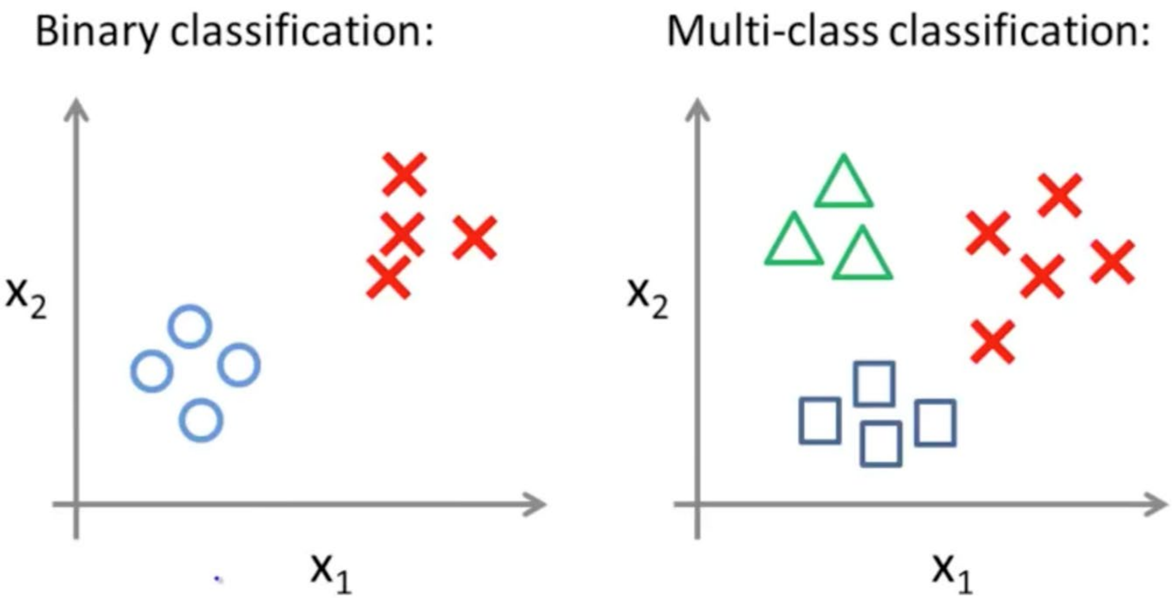
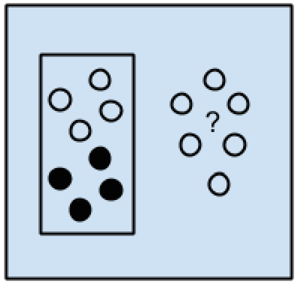
Unsupervised Learning

Classification



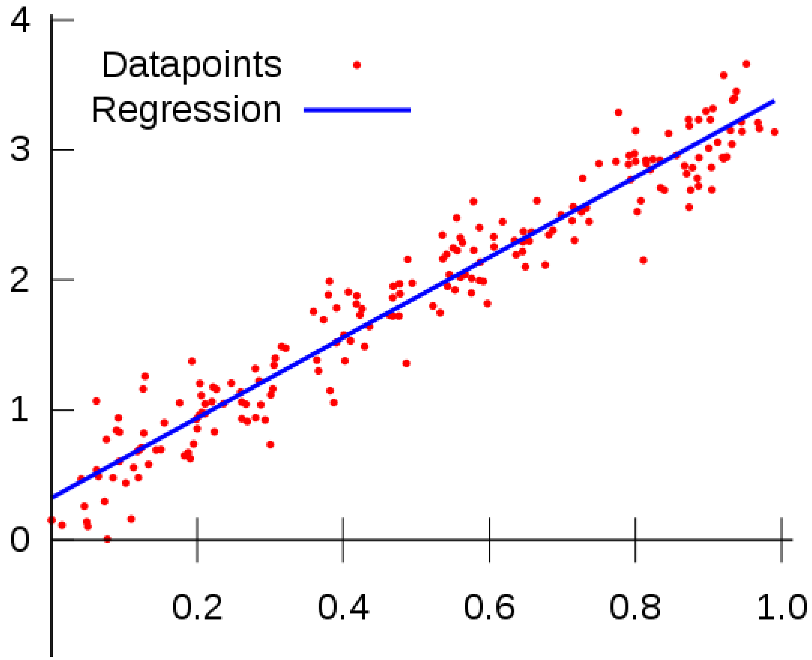
Supervised Learning



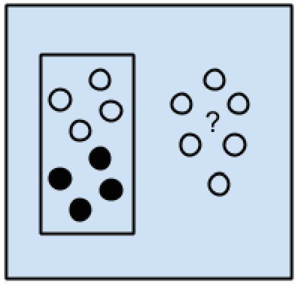
Supervised Learning Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Regression



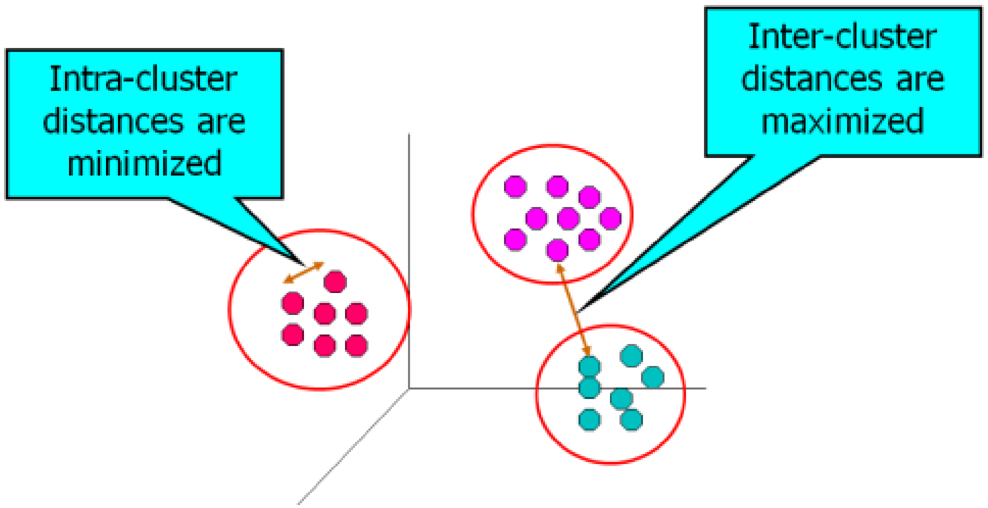
Supervised Learning



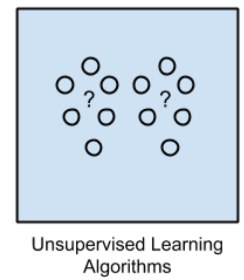
Supervised Learning Algorithms

https://quantdare.com/machine-learning-a-brief-breakdown/
https://medium.com/simple-ai/linear-regression-intro-to-machine-learning-6-6e320dbdaf06

