GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LECTURE 12: K-MEANS AND HIERARCHICAL CLUSTERING

January 14, 2015

DAT11-SF

LAST TIME:

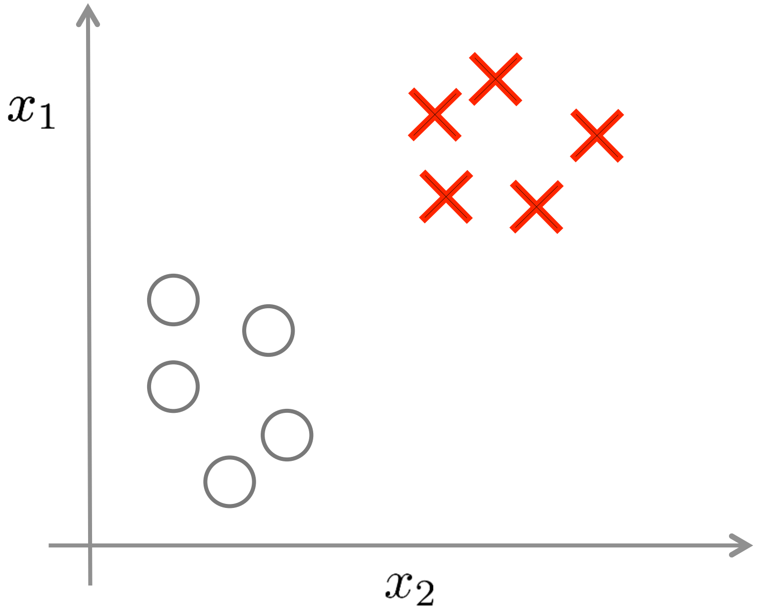
- NAÏVE BAYES CLASSIFIER
- ROC CURVES

QUESTIONS?

I. CLUSTER ANALYSIS
II. K-MEANS CLUSTERING
III. HIERARCHICAL CLUSTERING

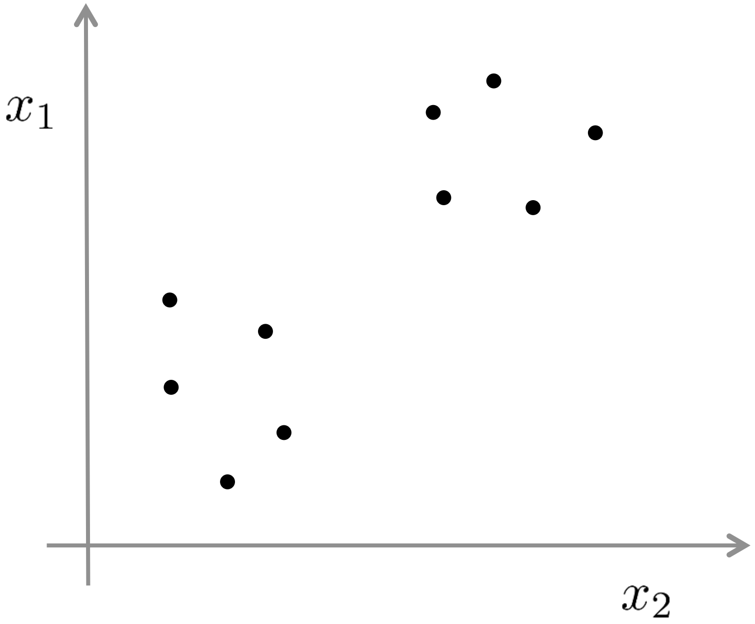
LABS:
IV. CLUSTERING IN SCIKIT-LEARN

I. CLUSTER ANALYSIS



Supervised: every training data point has features X=(x1,x2) and class label Y.

Goal: predict Y when it is not known



Unsupervised: every data point has features X =(x1,x2), unlabeled data

Goal: discover structure in data

Clustering – set of methods for partitioning data into subgroups (clusters)
Cluster – a group of **similar** data points.

The concept of similarity is central to the definition of a cluster, and therefore to cluster analysis.

SIMILARITY / AFFINITY

Euclidian distance (dissimilarity):

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

Cosine similarity:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum\limits_{i=1}^{n} A_i \times B_i}{\sqrt{\sum\limits_{i=1}^{n} (A_i)^2} \times \sqrt{\sum\limits_{i=1}^{n} (B_i)^2}}$$

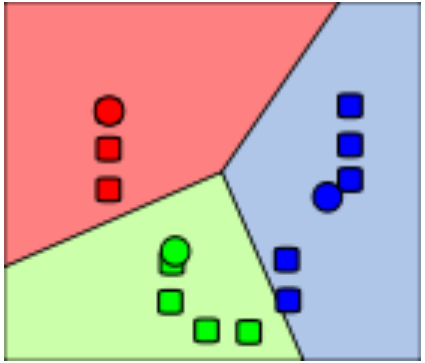
Jaccard similarity:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

Construct customized distance matrix - D

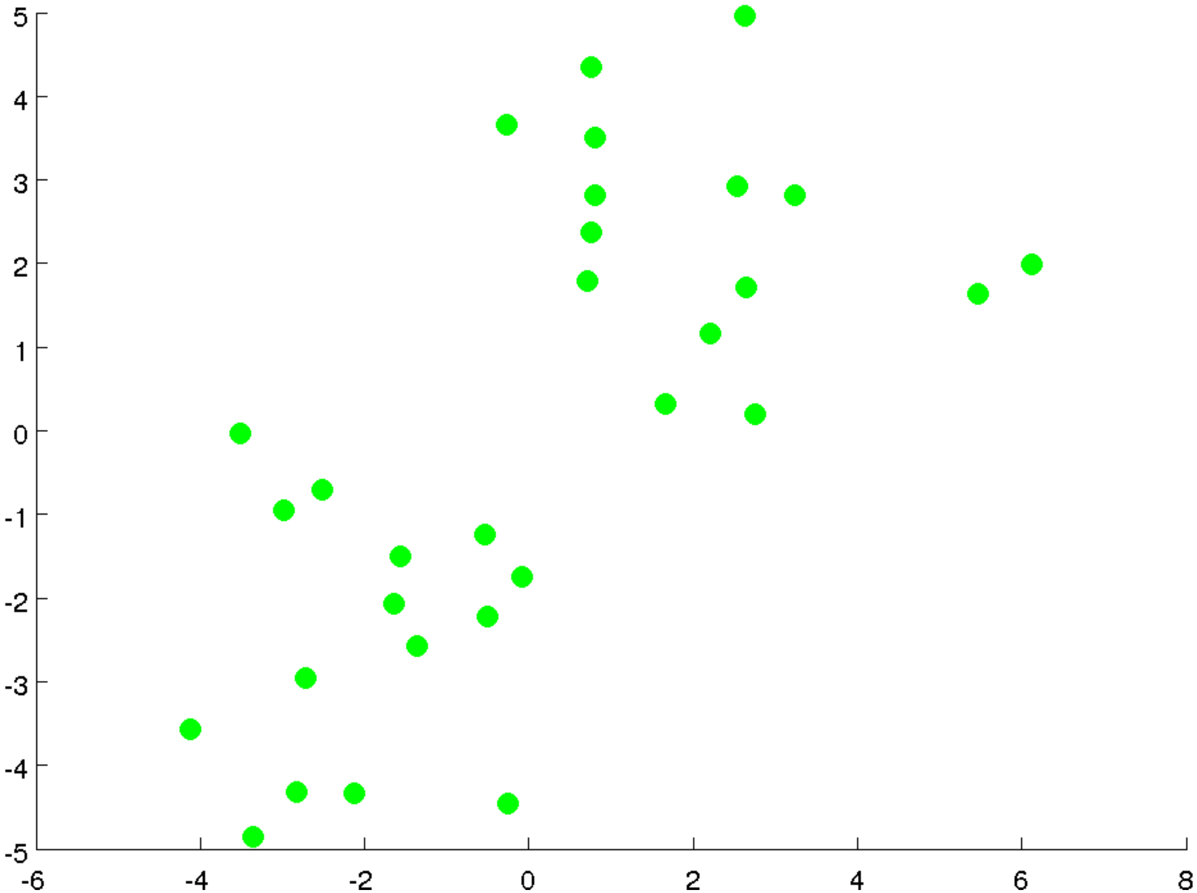
II. K-MEANS CLUSTERING

K-means clustering is an iterative algorithm that partitions observations (data points) into k clusters, where every point belongs to the cluster with the nearest mean value

