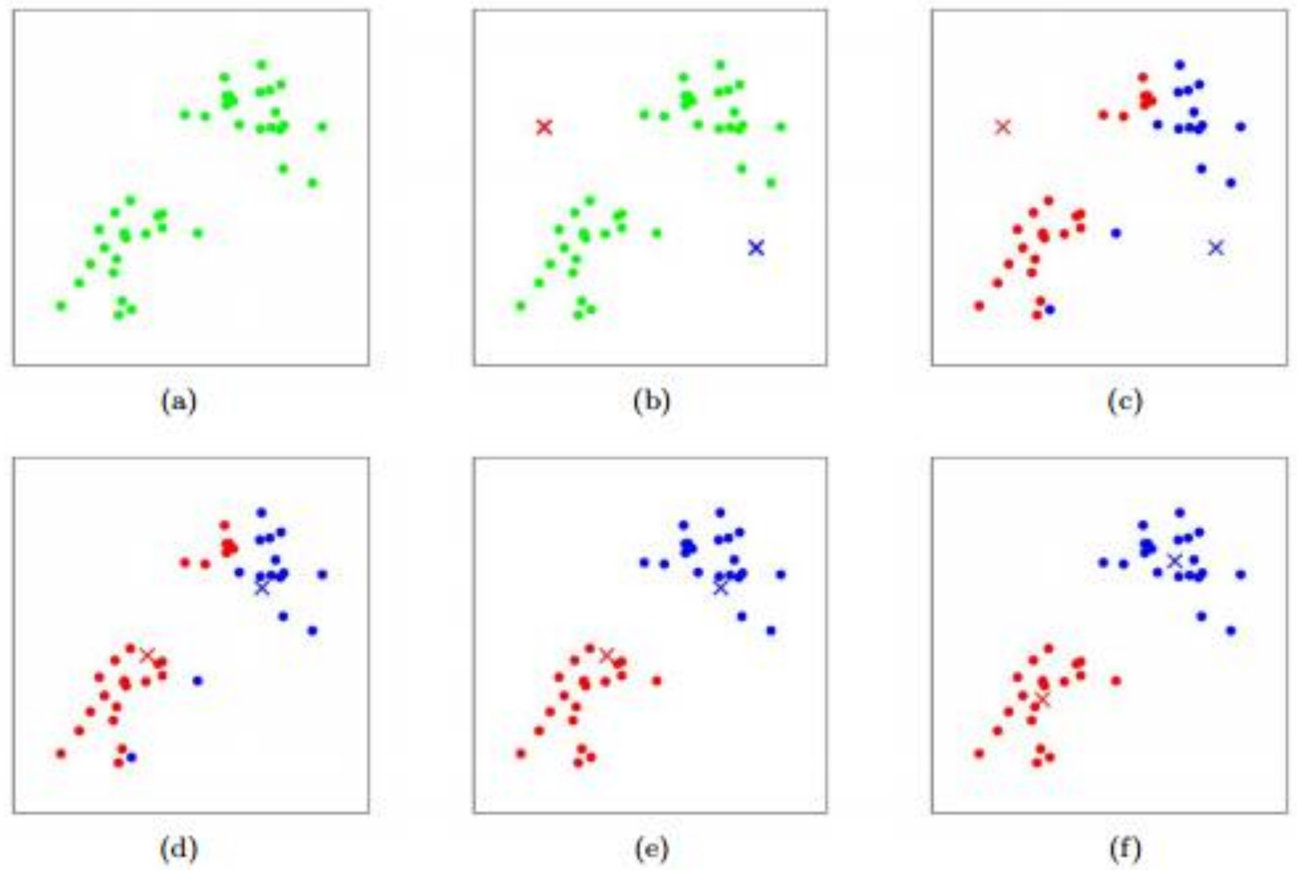
Centroid-Based Clustering

- Specify the number of clusters: K
  - Let K=3 for the right example
- Represent a cluster with a "centroid"
  - Compute centroid as the mean of data points in the cluster
  - Require cluster assignment for all data points
  - Ignore the shape of the cluster
- Assign each data point to its closest centroid
  - Require centroids computed
- "Chicken and egg problem"?
  - Iterative!