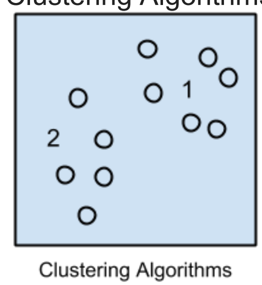
Clustering



Unsupervised Learning

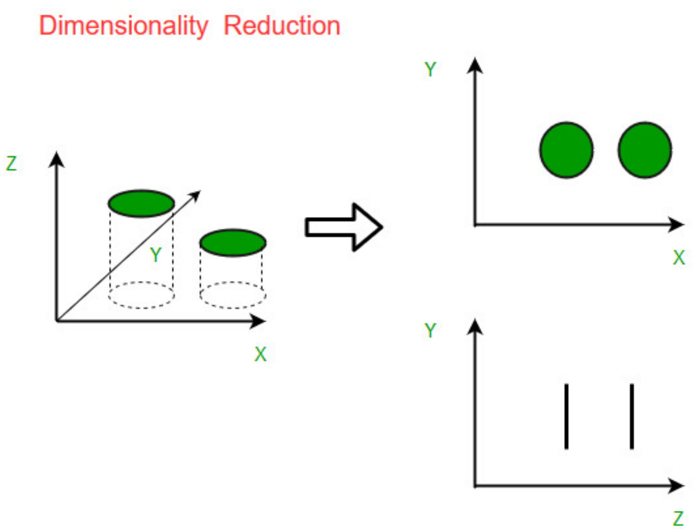


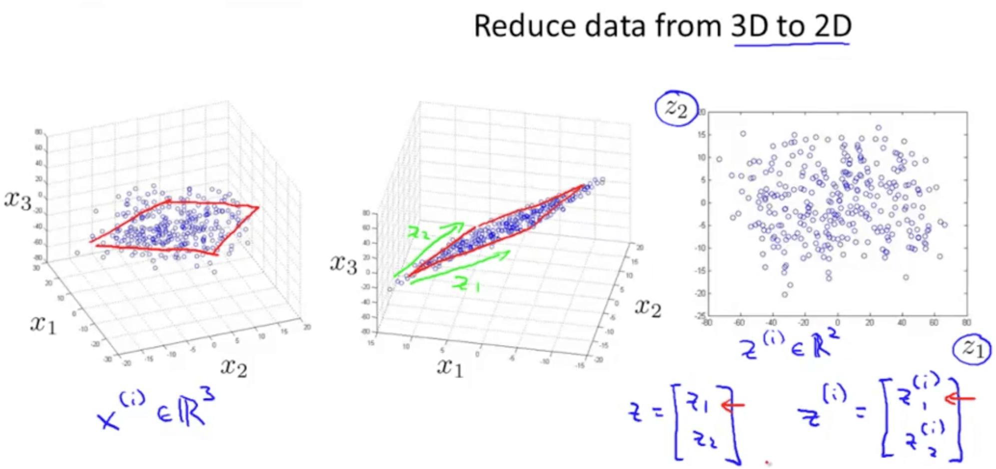
Clustering Algorithms



https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/
https://apandre.wordpress.com/visible-data/cluster-analysis/

Dimensionality Reduction

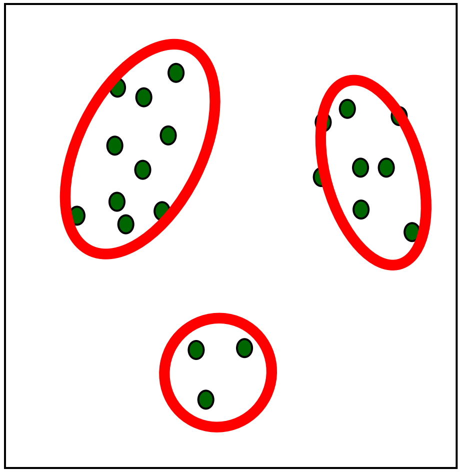




Clustering

Adopted from slides by Alexander Ihler

Unsupervised learning

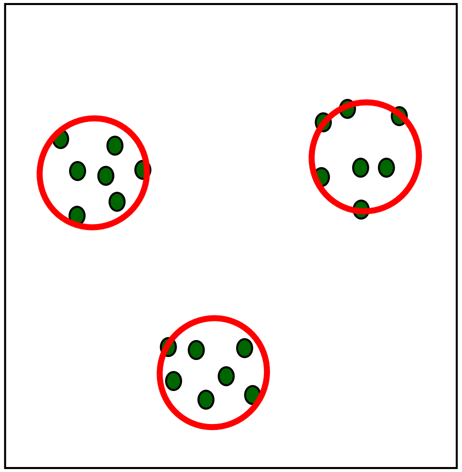


- Supervised learning
  - Predict target value ("y") given features ("x")
- Unsupervised learning
  - Understand patterns of data (just "x")
  - Useful for many reasons
    - Data mining ("explain")
    - Missing features ("impute")
    - Representation (feature generation or selection)
- One example: *clustering*
  - Describe data by discrete "groups" with some characteristics

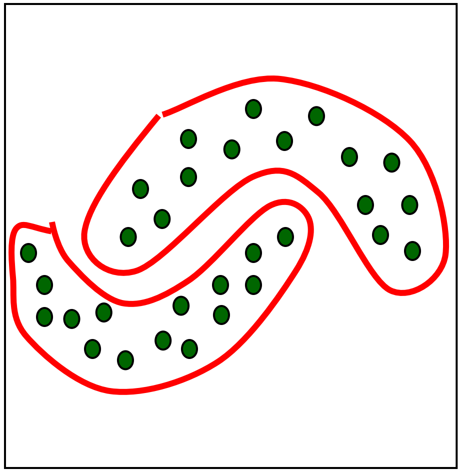
Clustering

- Clustering describes data by "groups"
- The meaning of "groups" may vary by data!

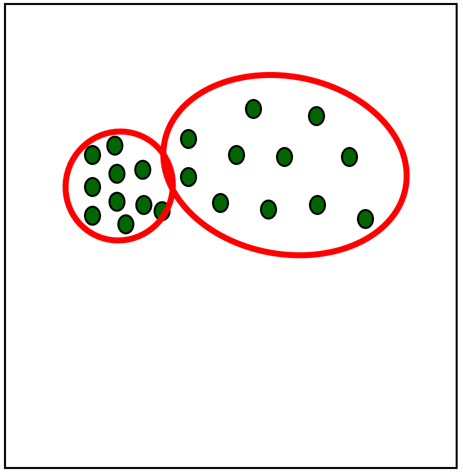
- Examples



Location



Shape



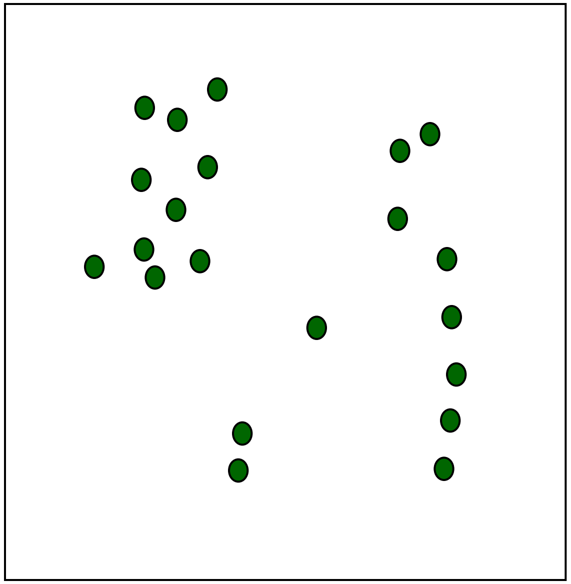
Density

Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering

Initially, every datum is a cluster

Data:



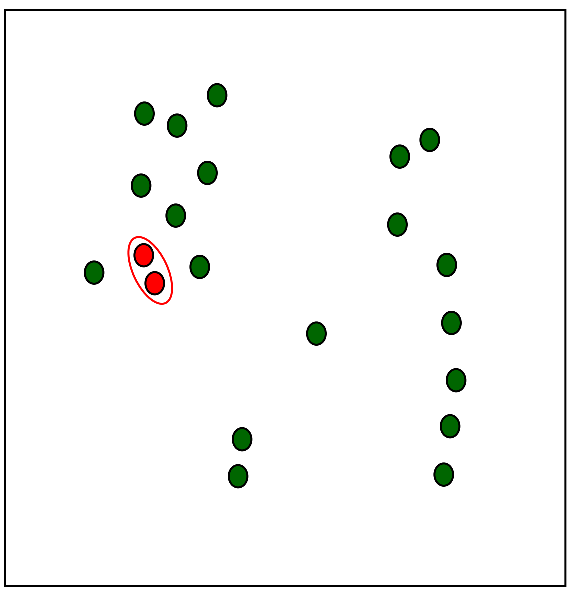
- A simple clustering algorithm
- Define a distance (or dissimilarity) between clusters (we'll return to this)
- Initialize: every example is a cluster
- Iterate:
  - Compute distances between all clusters
    (store for efficiency)
  - Merge two closest clusters
- Save both clustering and *sequence* of cluster operations
- "Dendrogram"