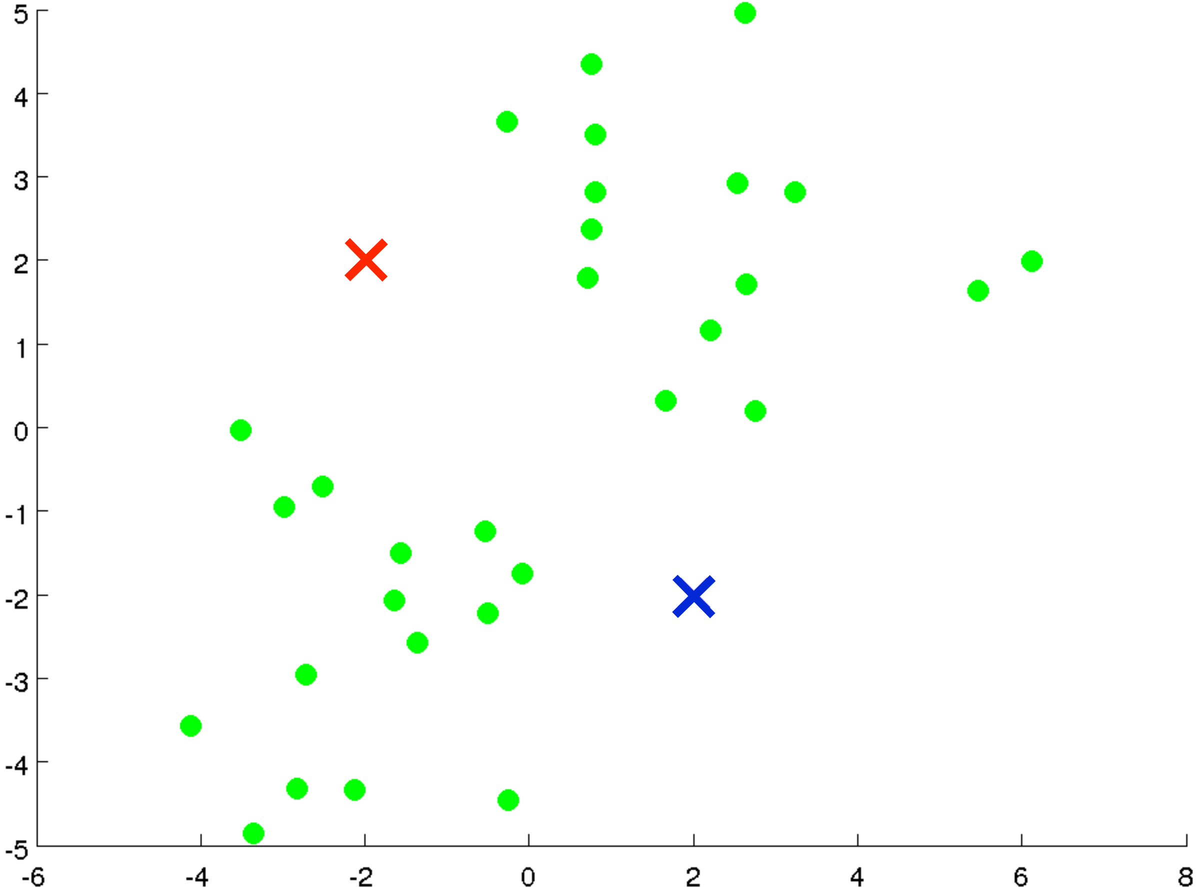


Optimal partitioning NP hard, k-means is greedy algorithm

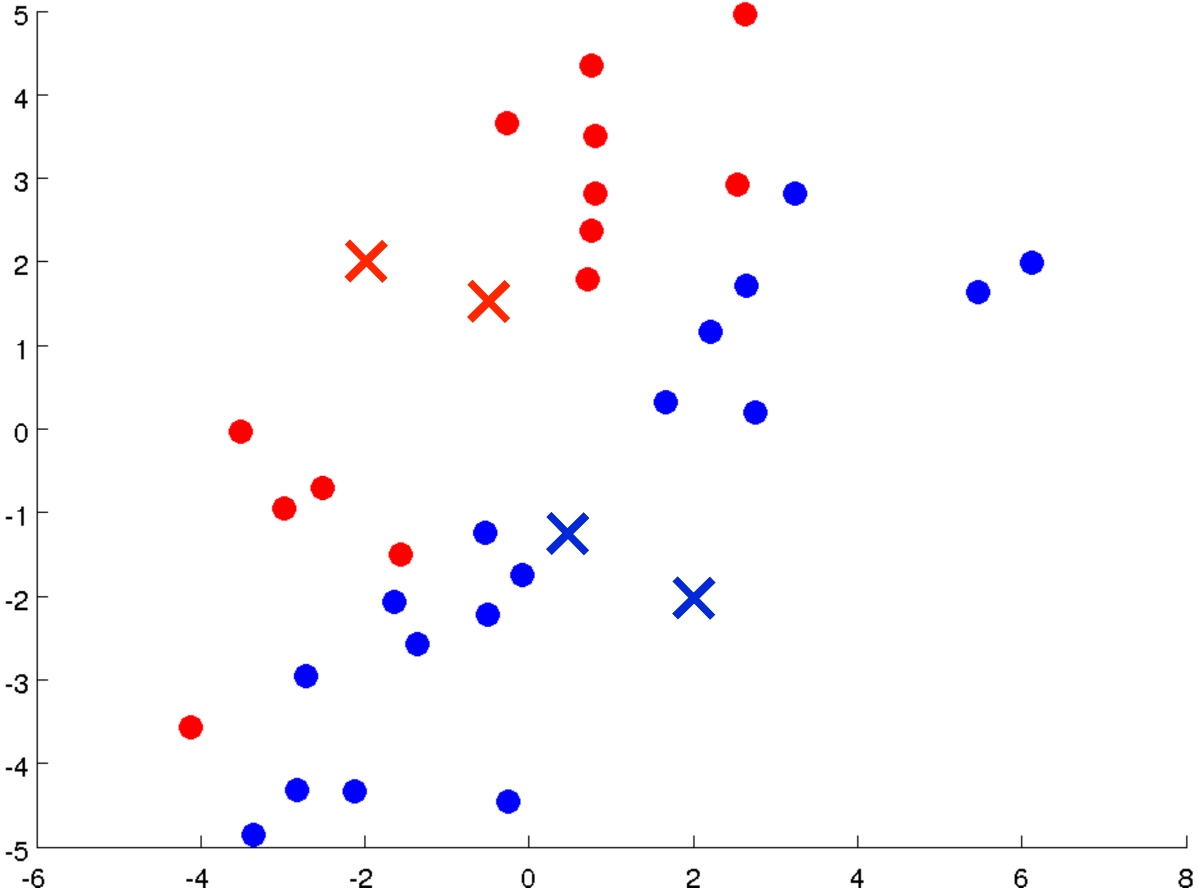
K-MEANS



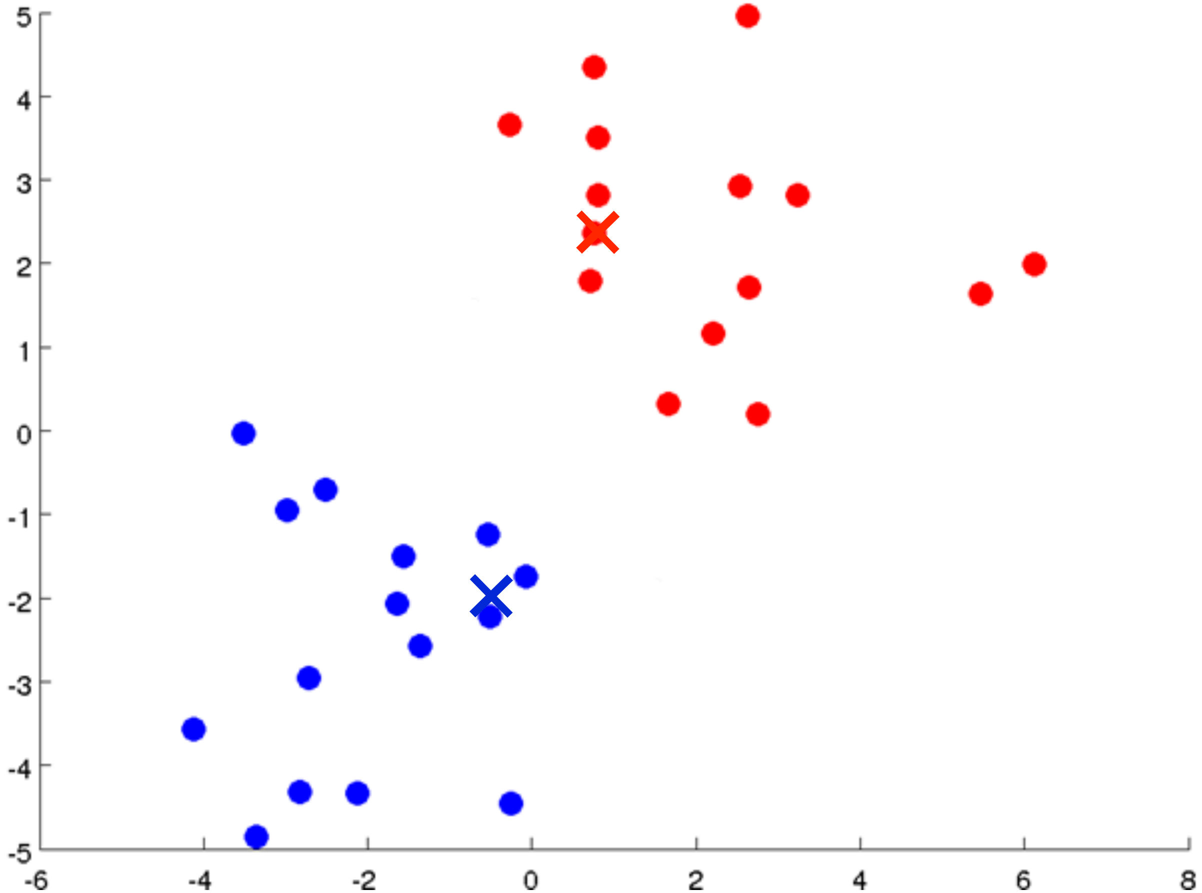
K-MEANS



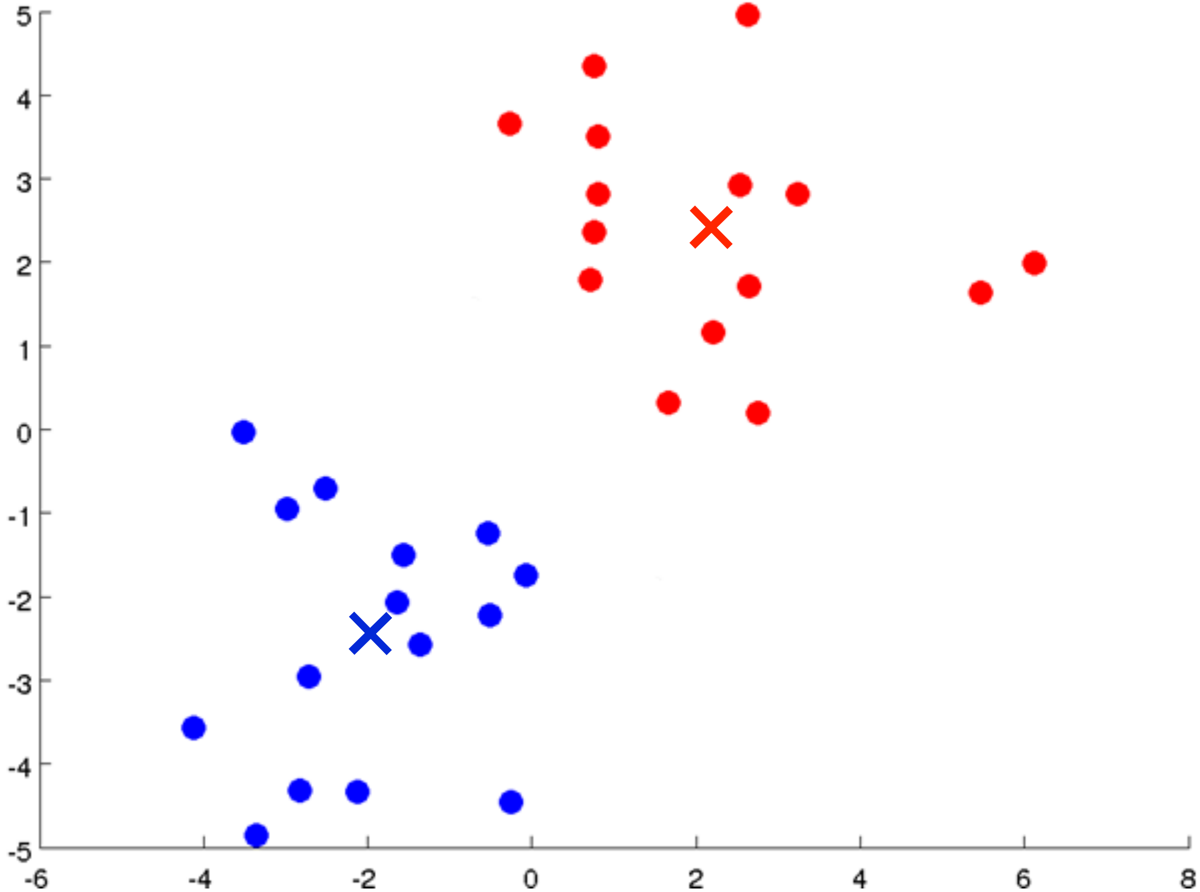
K-MEANS



K-MEANS



K-MEANS



THE BASIC K-MEANS ALGORITHM

1) choose k initial centroids (note that k is an input)

2) for each point:
   - find distance to each centroid
   - assign point to nearest centroid

3) recalculate centroid positions
4) repeat steps 2-3 until stopping criteria met

