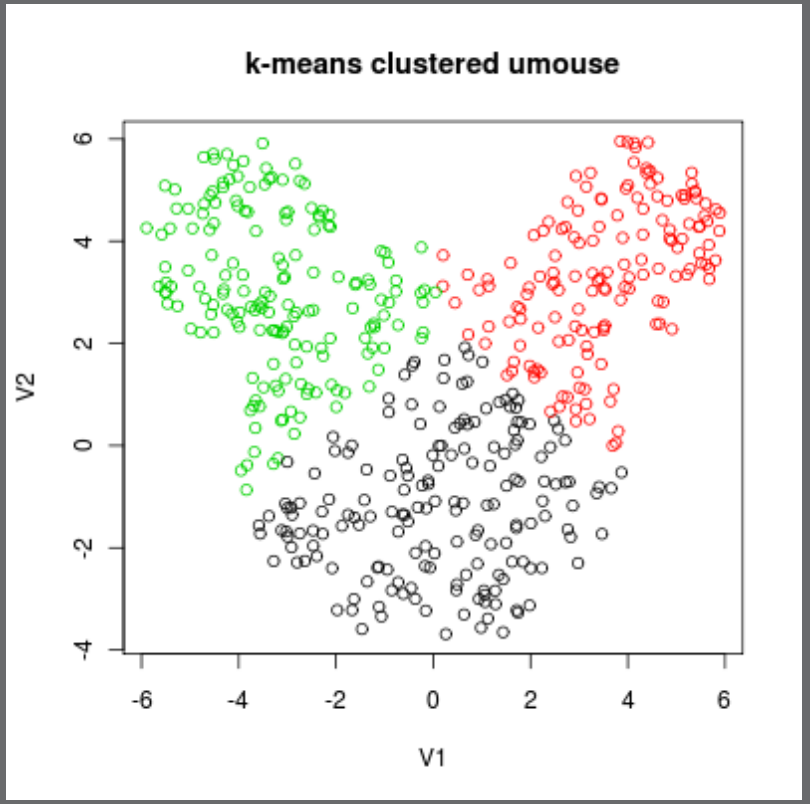
Clustering

Kayla Thomas
Thanks:
Taryn Heilman, Frank Burkholder

galvanize



Learning Objectives

galvanize

- Introduce Unsupervised Learning, compare to Supervised
- Intro to clustering
- Use cases for clustering algorithms
- K-means algorithm:
  - How to code
  - Centroid initialization
  - Stopping Criteria
  - Evaluating the algorithm (metrics)
  - How to choose k
- Hierarchical Clustering
  - Dendrogram

Review relevant concepts

galvanize

- What does supervised learning mean? Give an example
- Describe the difference between parametric and non-parametric learners, and give an example of each
- How do we measure the “success” of a supervised learning algorithm? Give examples for classification and regression
- Describe the euclidean distance metric. Name two additional distance metrics we have discussed and describe.
- What is the curse of dimensionality?

Supervised vs. Unsupervised Learning

galvanize

Supervised

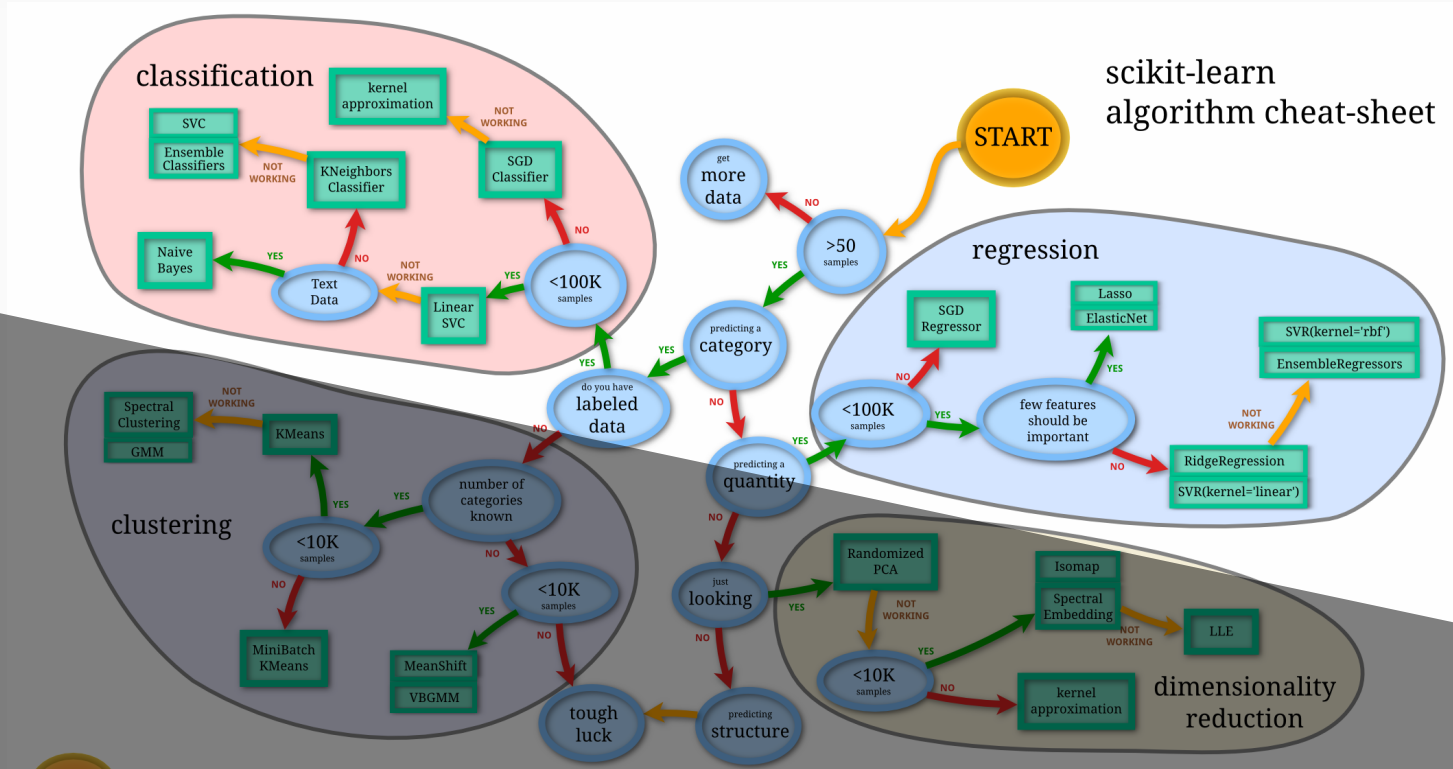
- Have a target/label that we model
- Models look like functions that take in features (X) and predict a label (y)
- Have an error metric that we can use to compare models
- Used to predict future unlabeled data

Unsupervised

- No target/label to predict
- Goal is to find underlying structure, patterns, or organization in data
- No stark error metric to compare models - determining if you have the optimal solution is very challenging. Cross validation often not applicable.