K-Means Clustering

- Initialization: pick K centroids $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_K$
- Iterate
  - Update clusters $S_1, \cdots, S_K$ by assigning each data point $\boldsymbol{x}$ to the closest centroid based on Euclidean distance
    $$a(\boldsymbol{x}) = \underset{k=1,\cdots,K}{\operatorname{argmin}} \|\boldsymbol{x} - \boldsymbol{c}_k\|_2$$
  - Update each centroid as the mean of data points in the cluster
    $$\boldsymbol{c}_k = \frac{1}{|S_k|} \sum_{\boldsymbol{x} \in S_k} \boldsymbol{x}$$
  - Repeat until convergence (?)
- Does it converge?
- How to measure clustering quality?



(Figure courtesy to Michael Jordan)

Intra-Cluster Squared Distance

- Measure a clustering (partition) Π with

$$J(\Pi) = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in S_k} \|\boldsymbol{x} - \boldsymbol{c}_k\|_2^2$$

- Minimizing $J(\Pi)$ results in more "compact" clusters

- Note that this is a combinatorial optimization problem, as the parameter Π is cluster assignment of all data points
  - Gradient-based optimization techniques cannot be used
  - Brute-force search is intractable

- K-Means is an iterative algorithm, but does it optimize (decrease) this objective function?

Does K-means decrease the objective?

$$J(\Pi) = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in S_k} \|\boldsymbol{x} - \boldsymbol{c}_k\|_2^2$$