Optimization objective:

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

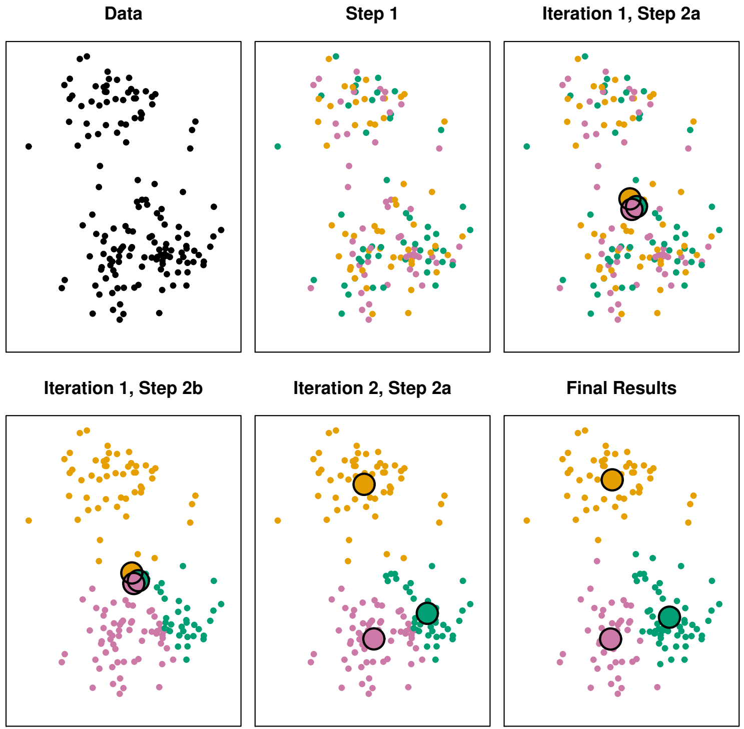
$$\min_{\substack{c^{(1)},\ldots,c^{(m)},\\ \mu_1, \ldots, \mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

$c^{(i)}$ - cluster index

μ_k - cluster centroid

$\mu_{c^{(i)}}$- assigned centroid

K-MEANS


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

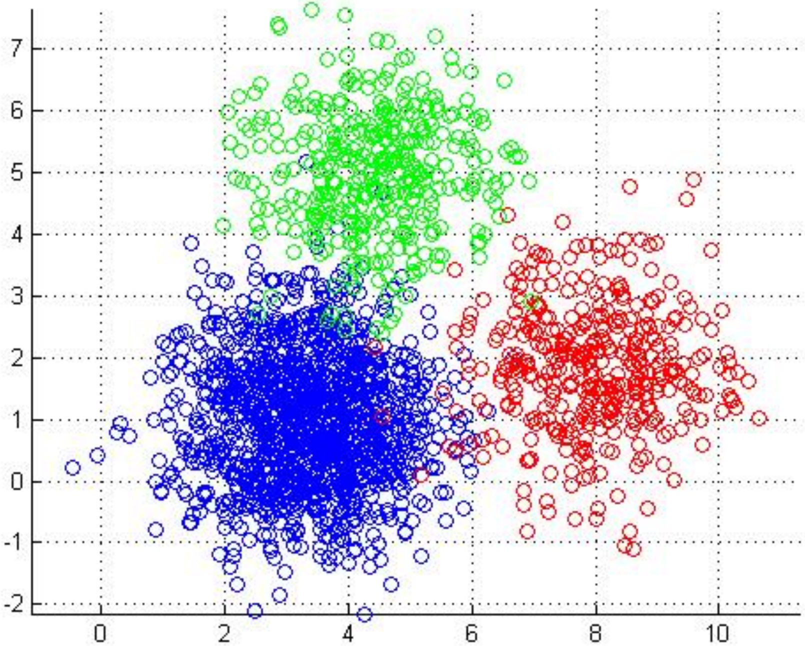
STRENGTHS & WEAKNESSES

K-means is algorithmically pretty efficient (time & space complexity is linear in number of records).