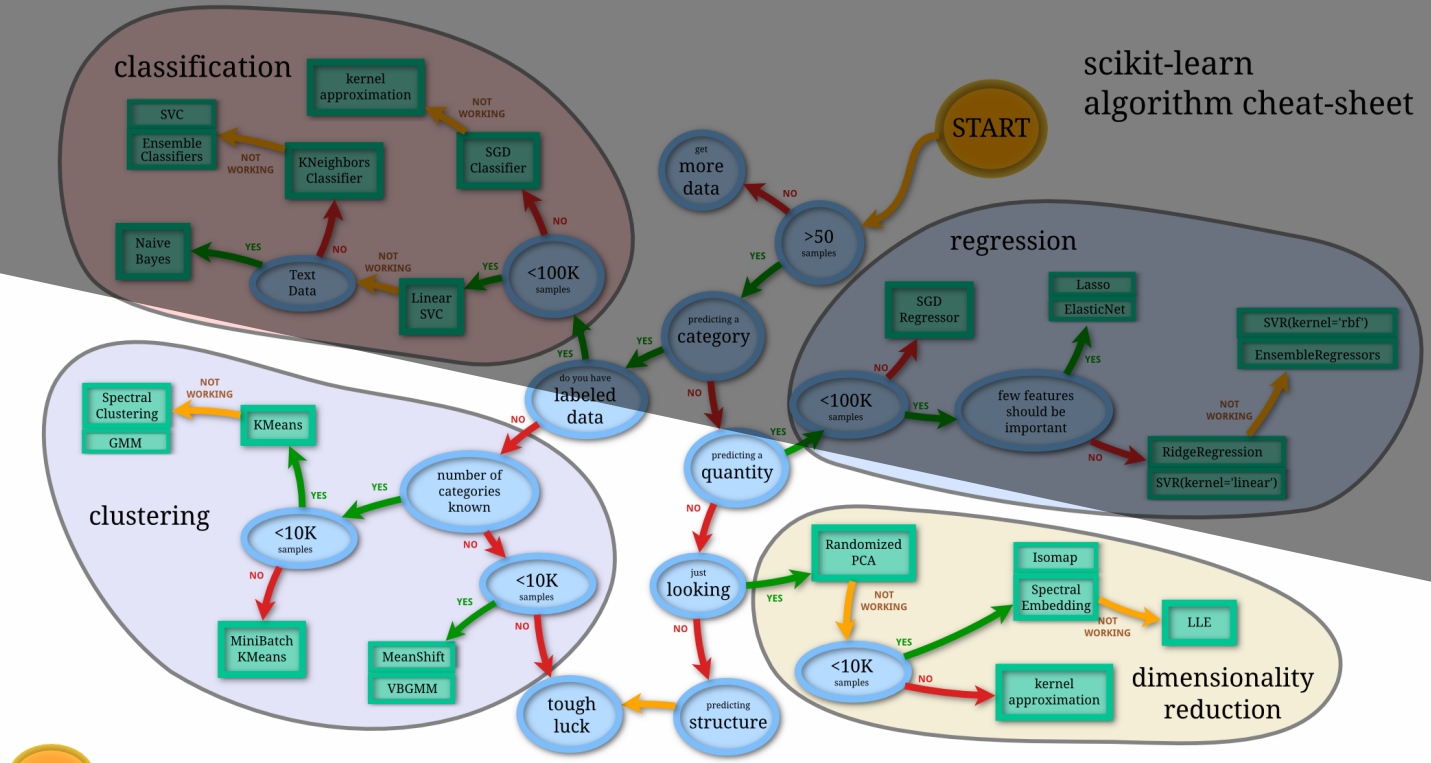
Where We've Been - Supervised Learning

galvanize



Unsupervised Learning - Where We are Going!

galvanize



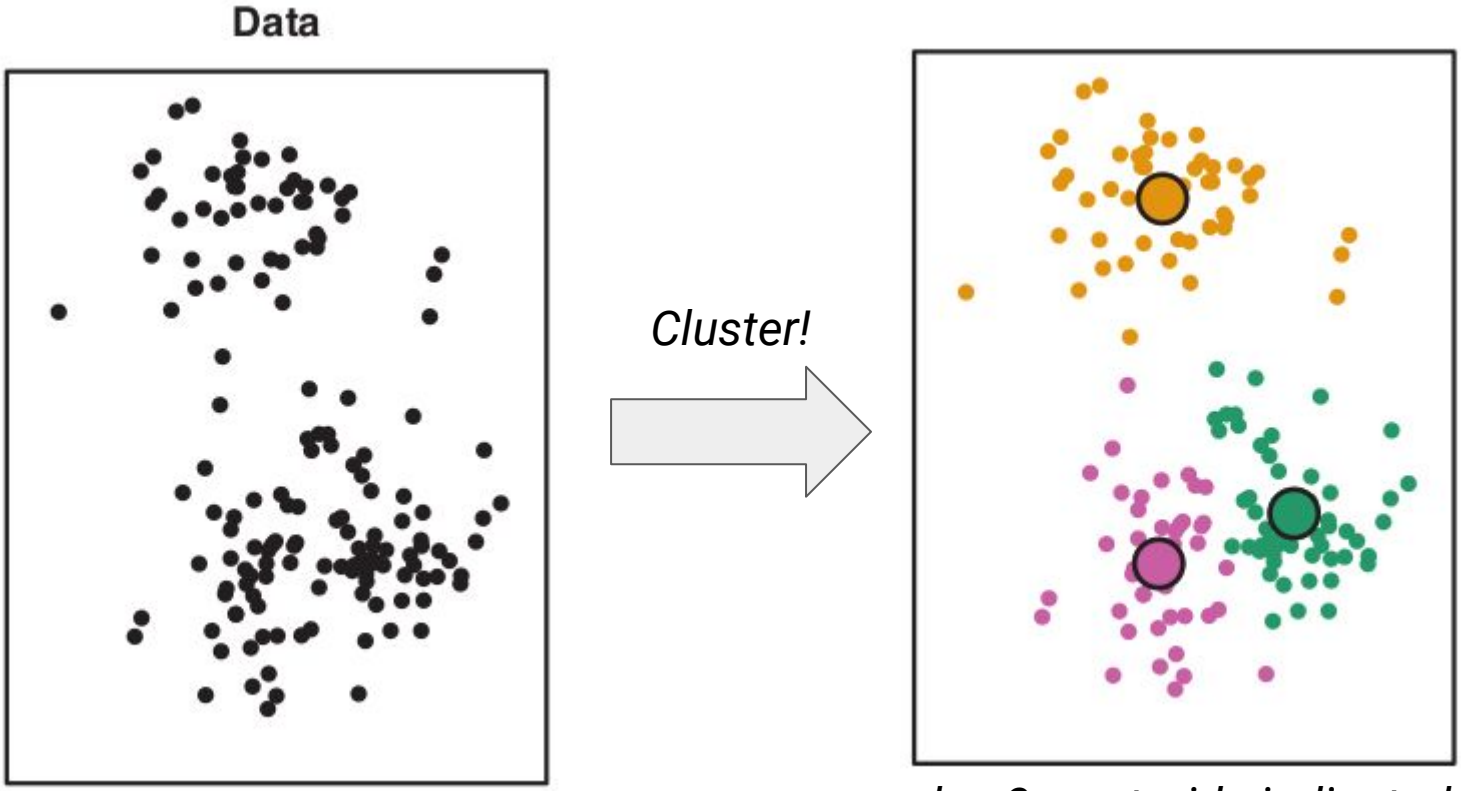
http://scikit-learn.org/stable/tutorial/machine_learning_map/