Algorithmic Complexity: $O(m^2 \log m) +$

Iteration 1

Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:

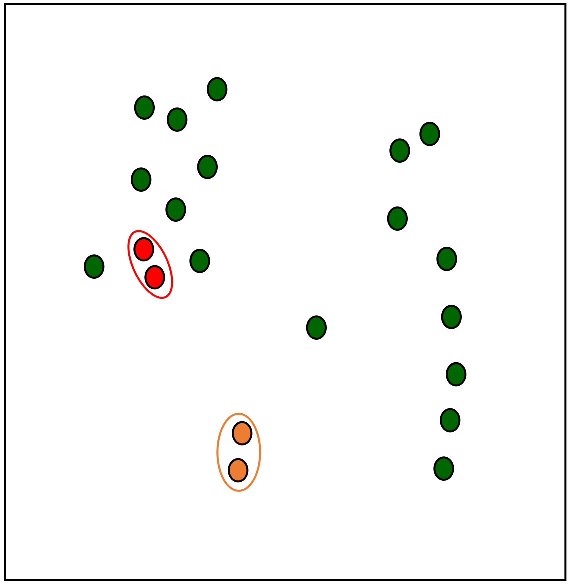
Height of the join
indicates dissimilarity

Algorithmic Complexity: $O(m^2 \log m) + O(m \log m) +$

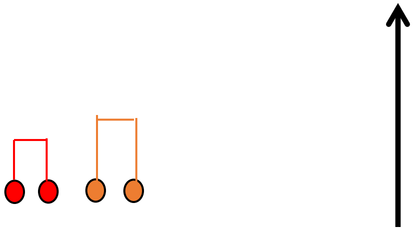
Iteration 2

Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:



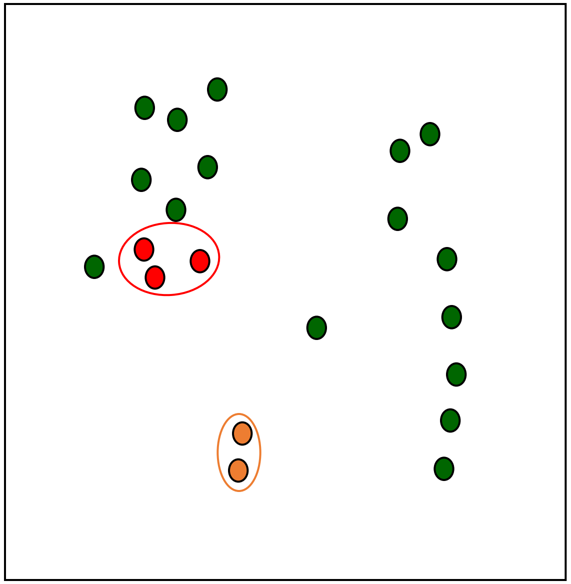
Height of the join indicates dissimilarity

Algorithmic Complexity: $O(m^2 \log m) + 2*O(m \log m) +$

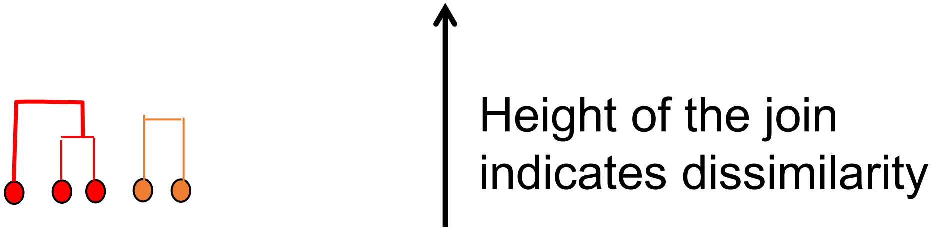
Iteration 3

Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:

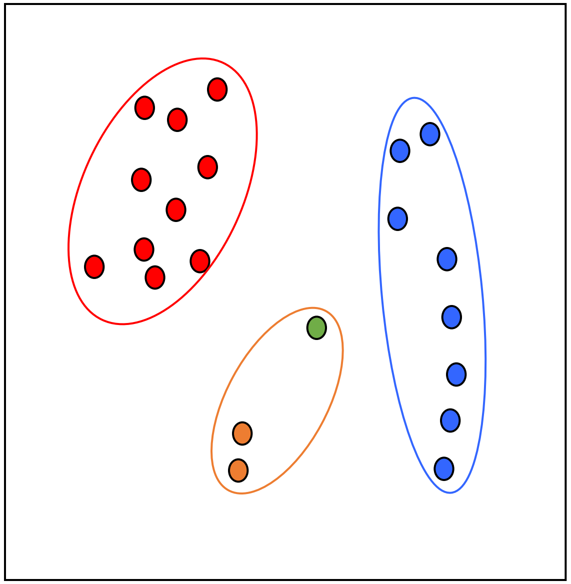


Algorithmic Complexity: $O(m^2 \log m) + 3*O(m \log m) +$

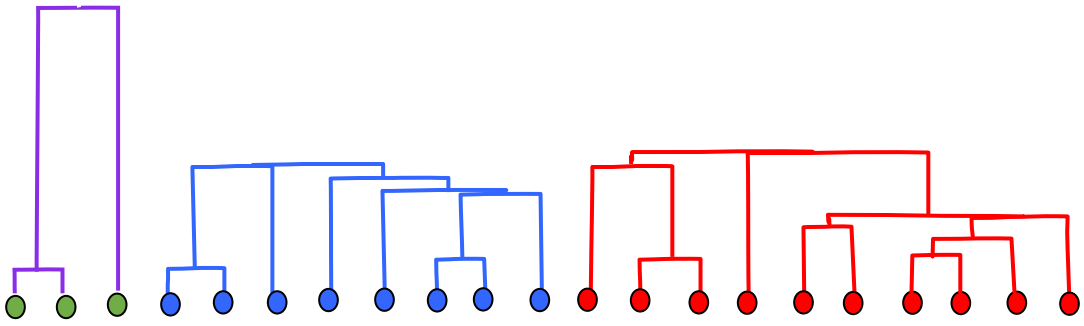
Iteration m-3

Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:

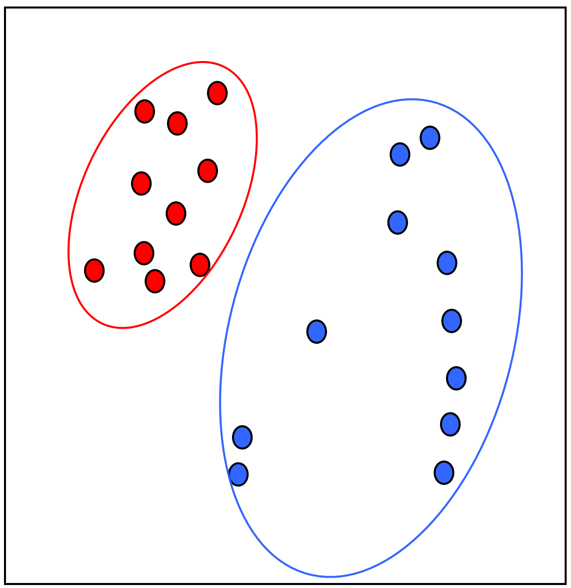


Algorithmic Complexity: $O(m^2 \log m) + (m-3)*O(m \log m) +$

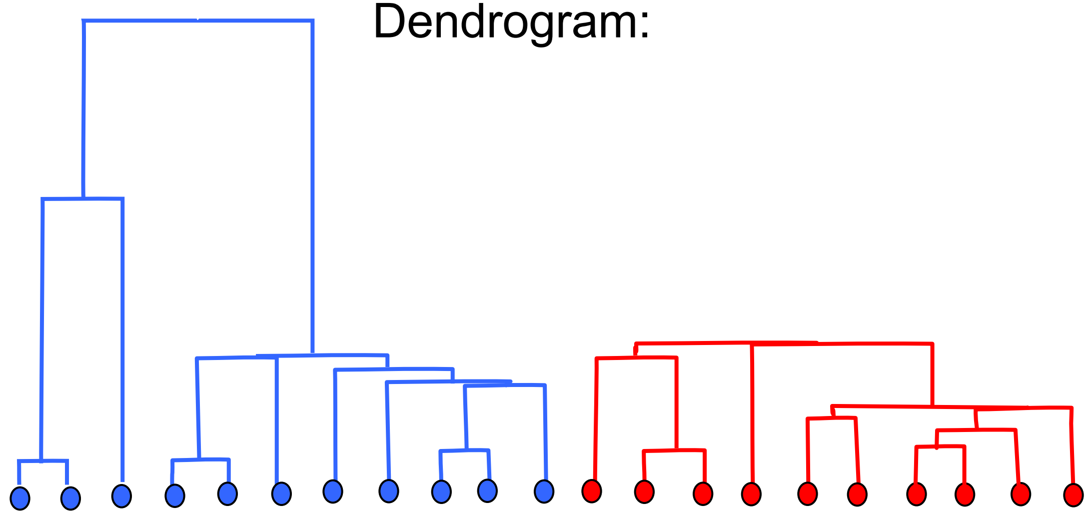
Iteration m-2

Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:

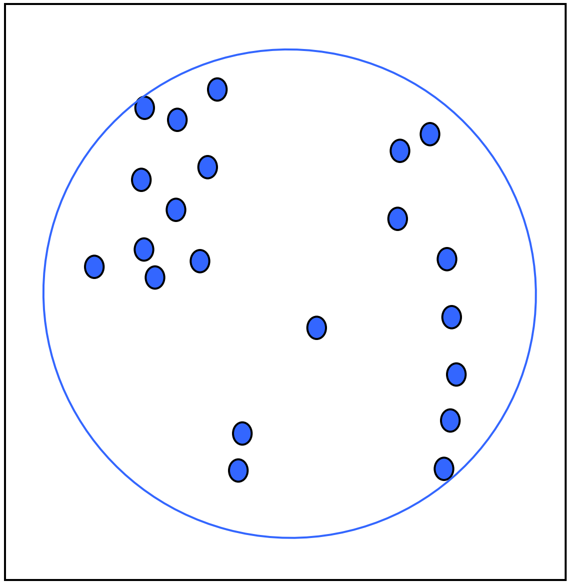


Algorithmic Complexity: $O(m^2 \log m) + (m-2)^*O(m \log m) +$

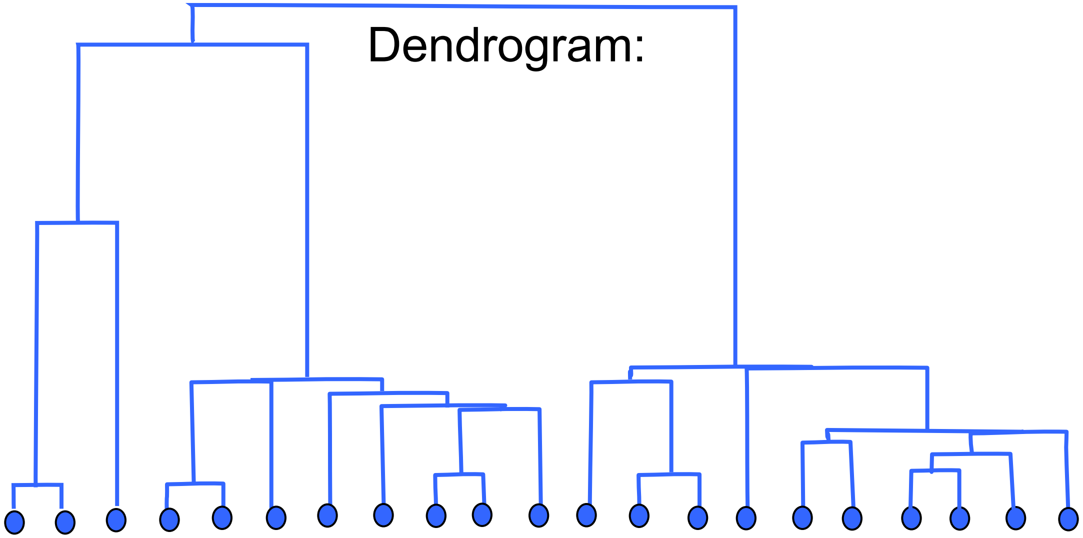
Iteration m-1

Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:

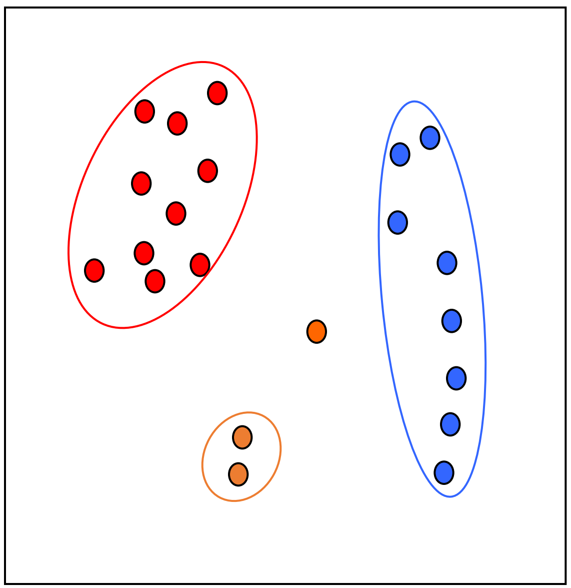


Algorithmic Complexity: $O(m^2 \log m) + (m-1)*O(m \log m) = O(m^2 \log m)$

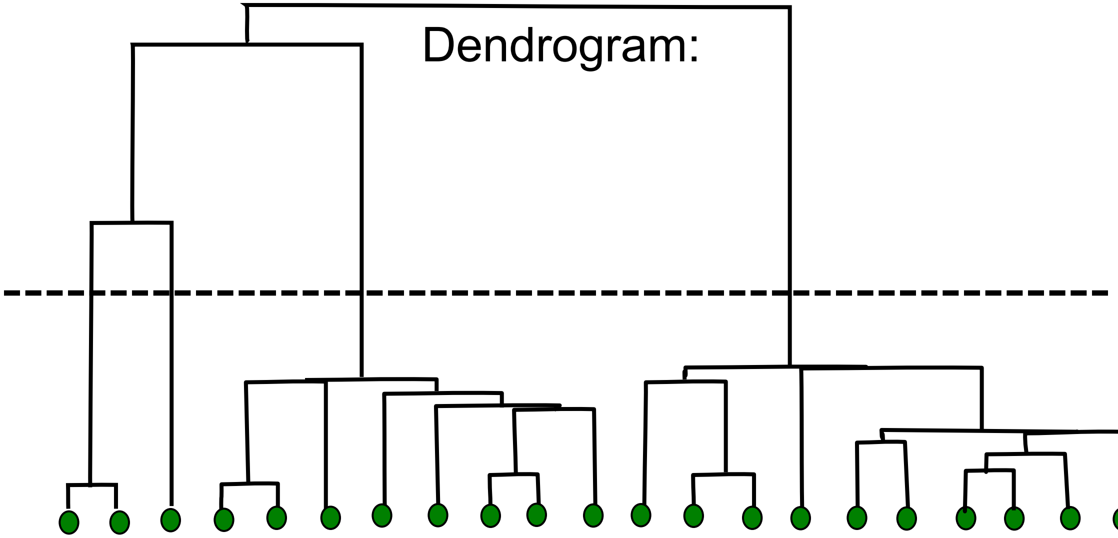
From dendrogram to clusters

Given the sequence, can select a number of clusters or a dissimilarity threshold:

Data:



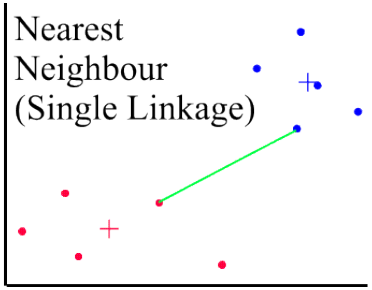
Dendrogram:



Algorithmic Complexity: $O(m^2 \log m) + (m-1)*O(m \log m) = O(m^2 \log m)$

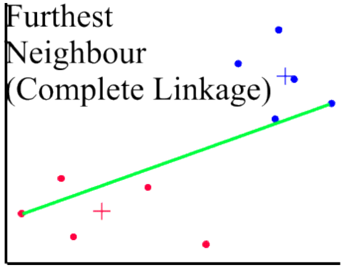
Cluster distances

$$D_{\min}(C_i, C_j) = \min_{x \in C_i,\ y \in C_j} \|x - y\|^2$$



produces minimal spanning tree.

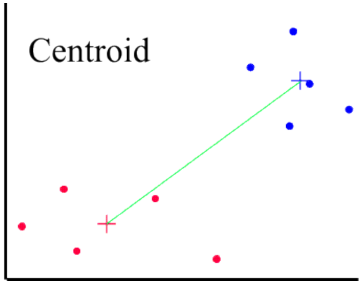
$$D_{\max}(C_i, C_j) = \max_{x \in C_i,\ y \in C_j} \|x - y\|^2$$



avoids elongated clusters.

$$D_{\text{avg}}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i,\ y \in C_j} \|x - y\|^2$$

$$D_{\text{means}}(C_i, C_j) = \|\mu_i - \mu_j\|^2$$



Need:

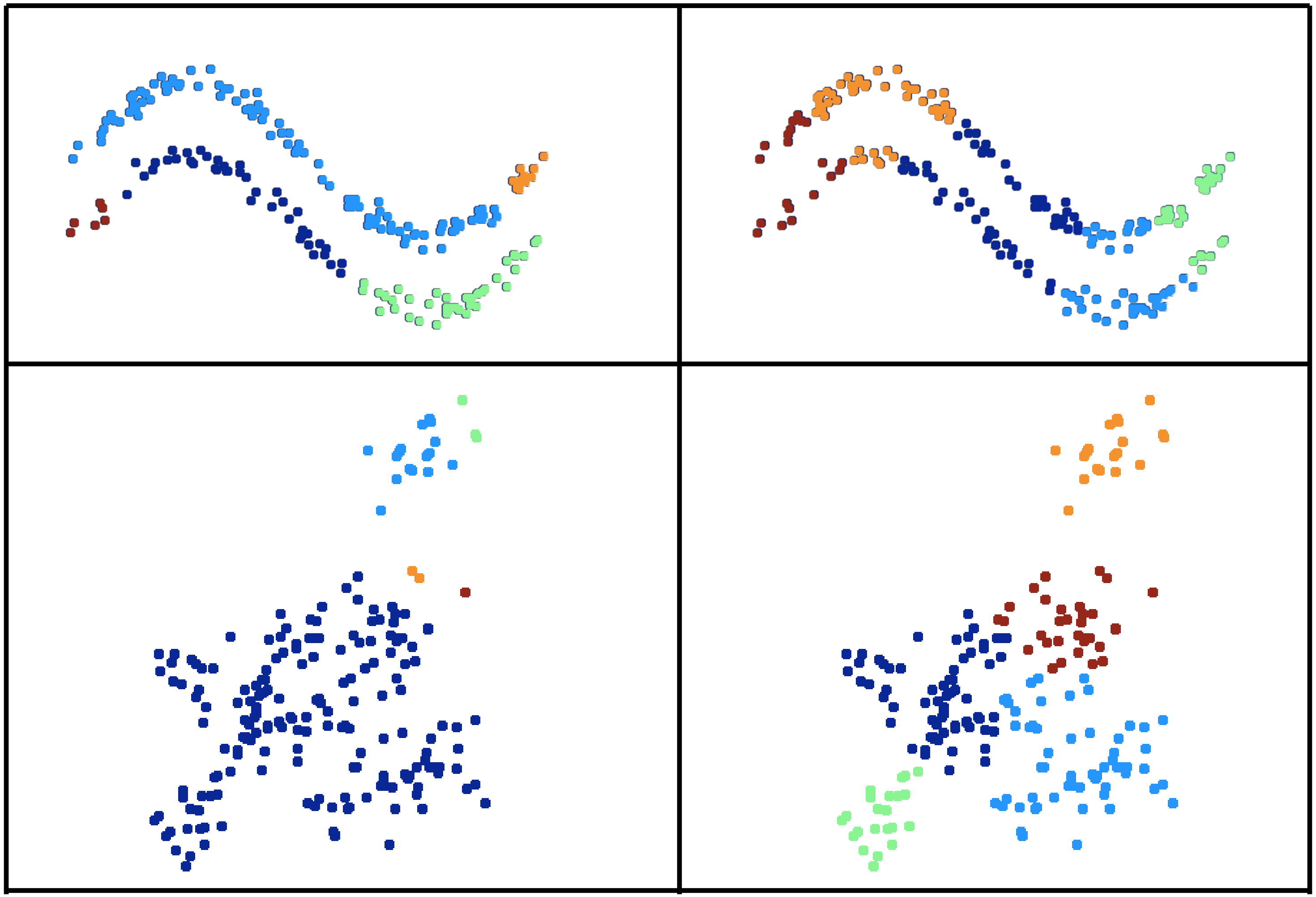
D(A,C) → D(A+B,C)

D(B,C) →

Cluster distances

- Dissimilarity choice will affect clusters created

Single linkage (min) | Complete linkage (max)



Summary

- Agglomerative clustering
  - Choose a cluster distance / dissimilarity scoring method
  - Successively merge closest pair of clusters
  - “Dendrogram” shows sequence of merges & distances
  - Complexity: $O(m^2 \log m)$

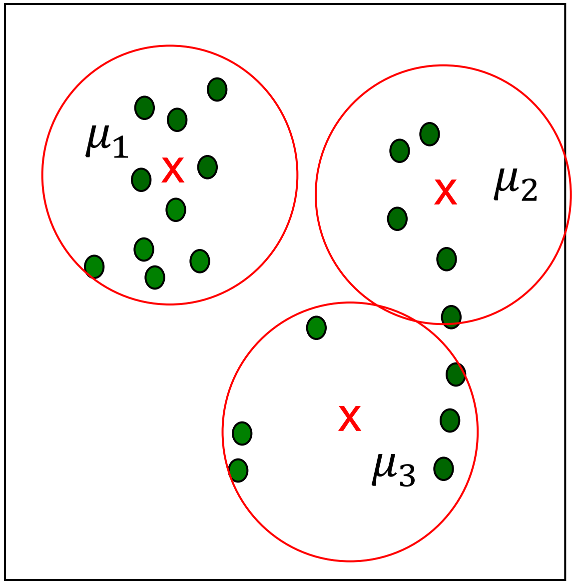
- “Clustergram” for understanding data matrix
  - Build clusters on rows (data) and columns (features)
  - Reorder data & features to expose behavior across groups

- Agglomerative clusters depend critically on dissimilarity
  - Choice determines characteristics of “found” clusters

k-Means Clustering

K-Means Clustering

- A simple clustering algorithm
- Iterate between
  - Updating the assignment of data to clusters
  - Updating the cluster's summarization



Notation:

Data example i has features x_i

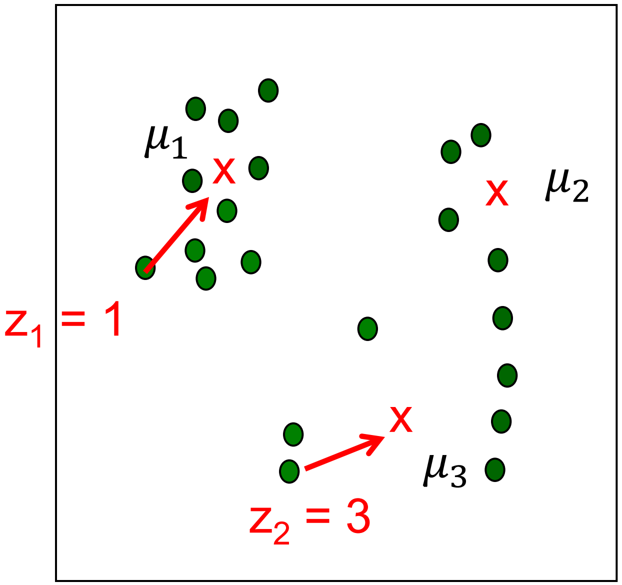
Assume K clusters

Each cluster c "described" by a center μ_c

Each cluster will "claim" a set of nearby points

K-Means Clustering

- A simple clustering algorithm
- Iterate between
  - Updating the assignment of data to clusters
  - Updating the cluster's summarization



Notation:

Data example i has features x_i

Assume K clusters

Each cluster c "described" by a center μ_c

Each cluster will "claim" a set of nearby points

"Assignment" of ith example: $z_i \in 1..K$

K-Means Clustering

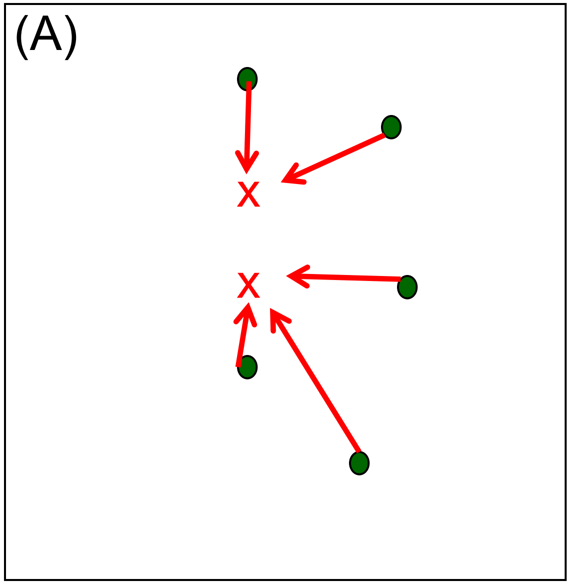
- Iterate until convergence:
  - (A) For each datum, find the closest cluster

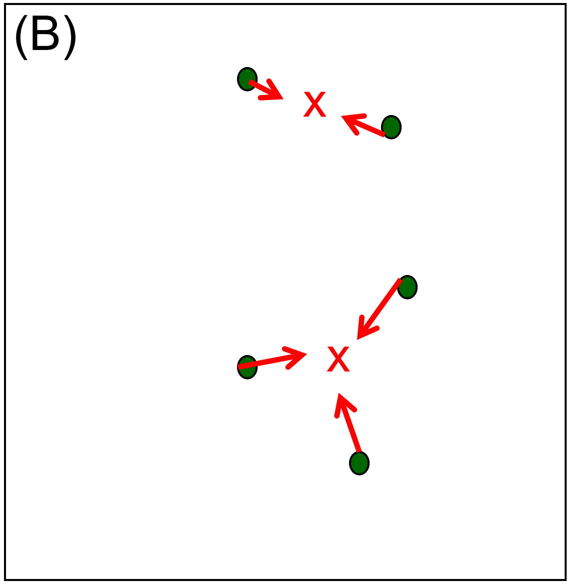
$$z_i = \arg\min_c \|x_i - \mu_c\|^2 \qquad \forall i$$

  - (B) Set each cluster to the mean of all assigned data:

$$\forall c, \qquad \mu_c = \arg\min_{\mu_c} \sum_{i:z_i=c} \|x_i - \mu_c\|^2$$

$$\forall c, \qquad \mu_c = \frac{1}{m_c} \sum_{i \in S_c} x_i \qquad\qquad S_c = \{i : z_i = c\},\ m_c = |S_c|$$





K-Means Clustering

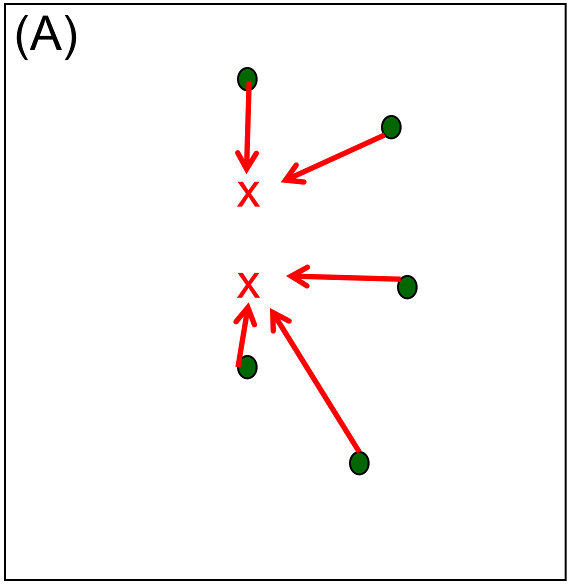
- Optimizing the cost function:

$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$

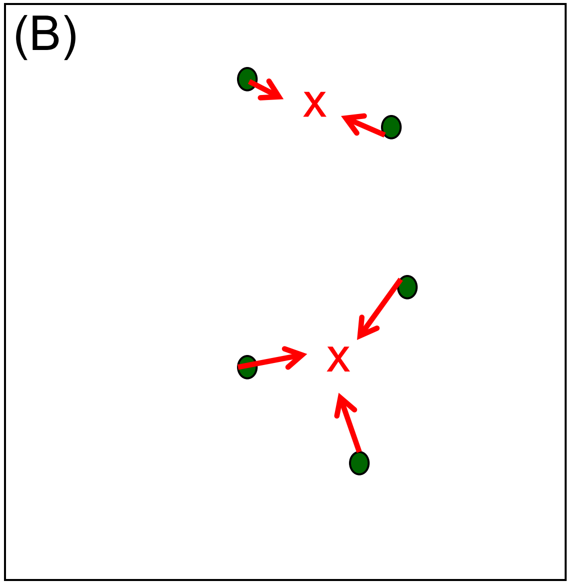
- Alternating optimization:

Descent => guaranteed to converge
New means = same assignments
Same assignments = same means
Same means = same assignments
…

Over the cluster assignments:
Only one term in sum depends on z_i
Minimized by selecting closest μ_c



Over the cluster centers:
Cluster c only depends on x_i with z_i=c
Minimized by selecting the mean