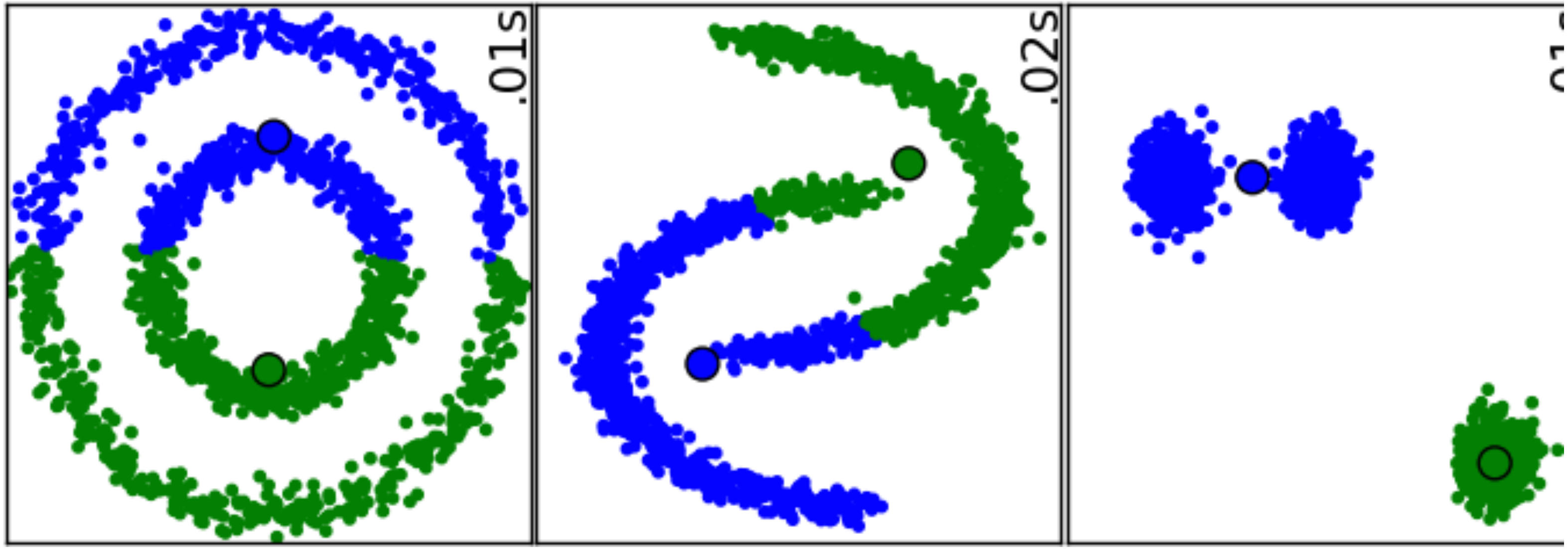
It has a hard time dealing with *non-convex* clusters, or data with widely varying shapes and densities.

Difficulties can sometimes be overcome by increasing the value of k and combining subclusters in a post-processing step.

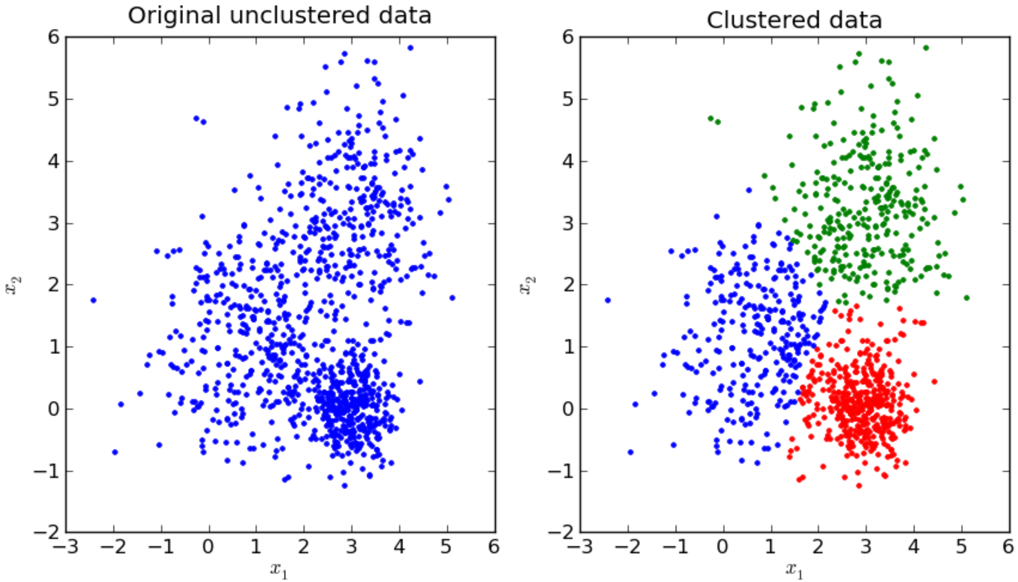
STRENGTHS & WEAKNESSES



STRENGTHS & WEAKNESSES



In general, k-means will converge to a solution and return a partition of k clusters, even if no natural clusters exist in the data.



SCALE DEPENDENCE

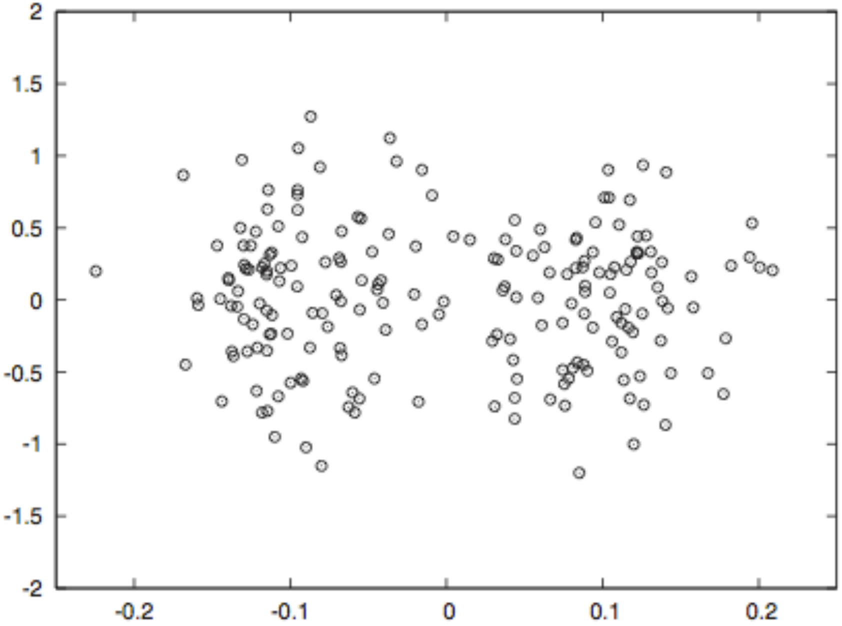
One important point to keep in mind is that partitions are not scale-invariant!

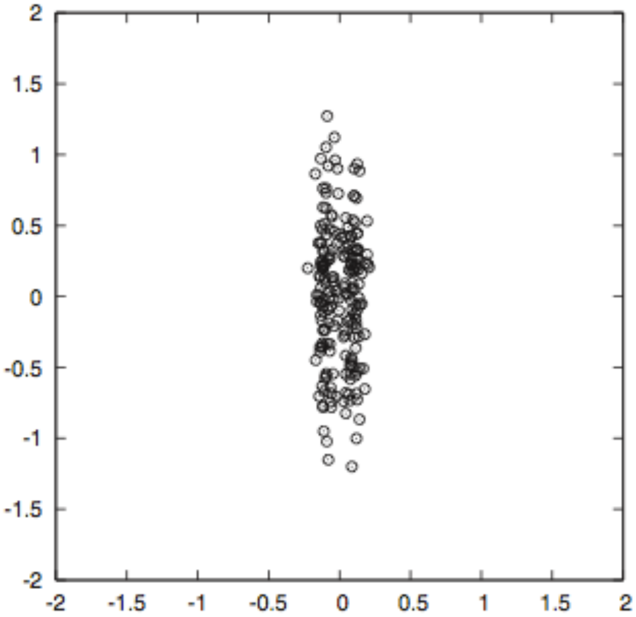
This means that the same data can yield very different clustering results depending on the scale and the units used.

Therefore it’s important to think about your data representation before applying a clustering algorithm.

SCALE DEPENDENCE

These graphs show two different representations of the same data:





source: Data Analysis with Open Source Tools, by Philipp K. Janert. O'Reilly Media, 2011.

INITIATING K-MEANS

There are several options for choosing initial centroids:

- randomly pick k points, make them centroids
- randomly assign every point to one of k clusters
- use some external knowledge

We iterate until some stopping criteria are met; in general, suitable convergence is achieved in a small number of steps.

Stopping criteria can be based on the centroids (eg, if positions change by no more than ε) or on the points (eg, if no more than $x\%$ change clusters between iterations).