Real world use cases

galvanize

- Customer segmentation
- Product segmentation
- Image segmentation
- Anomaly detection
- Social network analysis
- and many more…

Clustering



k = 3, centroids indicated by large circles

k-means clustering

Would like to partition data into k groups (clusters) so that the total “within cluster variation” (WCV) is the smallest:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \text{WCV}(C_k) \right\}$$

Where WCV for the k-th cluster is the sum of all the pairwise Euclidean distances:

$$\text{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

$|C_k|$ is number of observations in k-th cluster

Centroid of cluster C_k

This is the goal (and factors into the cost function), but how to get there?

Basic K-Means Algorithm

galvanize

Initialize k centroids*

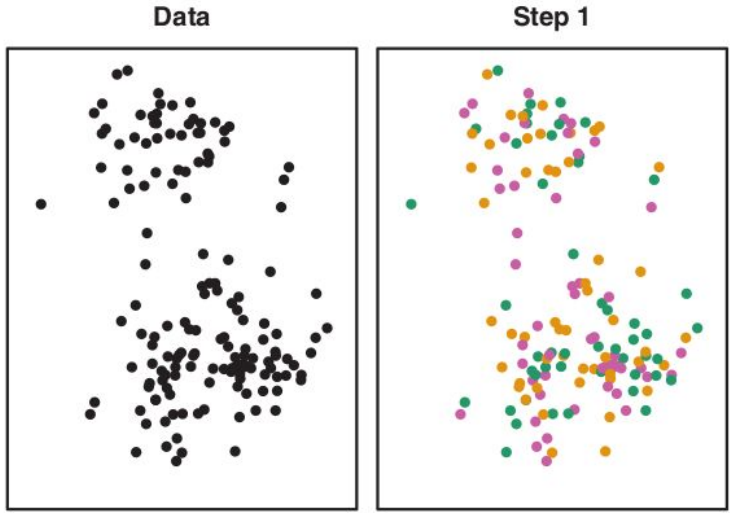
Until convergence**:

- assign each data point to the nearest centroid
- recompute the centroids as the mean of the data points

```
# psuedocode
initialize_centroids
while not converged:
  assign_data_to_centroids
  compute_new_centoid_means
```

k-means clustering algorithm (visually)

1. Randomly assign each data point to a cluster.



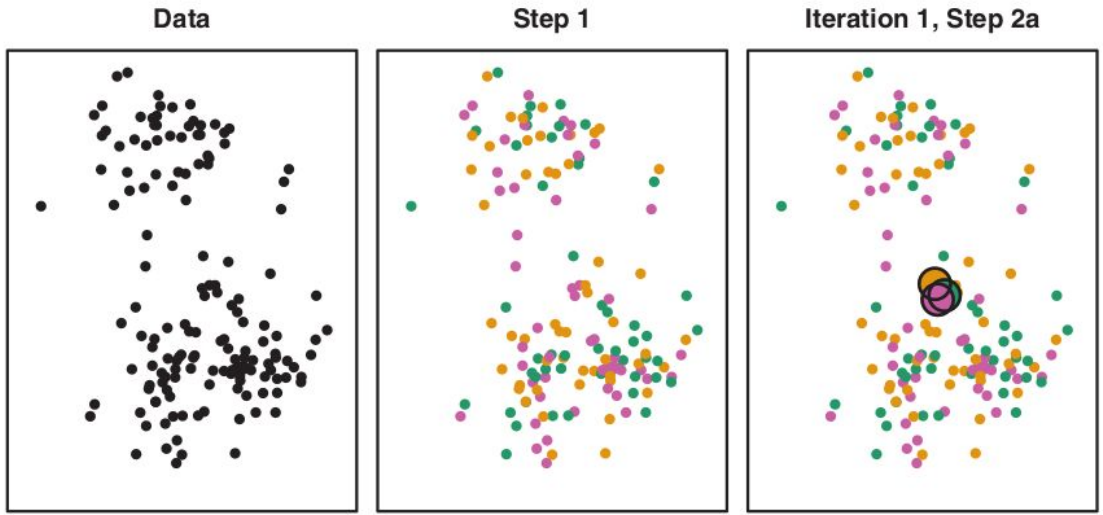
James, G. et al., Introduction to Statistical Learning, 2015

k-means clustering algorithm (visually)

1. Randomly assign each data point to a cluster.

-while cluster assignments change-

2a) Compute the centroid of each cluster.