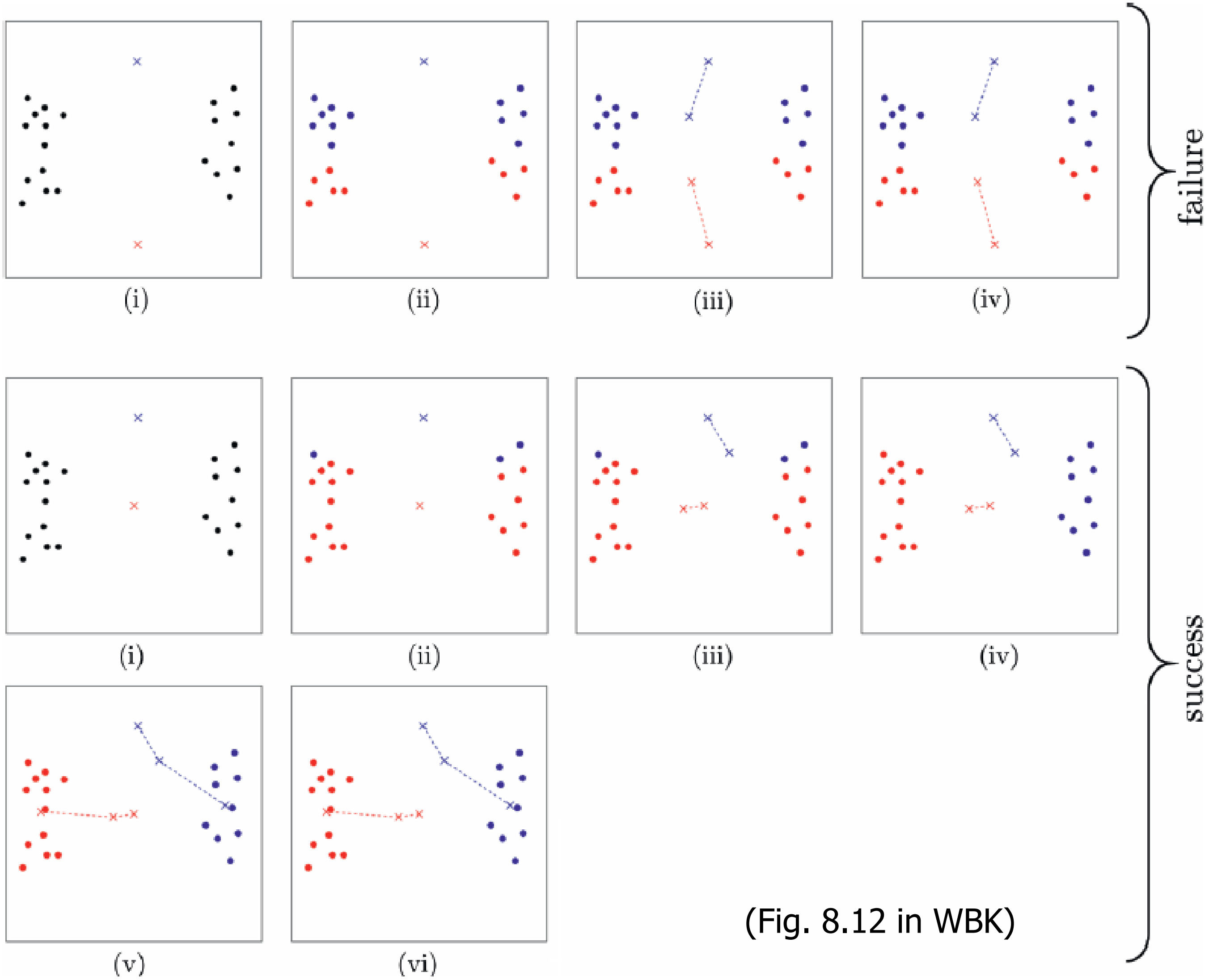
- Update clusters by assigning each data point $\boldsymbol{x}$ to the closest centroid
  - Yes! Each data point is involved in one term and the term is decreased
- Update each centroid as the mean of data points in the cluster
  - Yes! For each cluster, the mean $\boldsymbol{c}_k = \frac{1}{|S_k|} \sum_{\boldsymbol{x} \in S_k} \boldsymbol{x}$ minimizes $\sum_{\boldsymbol{x} \in S_k} \|\boldsymbol{x} - \boldsymbol{c}_k\|_2^2$
  - Quadratic function of $\boldsymbol{c}_k$: Let derivative w.r.t. $\boldsymbol{c}_k$ equal zero

- Both steps in each iteration decrease the objective (monotonically)!
  - Must converge because the solution space is finite

- Convergence condition: cannot re-assign any point to decrease objective

Initialization

- K-means algorithm is greedy, and it usually converges to a local minimum
- Different initialization of centroids may result in very different final clustering
- A common approach is to randomly pick K data points as the initial centroids
- Run multiple times with different initializations and choose the best one