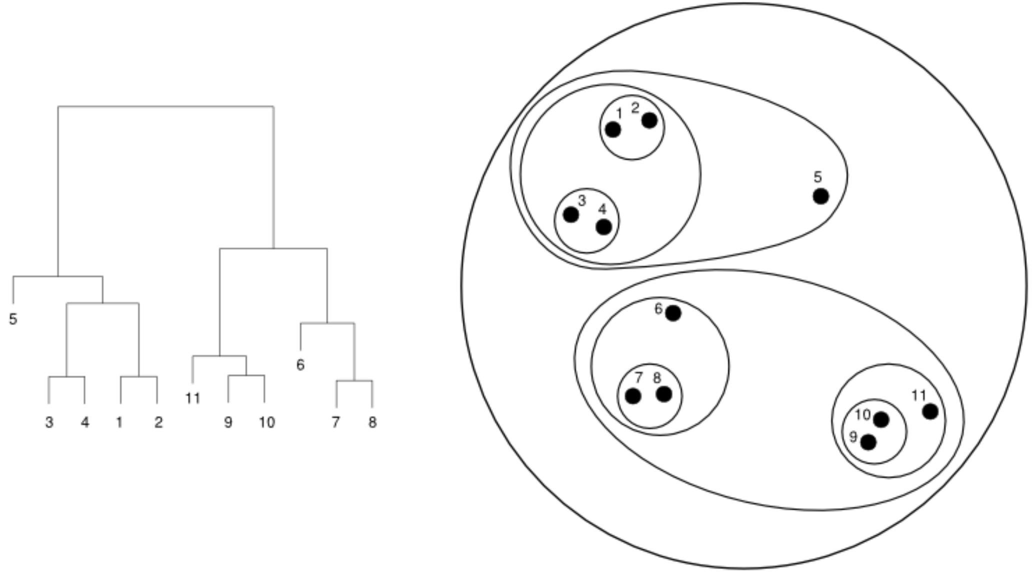
Recall that, in general, different runs of the algorithm will converge to different local optima (centroid configurations).

II. HIERARCHICAL CLUSTERING

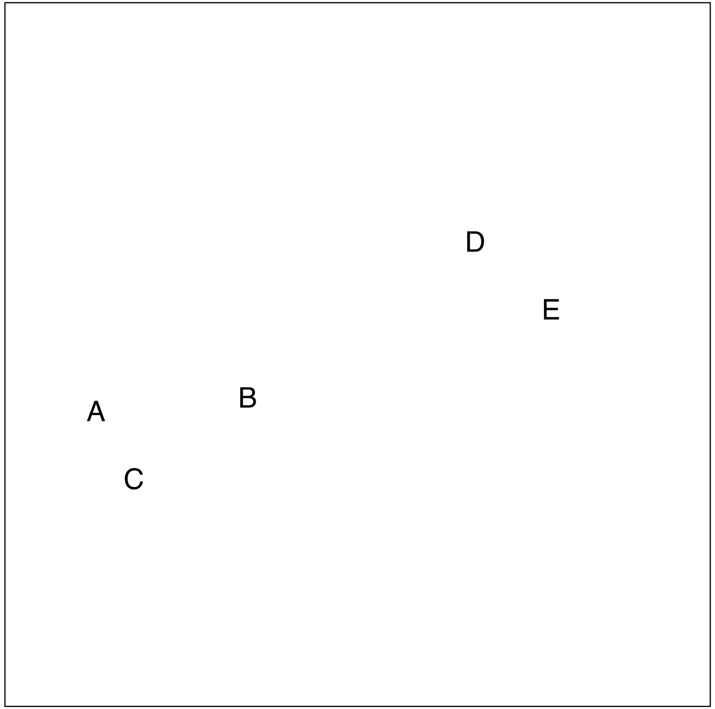
HIERARCHICAL CLUSTERING

Hierarchical clustering builds a hierarchy of clusters:

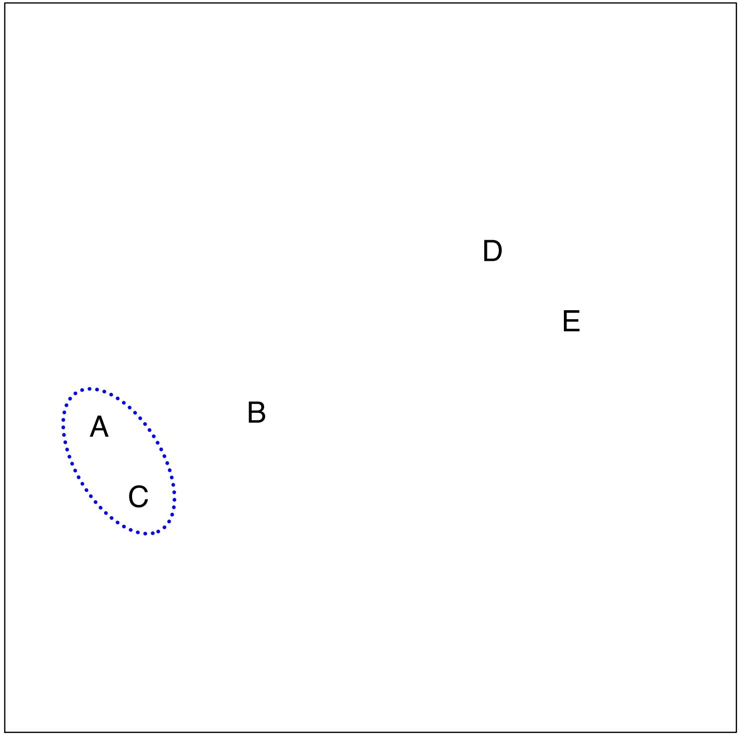
- Divisive, top-down approach, recursively splitting the data
- Agglomerative, bottom-up, iteratively merging data