James, G. et al., Introduction to Statistical Learning, 2015

k-means clustering algorithm (visually)

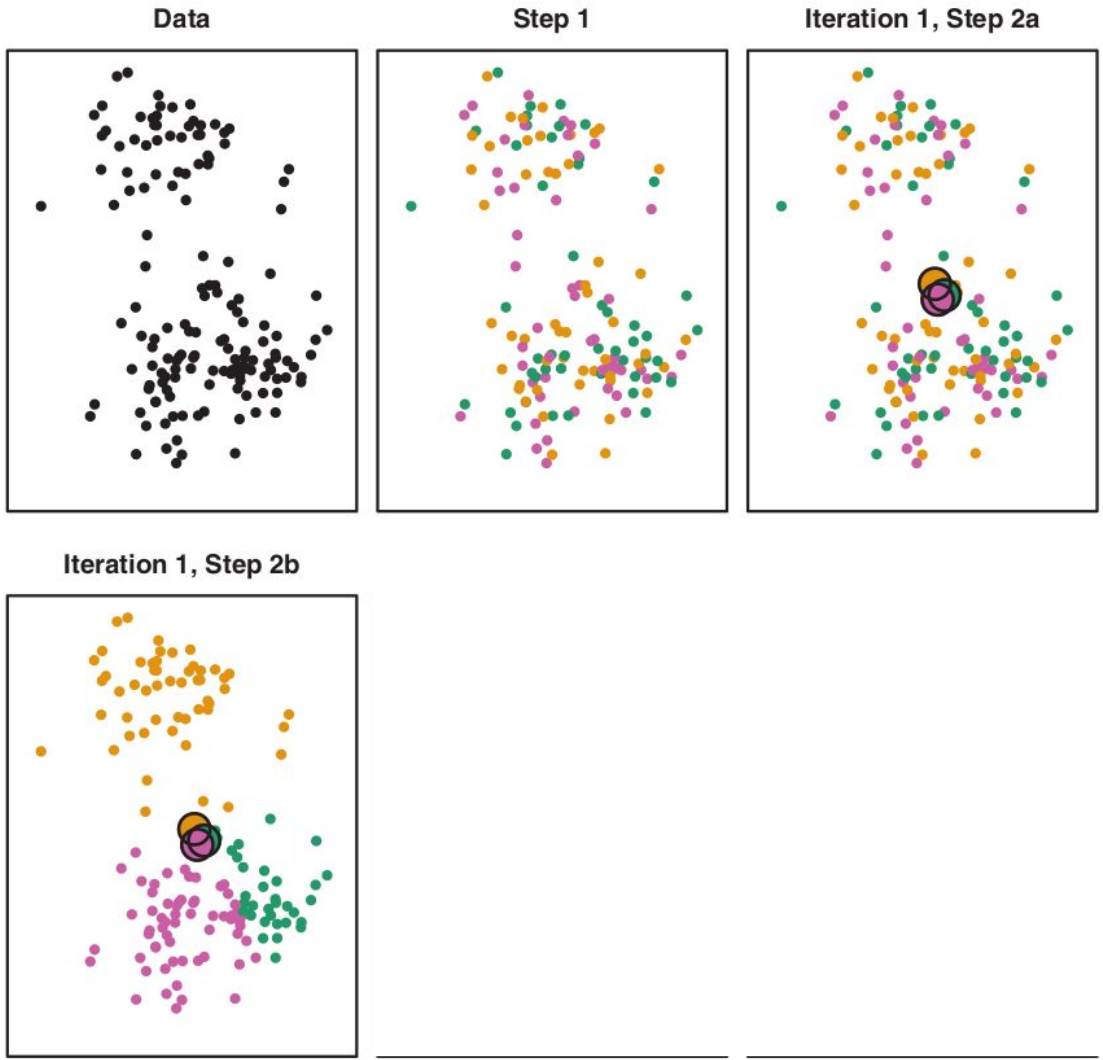
1. Randomly assign each data point to a cluster.

-while cluster assignments change-

2a) Compute the centroid of each cluster.

2b) Reassign each point to belong to the nearest cluster.

James, G. et al., Introduction to Statistical Learning, 2015



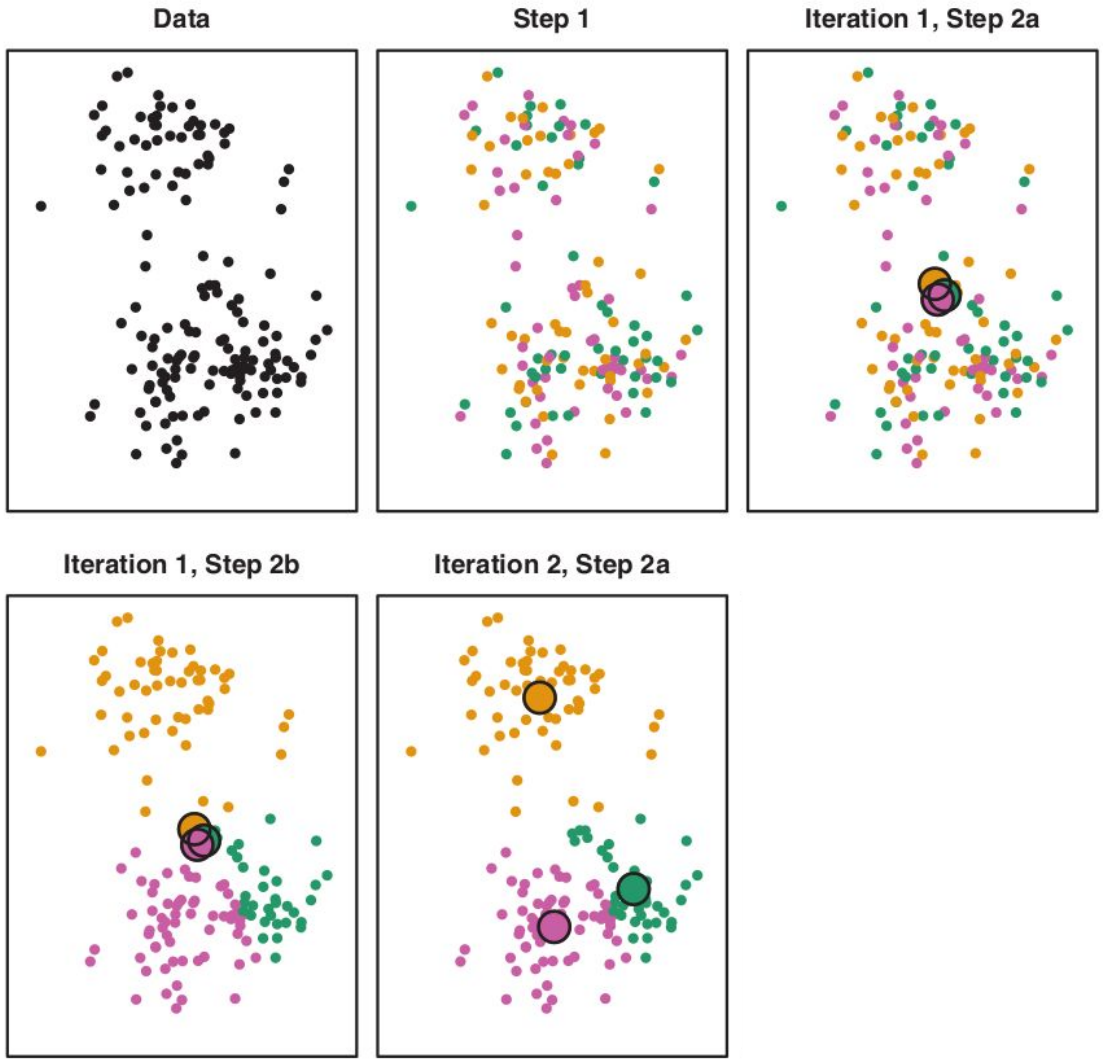
k-means clustering algorithm (visually)

1. Randomly assign each data point to a cluster.

-while cluster assignments change-

2a) Compute the centroid of each cluster.