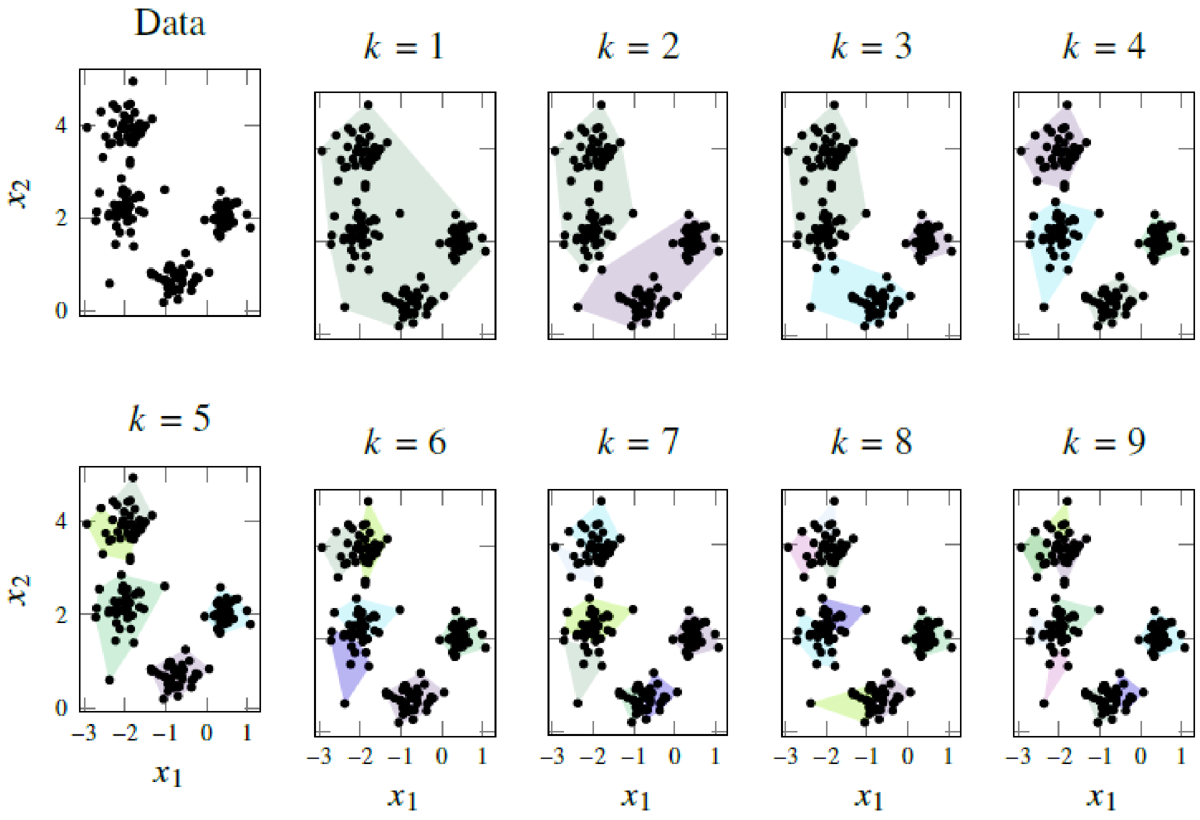


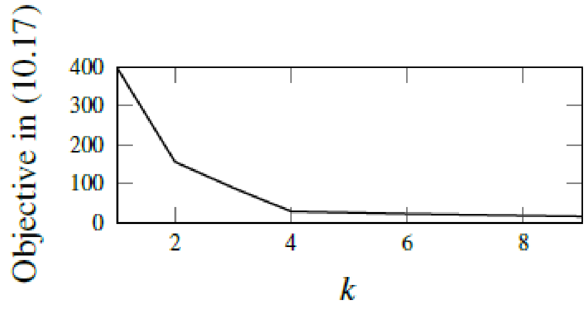
(Fig. 8.12 in WBK)

How to pick K?

- A too small K would combine some clusters
  - E.g., extreme case $K = 1$
  - Underfitting
- A too large K would split clusters
  - E.g., extreme case $K = N$
  - Overfitting

- As K increases, the objective (intra-cluster squared distance) at its optimal clustering decreases
- Choose K = "elbow"
  - Increasing K does not further decrease objective much





(Fig. 10.9 in LWLS)