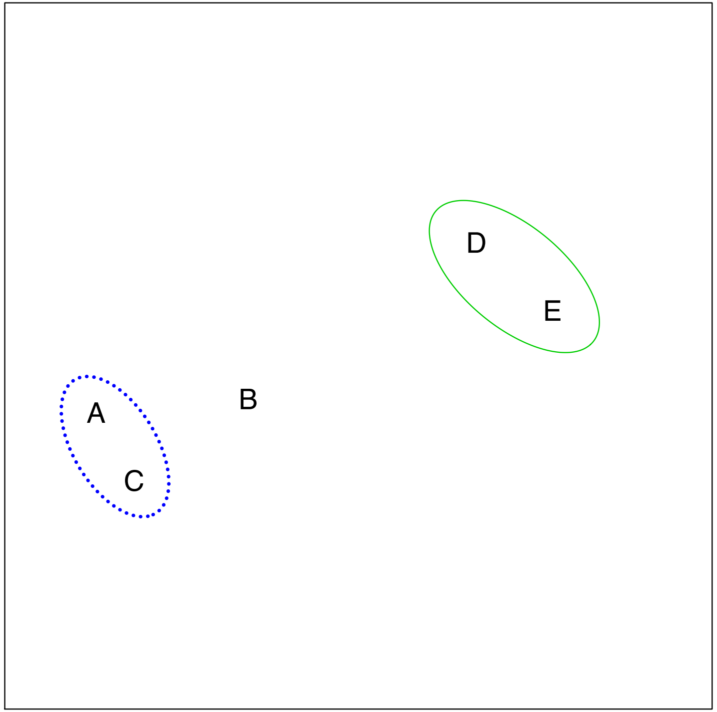
AGGLOMERATIVE CLUSTERING



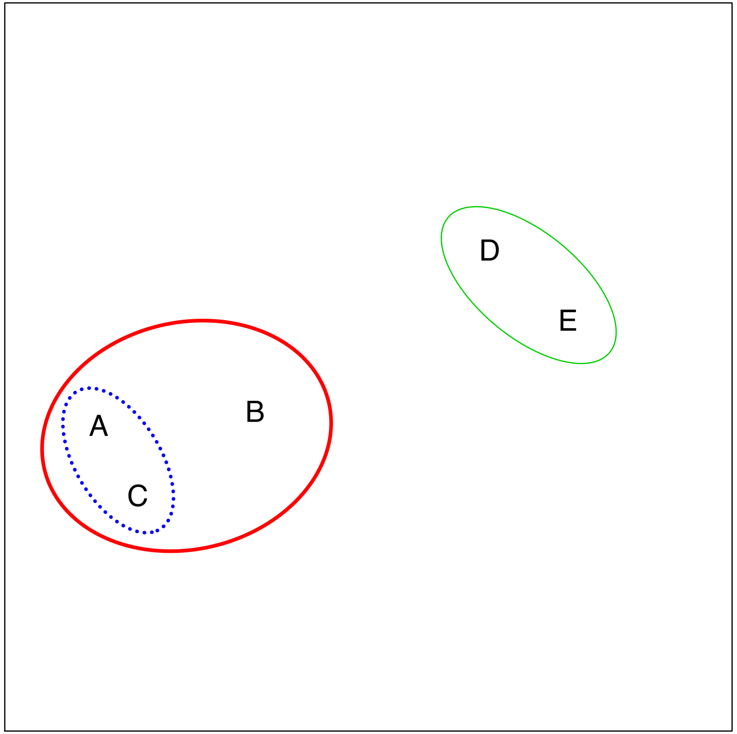
AGGLOMERATIVE CLUSTERING



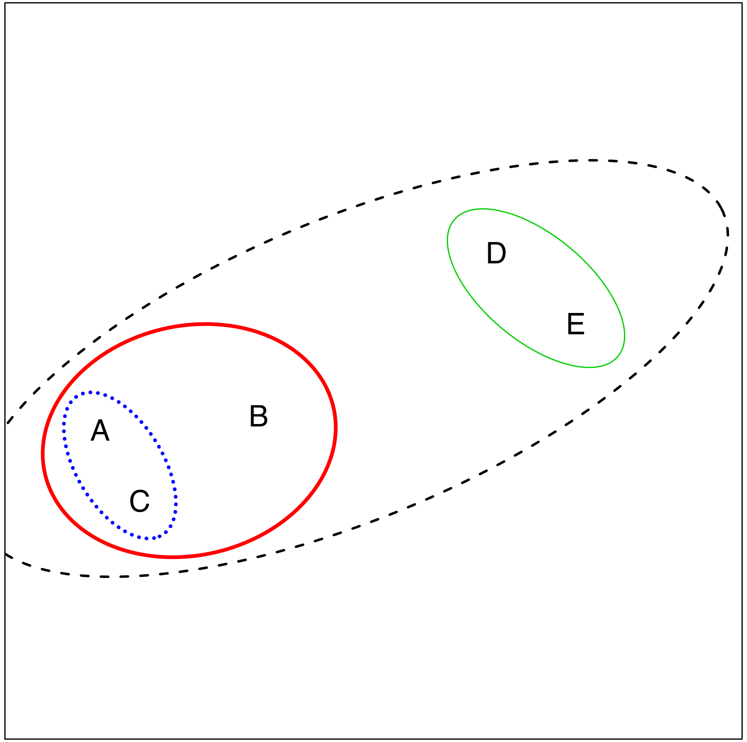
AGGLOMERATIVE CLUSTERING



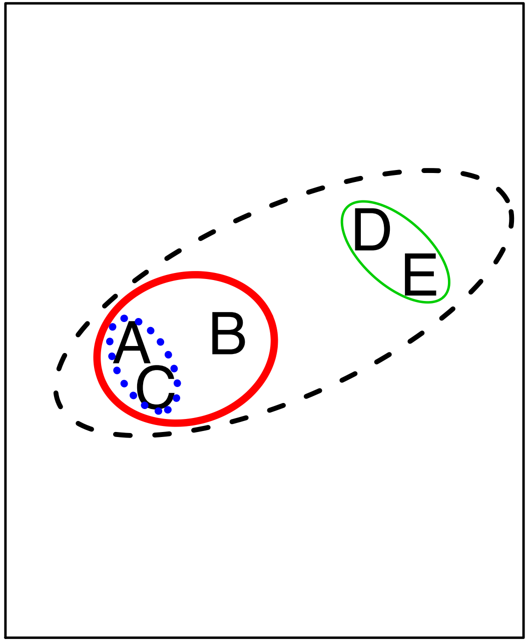
AGGLOMERATIVE CLUSTERING

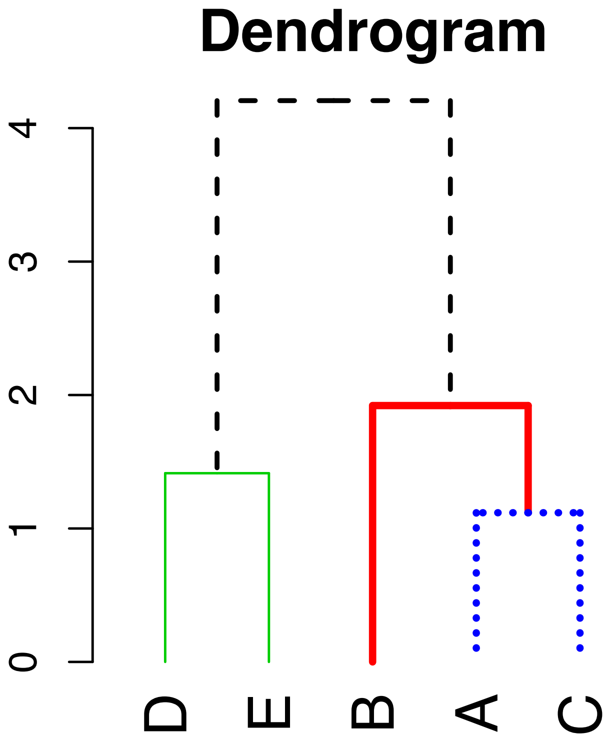


AGGLOMERATIVE CLUSTERING

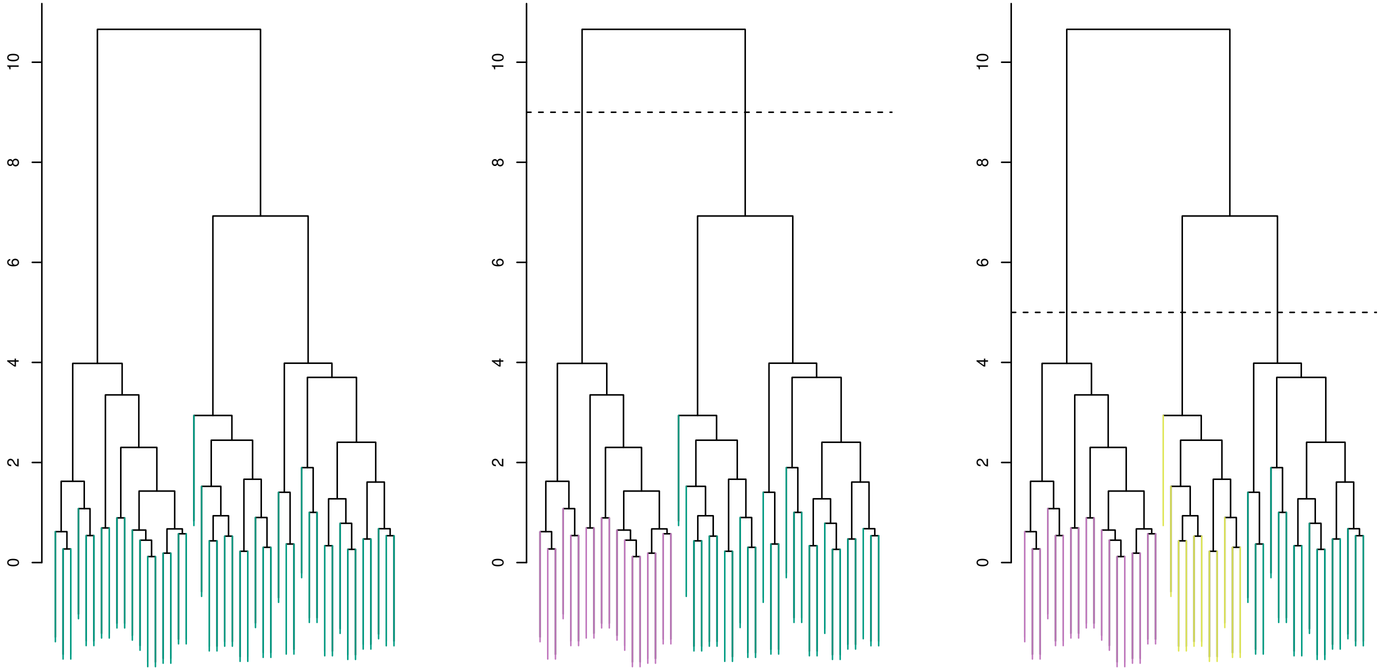


AGGLOMERATIVE CLUSTERING




Dendrogram

AGGLOMERATIVE CLUSTERING



Linkage function defines distance between subsets of points.