2b) Reassign each point to belong to the nearest cluster.



James, G. et al., Introduction to Statistical Learning, 2015

k-means clustering algorithm (visually)

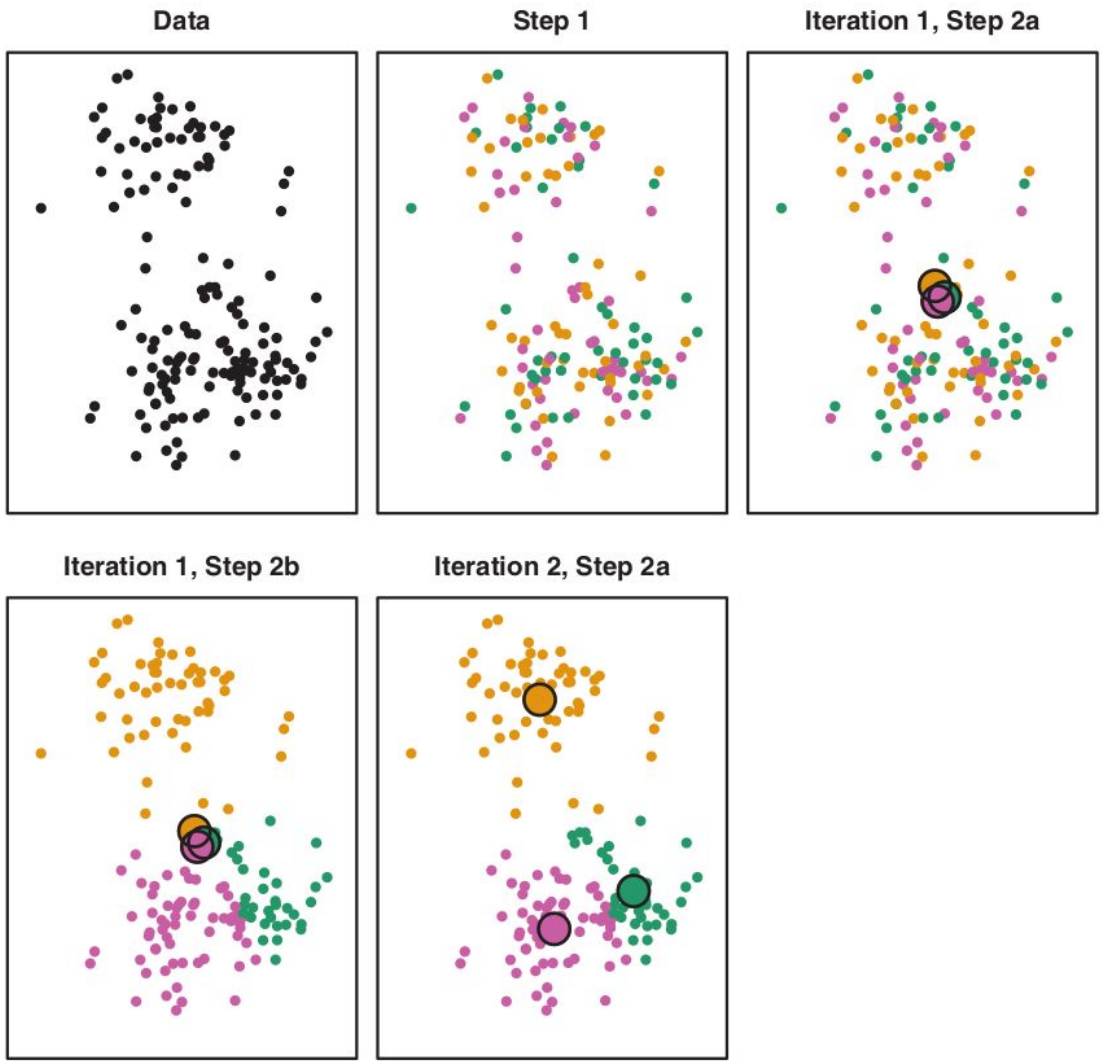
1. Randomly assign each data point to a cluster.

-while cluster assignments change-

2a) Compute the ce[illegible] of each cluster.

2b) Reassi[illegible]n point to belong to the ne[illegible] cluster.

After many steps



James, G. et al., Introduction to Statistical Learning, 2015

k-means clustering algorithm (visually)