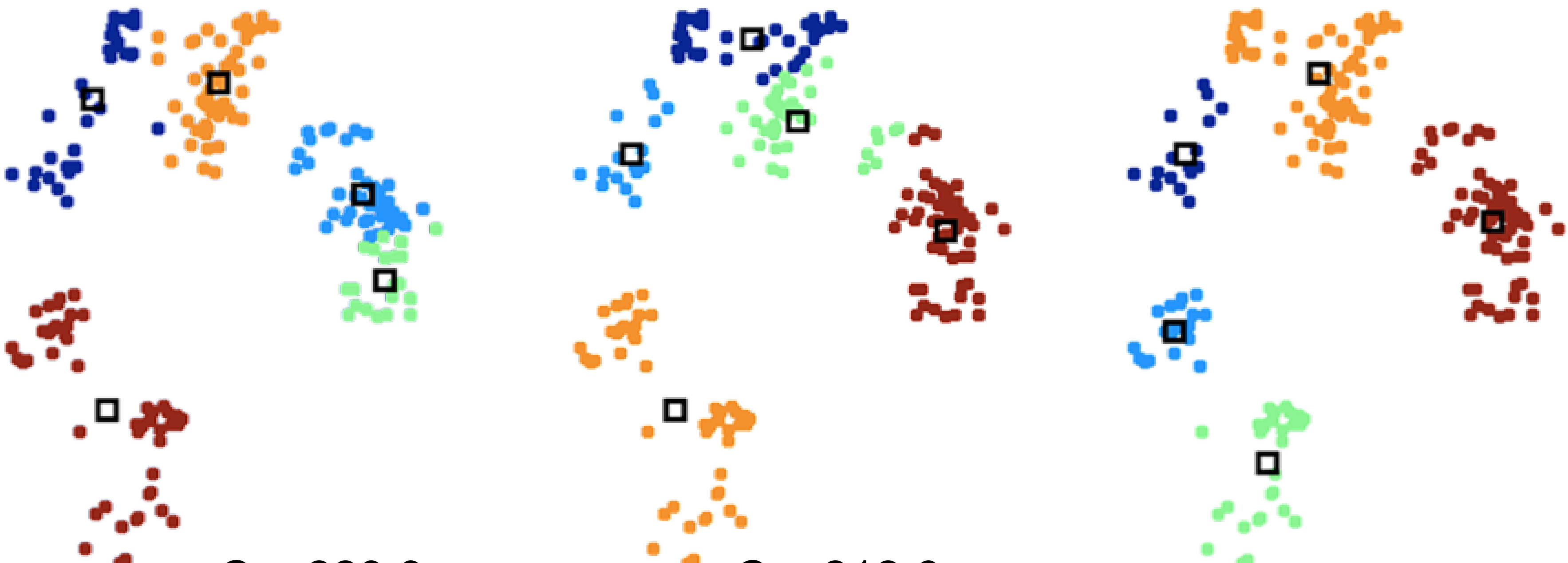
K-Means Clustering

- **1)** Initialize clusters centroids. For each point, place it in the cluster whose current centroid it is nearest
- **2)** After all points are assigned, update the locations of centroids of the ***k*** clusters
- **3)** Reassign all points to their closest centroid
  - Sometimes moves points between clusters
- **Repeat 2 and 3 until convergence**
  - **Convergence:** Points don't move between clusters and centroids stabilize

Initialization

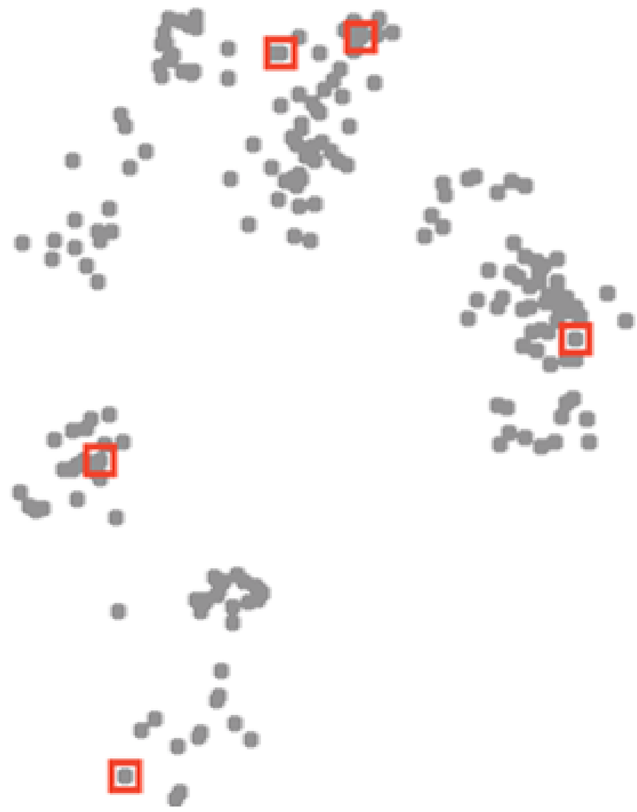
- Multiple local optima, depending on initialization

- Try different (randomized) initializations
- Can use cost C to decide which we prefer



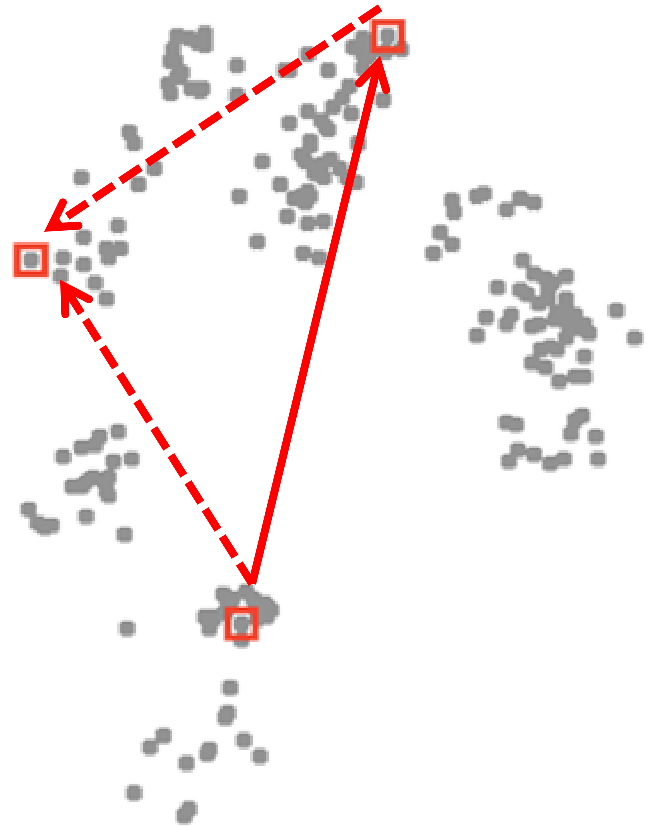
C = 223.3 C = 212.6 C = 167.0

Initialization methods



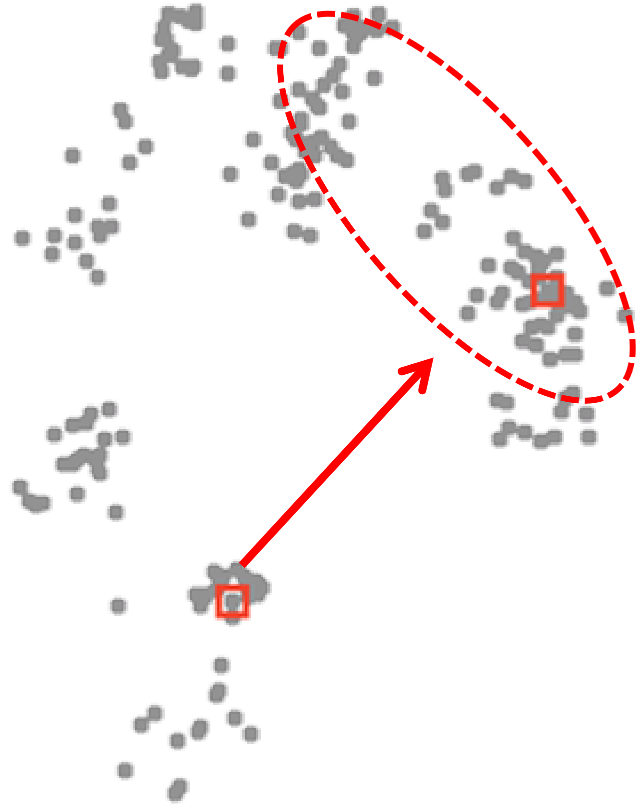
- Random
  - Usually, choose random data index
  - Ensures centers are near some data
  - Issue: may choose nearby points