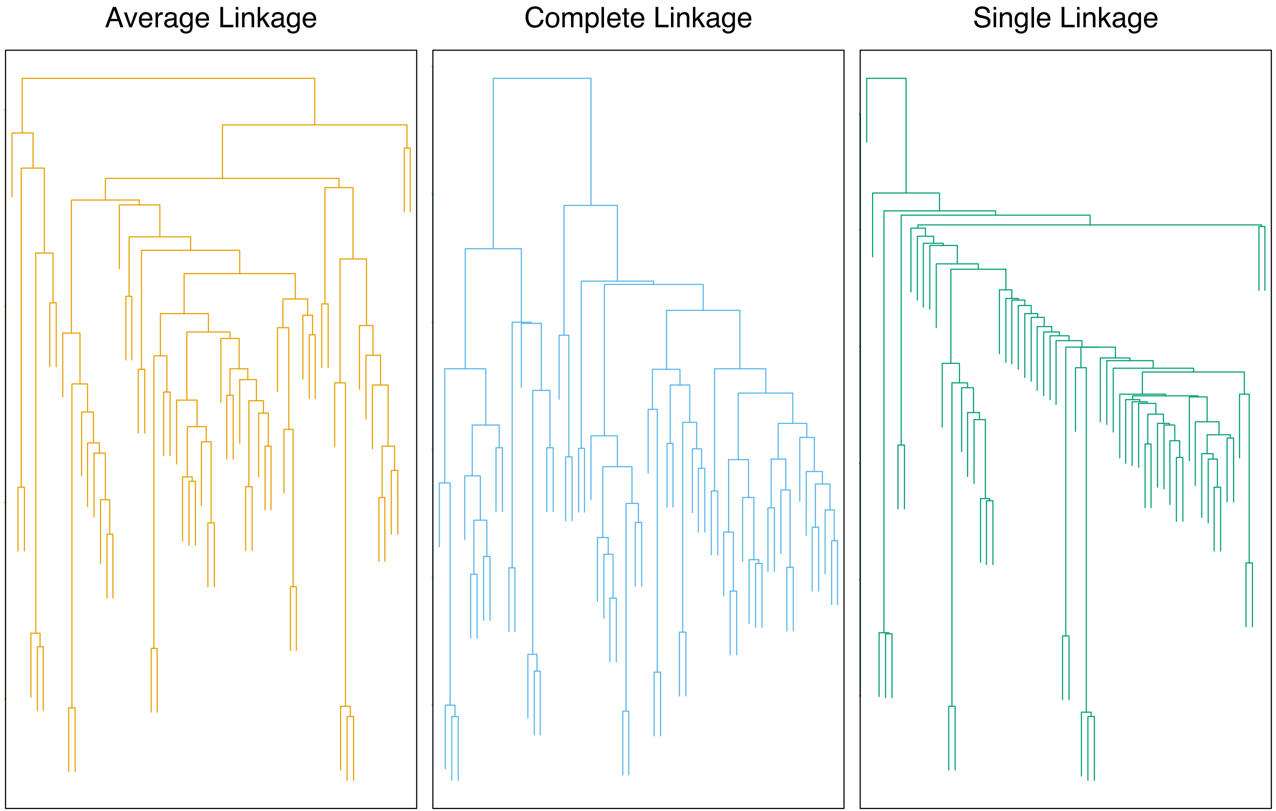
$$L_{\text{single}}(A, B) = \min_{x \in A, y \in B} \text{Dis}(x, y)$$

$$L_{\text{complete}}(A, B) = \max_{x \in A, y \in B} \text{Dis}(x, y)$$

$$L_{\text{average}}(A, B) = \frac{\sum_{x \in A, y \in B} \text{Dis}(x, y)}{|A| \cdot |B|}$$

$$L_{\text{centroid}}(A, B) = \text{Dis}\left(\frac{\sum_{x \in A} x}{|A|}, \frac{\sum_{y \in B} y}{|B|}\right)$$

AGGLOMERATIVE CLUSTERING

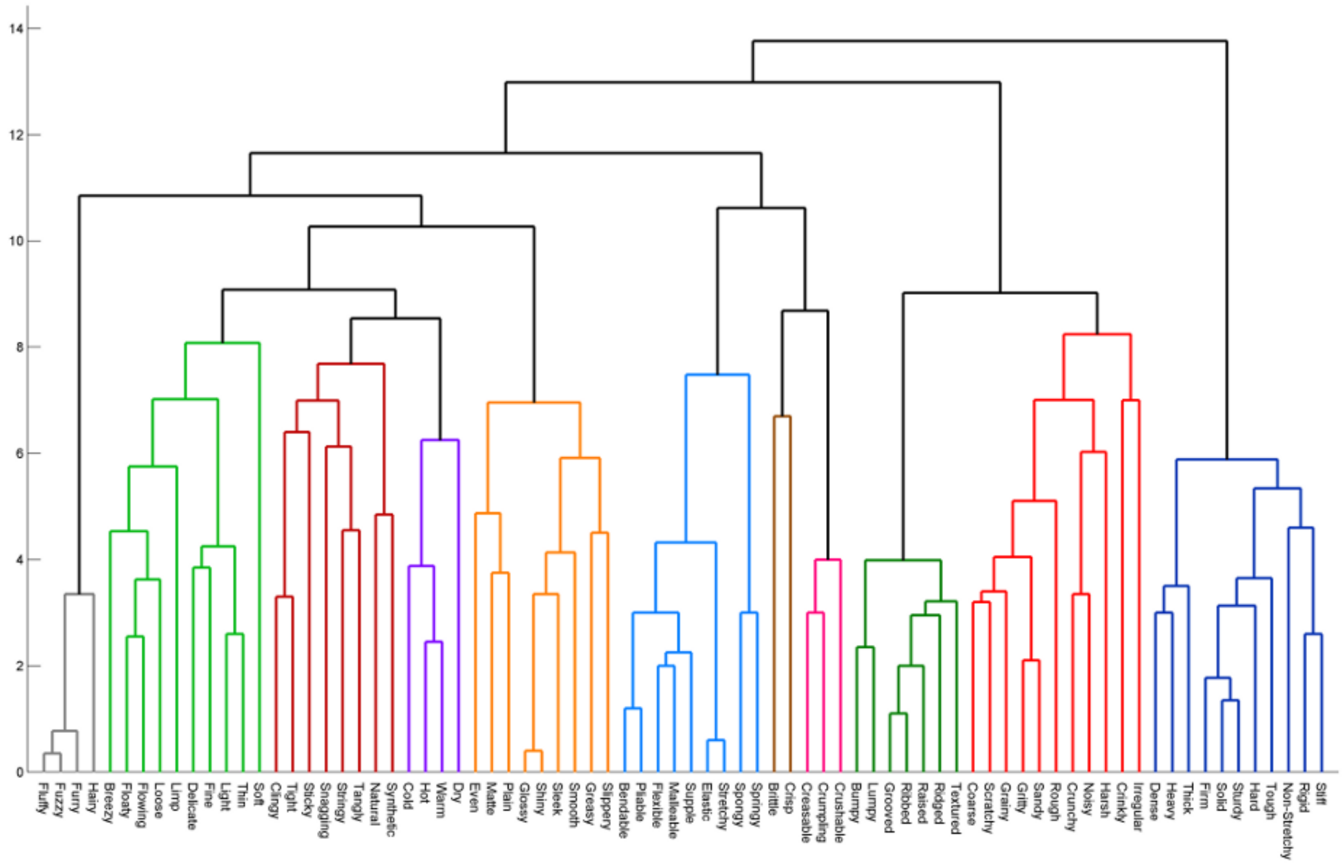


Average Linkage

Complete Linkage

Single Linkage

AGGLOMERATIVE CLUSTERING



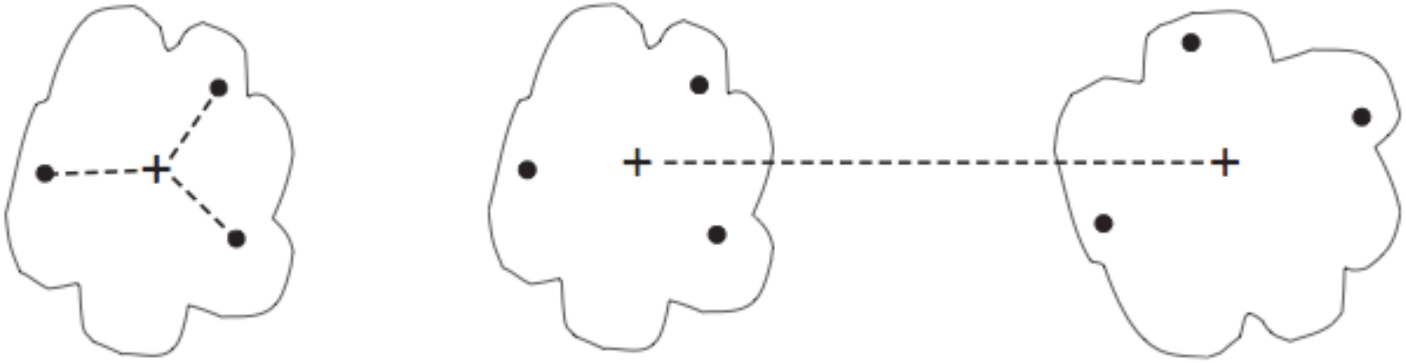
III. CLUSTER VALIDATION

Cohesion measures clustering effectiveness within a cluster.

$$\hat{C}(C_i) = \sum_{x \in C_i} d(x, c_i)$$

Separation measures clustering effectiveness between clusters.

$$\hat{S}(C_i, C_j) = d(c_i, c_j)$$



(a) Cohesion.

(b) Separation.

Figure 8.28. Prototype-based view of cluster cohesion and separation.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

We can turn these values into overall measures of clustering validity by taking a weighted sum over clusters:

$$\hat{V}_{total} = \sum_{1}^{K} w_i \hat{V}(C_i)$$

Here V can be cohesion, separation, or some function of both.

The weights can all be set to 1 (best for k-means), or proportional to the cluster *masses* (the number of points they contain).