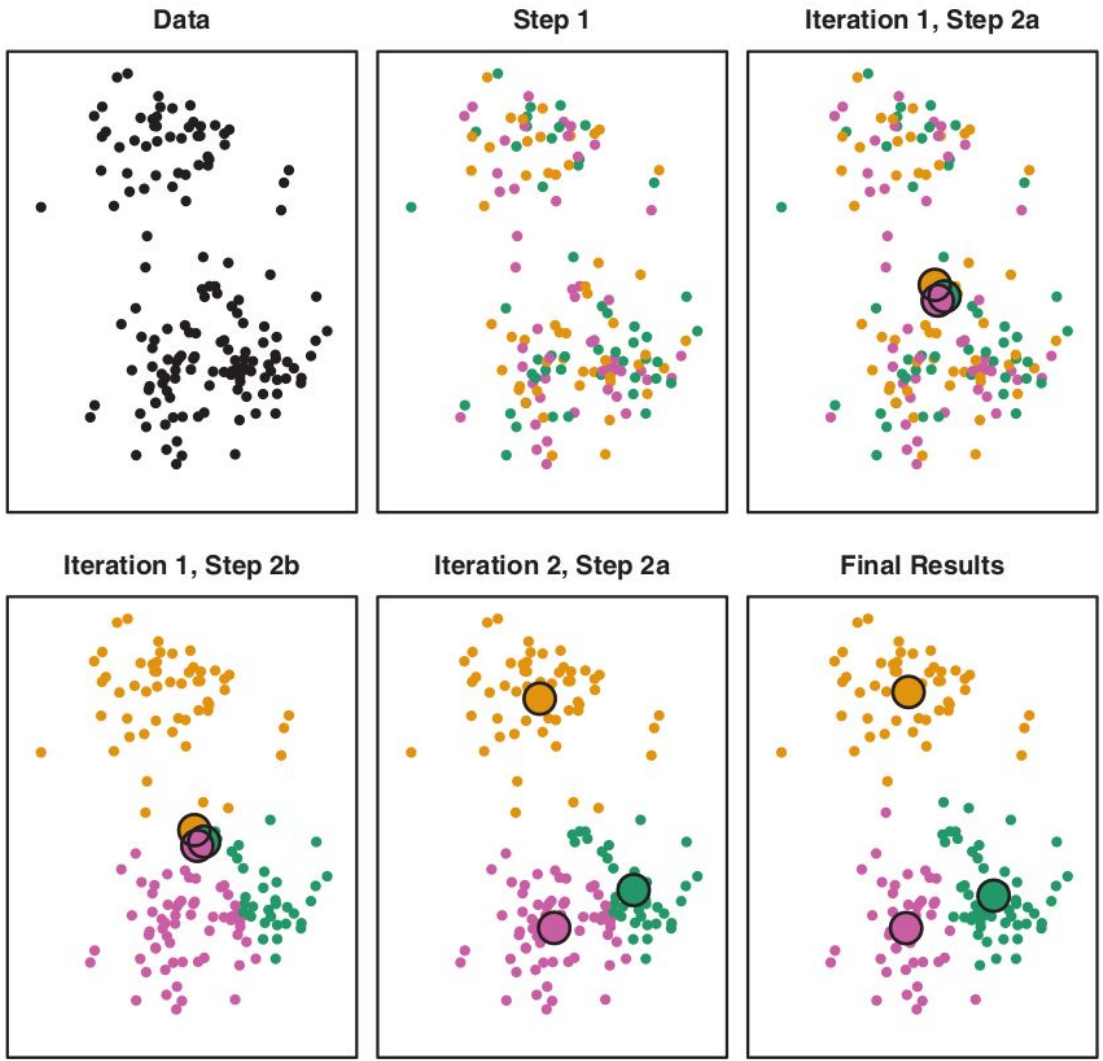
1. Randomly assign each data point to a cluster.

-while cluster assignments change-

2a) Compute the ce[illegible] of each cluster.

2b) Reassi[illegible]n point to belong to the ne[illegible] cluster.

After many steps



James, G. et al., Introduction to Statistical Learning, 2015

k-means convergence

Value above each chart is:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Values vary because initializations are random, and so are only guaranteed to find *local* minima.

Need to run several times (or k-means++ - show algorithm)

6 solutions:

James, G. et al., Introduction to Statistical Learning, 2015

Centroid Initialization Methods

galvanize

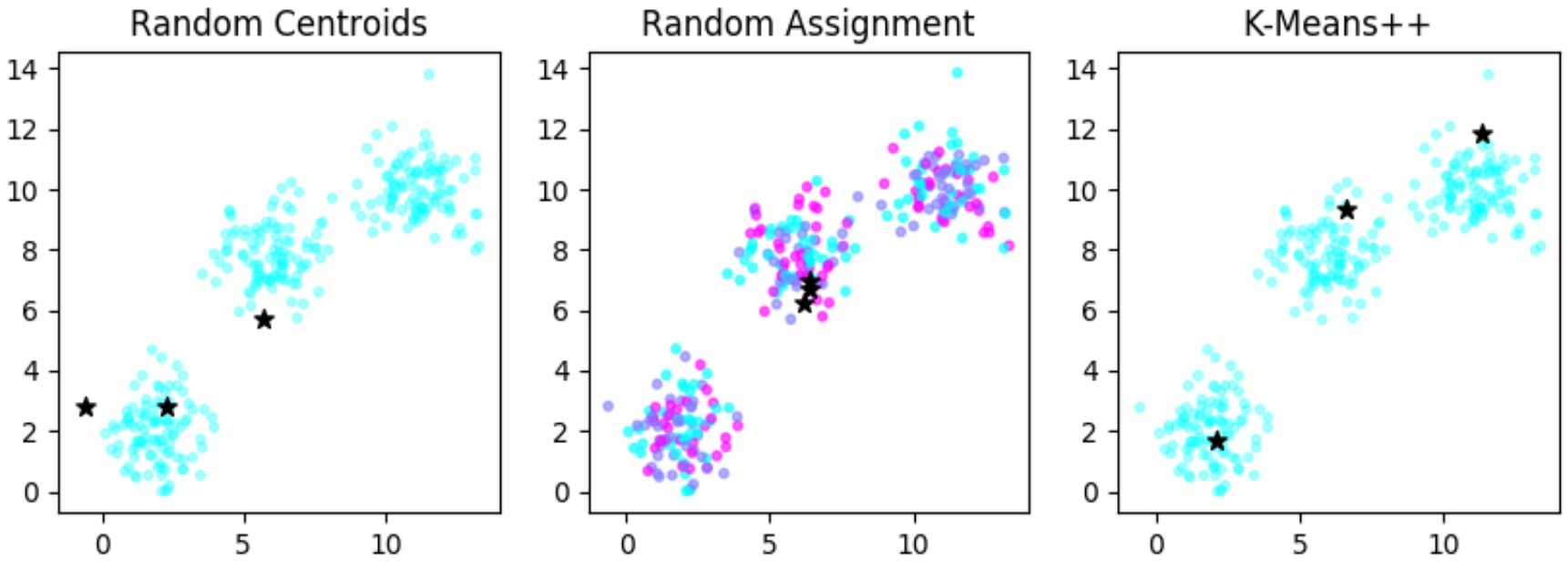
1) (Simplest) Randomly choose k points from your data and make those your initial centroids

2) Randomly assign each data point to a number 1-k and initialize the kth centroid to the average of the points with the kth label.

3) **k-means++** chooses well spread initial centroids. First centroid is chosen at random, with subsequent centroids chosen with probability proportional to the squared distance to the closest existing centroid. *This is the default initialization in sklearn.*

Centroid Initialization Methods

galvanize



Stopping Criteria

galvanize

We can update for:

1) A specified number of iterations (*sklearn default : max_iter= 1000*)

2) Until the centroids don't change at all

3) Until the centroids don't move by very much (*sklearn default : tol= .0001*)

Quick Review

galvanize

Goal of unsupervised learning?

Describe the 2 steps in each k-means fitting iteration

Name 3 ways to choose centroids

Name 3 stopping criteria

Is K-Means deterministic?

Should we standardize features?