Initialization methods



- Random
  - Usually, choose random data index
  - Ensures centers are near some data
  - Issue: may choose nearby points
- Distance-based
  - Start with one random data point
  - Find the point farthest from the clusters chosen so far
  - Issue: may choose outliers

Initialization methods

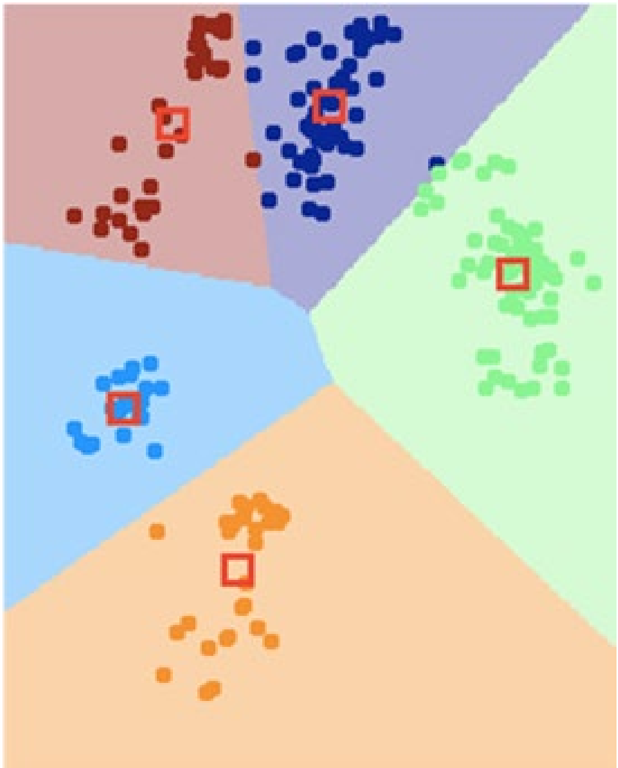


- Random
  - Usually, choose random data index
  - Ensures centers are near some data
  - Issue: may choose nearby points
- Distance-based
  - Start with one random data point
  - Find the point farthest from the clusters chosen so far
  - Issue: may choose outliers
- Random + distance ("k-means++") (Arthur & Vassilvitskii, 2007)
  - Choose next points "far but randomly"

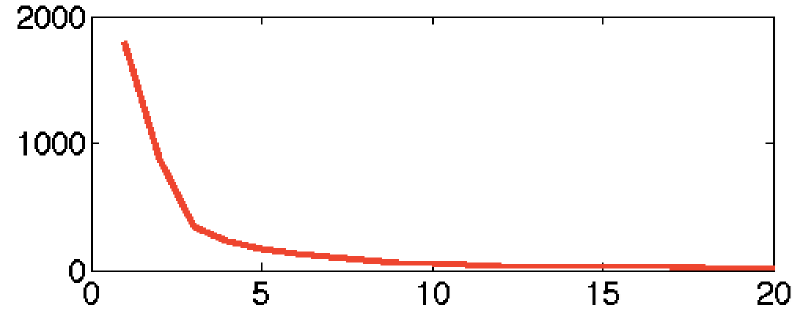
    p(x) / squared distance from x to current centers
  - Likely to put a cluster far away, in a region with lots of data

Out-of-sample points

- Often want to use clustering on new data
- Easy for k-means: choose nearest cluster center



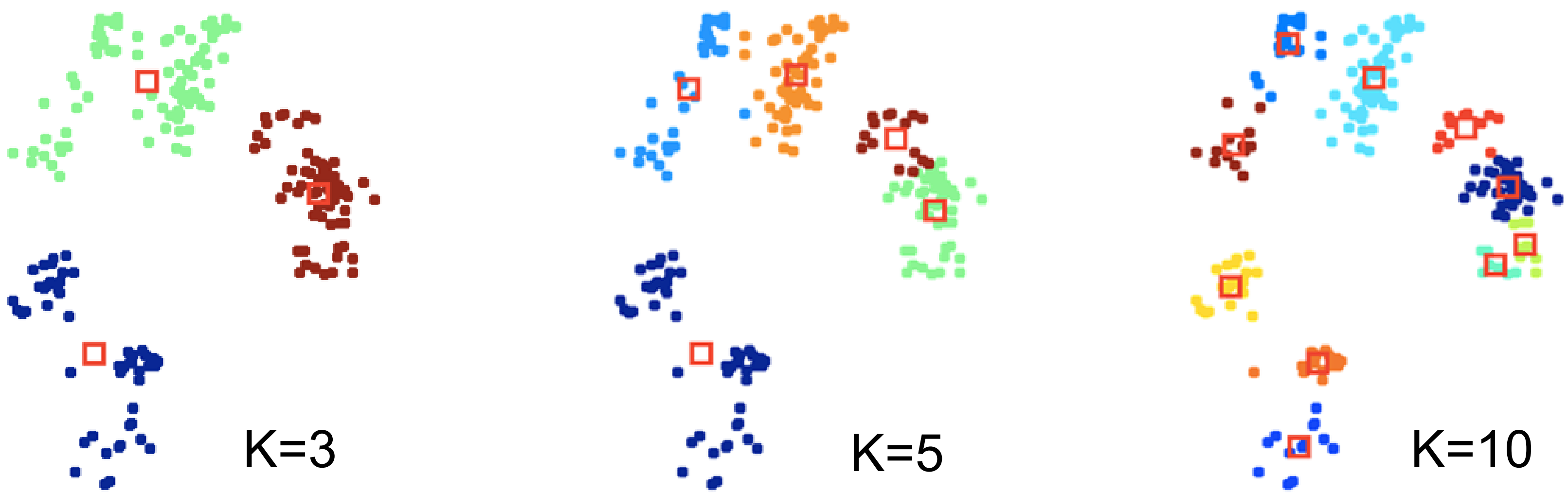
Choosing the number of clusters



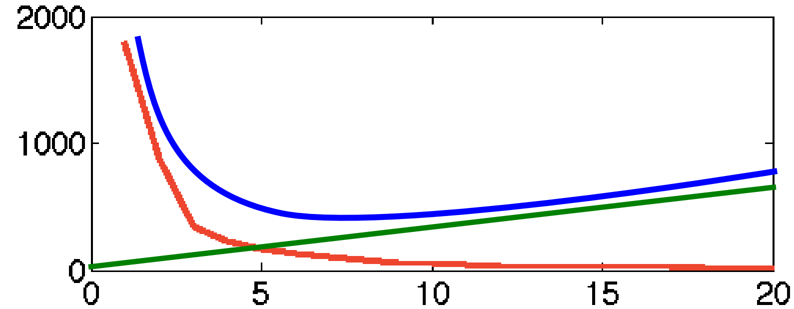
- With cost function

$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$

  what is the optimal value of k?
- Cost always decreases with k!
- A model complexity issue…



Choosing the number of clusters



- With cost function

$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$

  what is the optimal value of k?
- Cost always decreases with k!
- A model complexity issue…

- One solution is to penalize for complexity
  - Add penalty: Total = Error + Complexity
  - Now more clusters can increase cost, if they don't help "enough"
  - Ex: simplified Bayesian Information Criterion (BIC) penalty

$$J(\underline{z}, \underline{\mu}) = \log \left[\frac{1}{m\,d} \sum_i \|x_i - \mu_{z_i}\|^2 \right] + k \frac{\log m}{m}$$

Summary

- K-Means clustering
  - Clusters described as locations (“centers”) in feature space
- Procedure
  - Initialize cluster centers
  - Iterate: assign each data point to its closest cluster center
  - : move cluster centers to minimize mean squared error
- Properties
  - Coordinate descent on MSE criterion
  - Prone to local optima; initialization important
- Out-of-sample data
- Choosing the # of clusters, K
  - Model selection problem; penalize for complexity (BIC, etc.)