Issues of K-means

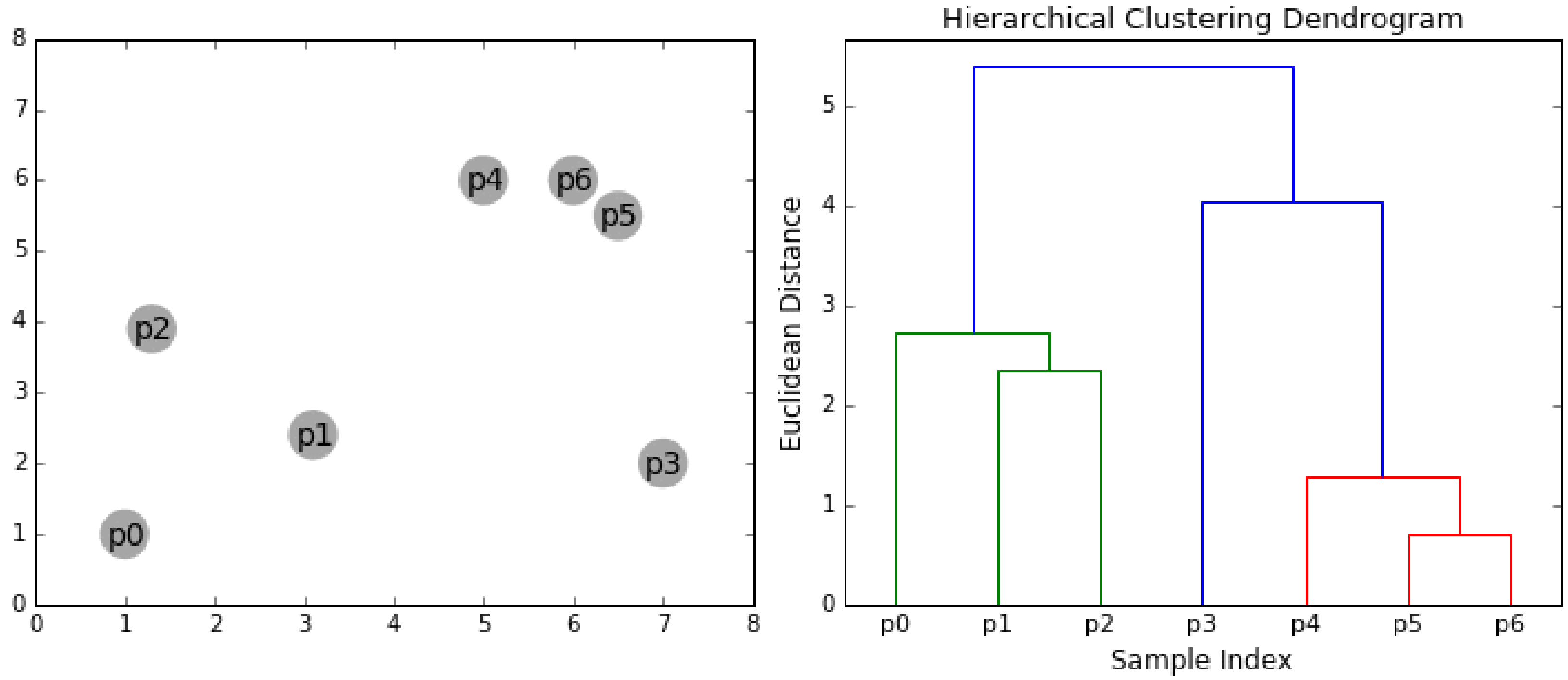
- Greedy algorithm, sensitive to initialization
  - Introduce some randomness into the cluster assignment process through simulated annealing?

- Each cluster is represented only by its mean, not suitable for non-round shapes (in fact, non isotropic Gaussian distributions)
  - Perhaps using both mean and covariance to represent a cluster?
  - Then use Mahalanobis distance?

- Cluster assignment is "hard", not allowing "soft" membership
  - Change to "soft" clustering methods?
  - Fuzzy K-means
  - Gaussian Mixture Model (GMM)

Hierarchical Agglomerative Clustering

- Start with each data point as a separate cluster
- Merge two clusters with the smallest average linkage
  - Average linkage between two clusters is defined as the average distance of all data pairs across the two clusters
- Repeat



(Figure from https://www.digitalvidya.com/blog/the-top-5-clustering-algorithms-data-scientists-should-know/)

Summary

- Clustering (e.g., assigning data points to different clusters) is an unsupervised learning problem

- Centroid-based
  - K-means
    - Optimizes the intra-cluster squared distance objective
    - Converges to local minimum
    - Initialization
    - How to choose K
- Density-based
  - DBSCAN
- Hierarchical Agglomerative Clustering