Breakout (AKA numpy practice)

galvanize

Use these arrays:

```
features = np.array([[ 3,  4],
                     [ 2,  2],
                     [ 5,  4],
                     [ 6,  9],
                     [-1,  0]])

labels = np.array([1, 0, 1, 0, 1])
```

1. Use masking to grab the rows of features that correspond to the label of 1
2. Find the column-wise mean of that subset of features
3. Compute the euclidean distance from each data point in features to this mean. Try to do this in one line with broadcasting!

Breakout (AKA numpy practice)

galvanize

Use these arrays:

```
features = np.array([[ 3,  4],
                     [ 2,  2],
                     [ 5,  4],
                     [ 6,  9],
                     [-1,  0]])

labels = np.array([1, 0, 1, 0, 1])
```

1. `subset = features[labels == 1]`
2. `mean = subset.mean(axis = 0)`
3. `np.linalg.norm(features - mean, axis = 1)`

Evaluating K-Means

galvanize

How do we measure how well our clustering does?

A good measure should quantify how similar things are within a cluster

Want to minimize Intra-Cluster Variance or Within Cluster Variance (WCV)

$$\min\left\{\sum_{k=1}^{K} WCV(C_k)\right\}$$

Where WCV for the kth cluster is the sum of all the pairwise Euclidean distances

$$WCV(C_k) = \frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2$$

Centroid of cluster C_k

Choosing k

galvanize

Not easy or very precise!

Possible methods:

- Silhouette Score
- Other methods (e.g. GAP statistic) that you don't need to know today
    - See clustering docs for more performance evaluations
- Domain knowledge

Silhouette Score

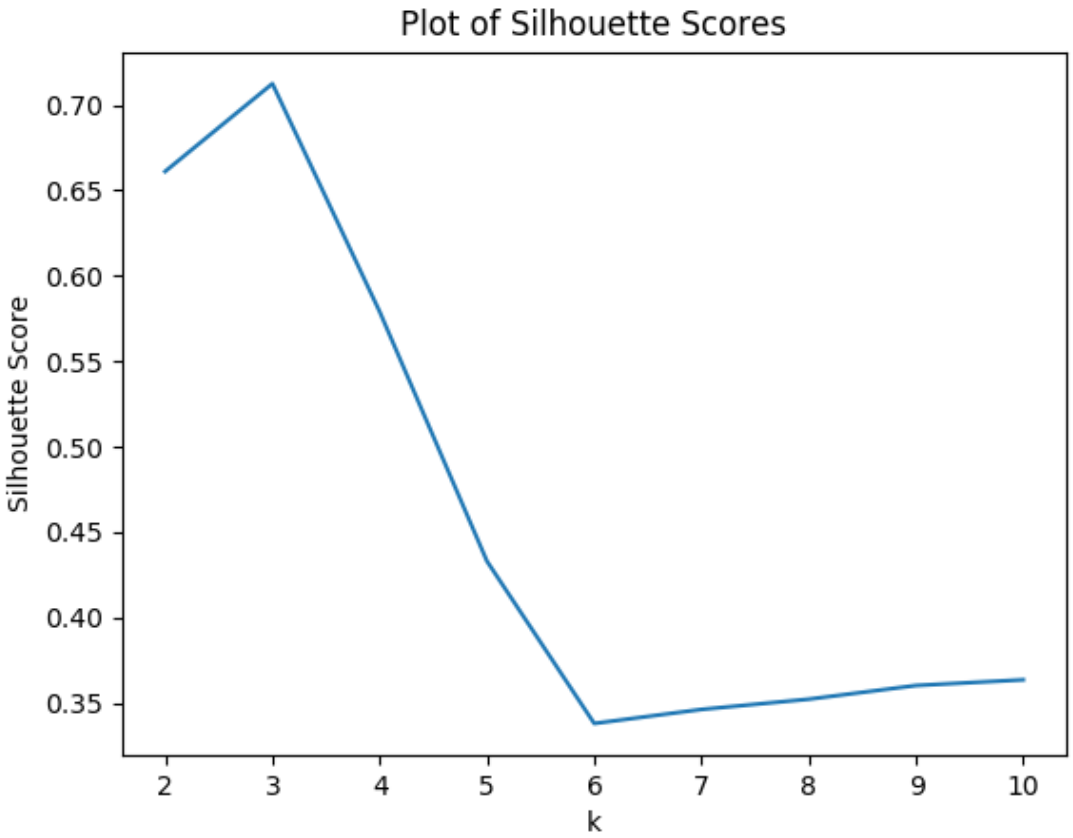
The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample

(b - a) / max(a, b)

*only defined for 2 <= k < n

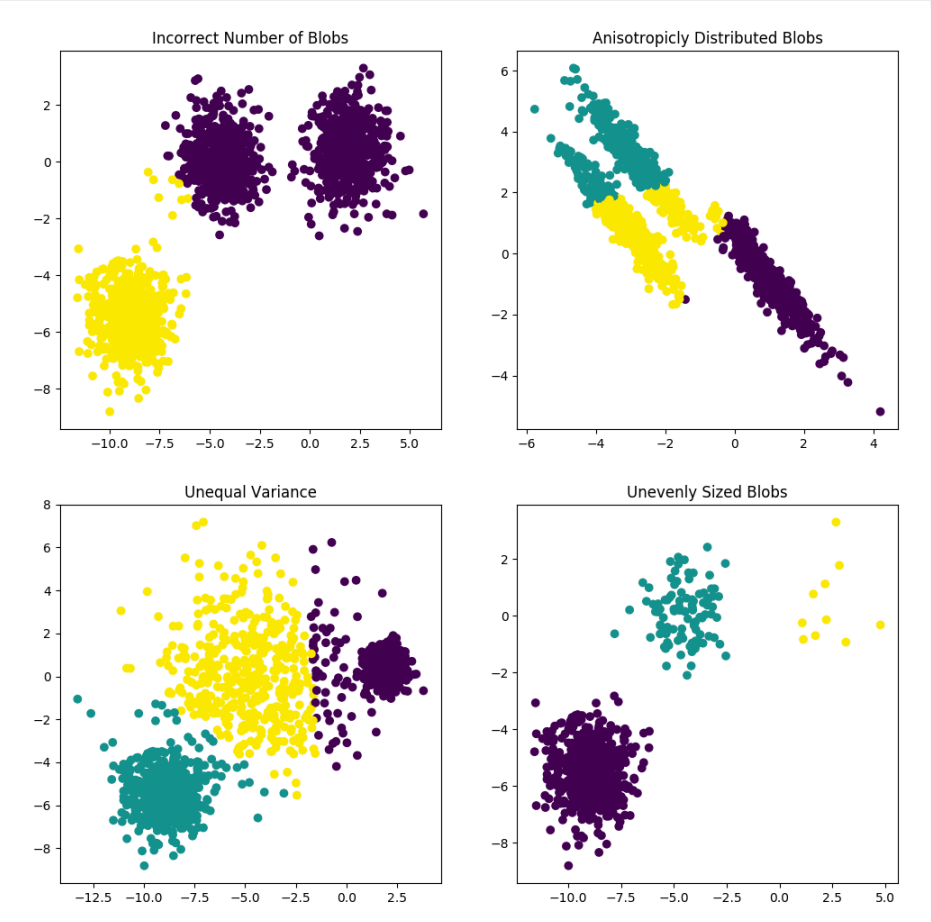
Values range from -1 to 1, with 1 being optimal and -1 being the worst

http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html#sklearn.metrics.silhouette_score