SILHOUETTE COEFFICIENT

One useful measure than combines the ideas of cohesion and separation is the **silhouette coefficient**. For point x_i, this is given by:

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

such that:

a_i = average in-cluster distance to x_i

b_{ij} = average between-cluster distance to x_i

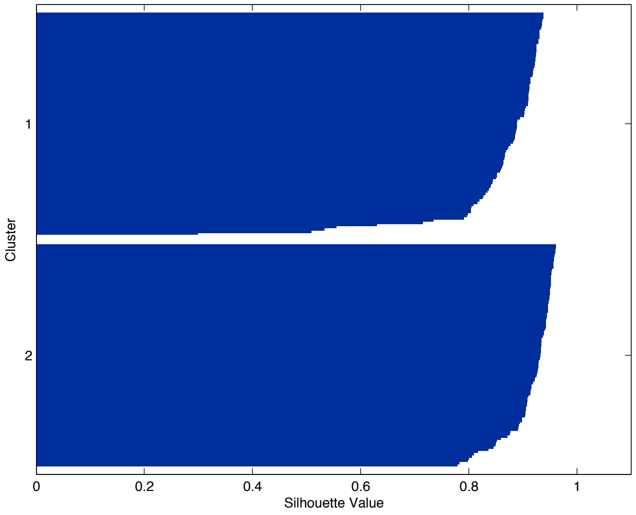
$b_i = min_j(b_{ij})$

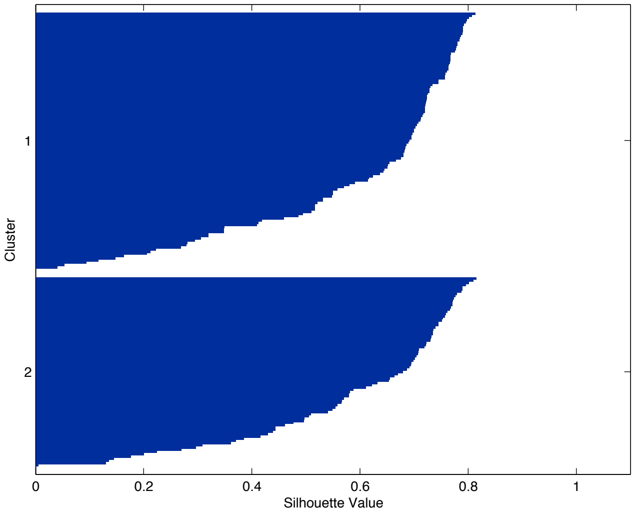
The silhouette coefficient can take values between -1 and 1.

In general, we want separation to be high and cohesion to be low. This corresponds to a value of SC close to +1.

A negative silhouette coefficient means the cluster radius is larger than the space between clusters, and thus clusters overlap.

SILHOUETTE COEFFICIENT





$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{\max(a(\mathbf{x}_i), b(\mathbf{x}_i))}$$

$$a(\mathbf{x}_i) = d(\mathbf{x}_i, D_{j(i)})$$

$$b(\mathbf{x}_i) = \min_{k \neq j(i)} d(\mathbf{x}_i, D_k)$$

SILHOUETTE COEFFICIENT

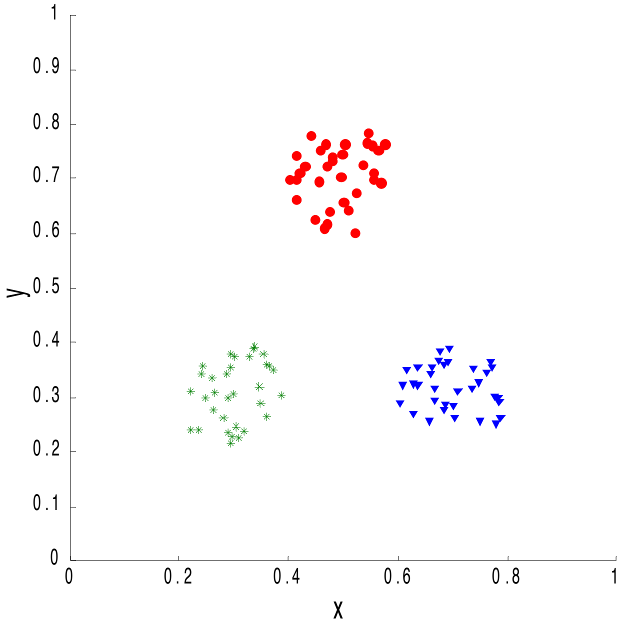
The silhouette coefficient for the cluster C_i is given by the average silhouette coefficient across all points in C_i:

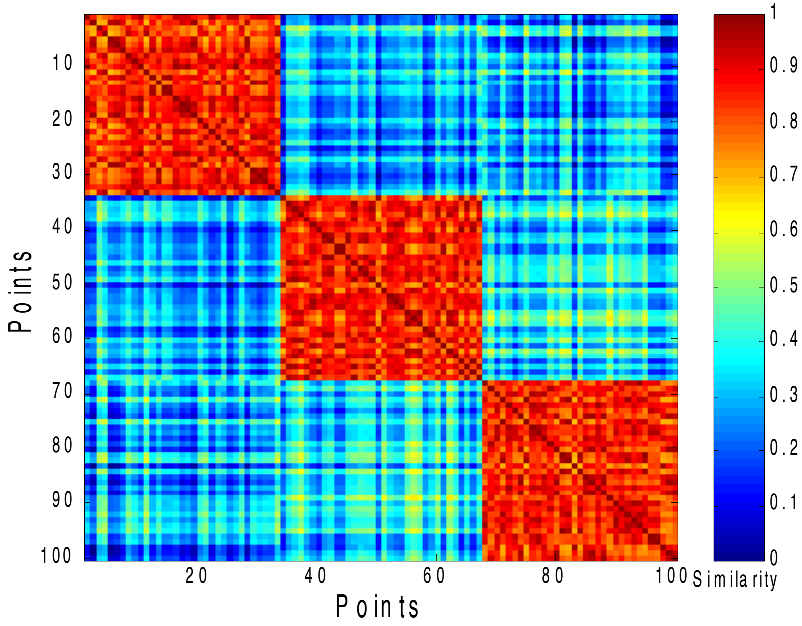
$$SC(C_i) = \frac{1}{m_i} \sum_{x \in C_i} SC_i$$

The overall silhouette coefficient is given by the average silhouette coefficient across all points:

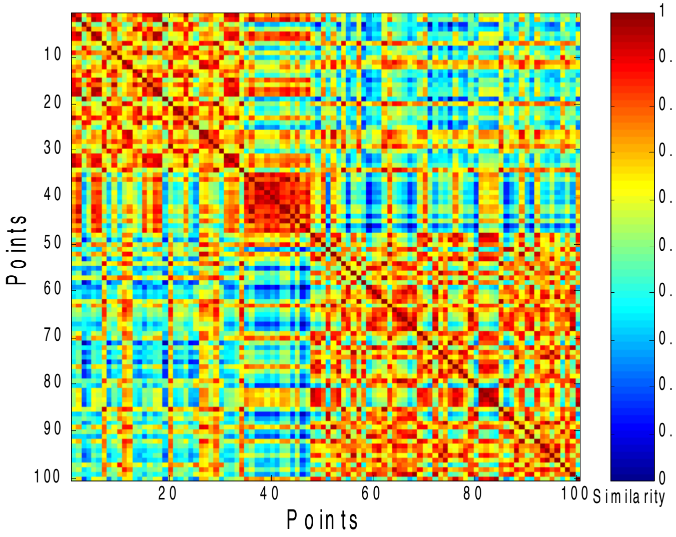
$$SC_{total} = \frac{1}{k} \sum_{1}^{k} SC(C_i)$$

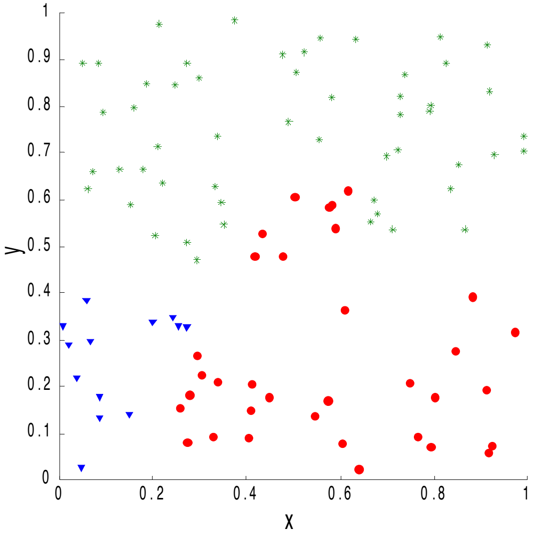
SIMILARITY MATRIX





SIMILARITY MATRIX





Ultimately, cluster validation and clustering in general are suggestive techniques that rely on human interpretation to be meaningful.

IV. LAB: CLUSTERING IN SCIKIT-LEARN