K-Means Assumptions

- Picked the “correct” k
- Clusters have equal variance
- Clusters are isotropic (variance spherical)
- Clusters do NOT have to contain the same number of observations



http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_assumptions.html

Practical Considerations

galvanize

- K-means is not deterministic -> falls into local minima. Should restart multiple times and take the version with the lowest within-cluster variance (sklearn does multiple initializations by default)
- Susceptible to curse of dimensionality
- One hot encoded categorical can overwhelm - look into k-modes (https://pypi.python.org/pypi/kmodes/)
- Try MiniBatchKMeans for large datasets (finds local minima, so be careful)

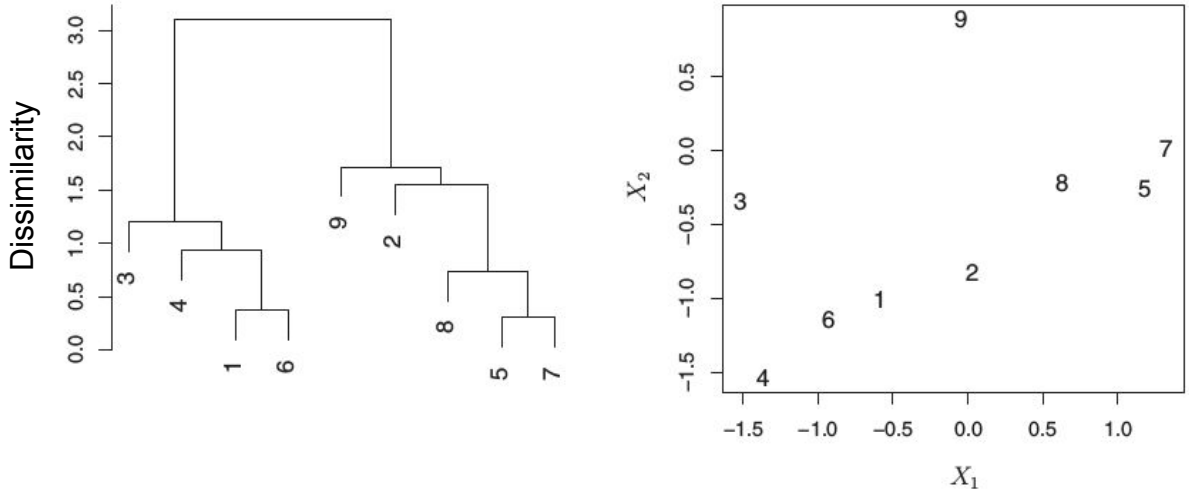
DBScan Algorithm

galvanize

- Also computes clusters using distance metric, but decides the number of clusters for you
- With K-Means, you choose k - this is your **main** hyper-parameter
- With DBScan, your main hyper-parameter is **eps,** which is the maximum distance between two points that are allowed to be in the same ‘neighborhood’

Hierarchical clustering

- Another clustering method (creating groups through *hierarchies*)
- Don’t have to commit to a value of k beforehand
- Results don’t depend on initialization
- Not limited to euclidean distance as the similarity metric
- Easy visualized through dendrograms
    - “Height of fusion” on dendrogram quantifies the separation of clusters



Hierarchical clustering algorithm

Algorithm 10.2 *Hierarchical Clustering* *(Agglomerative, bottom up)*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

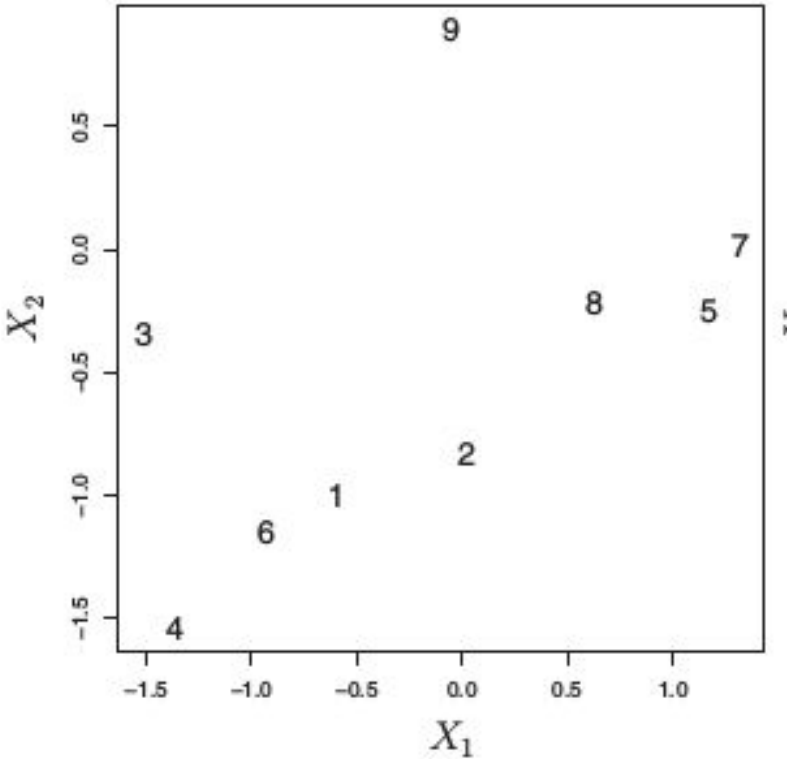
2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed. In units of your dissimilarity metric

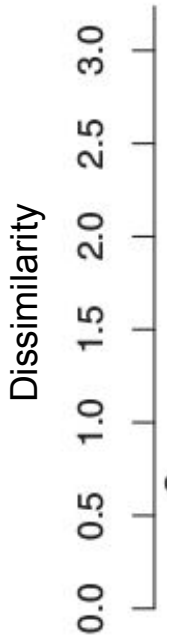
   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Hierarchical clustering algorithm (visualized)

Data

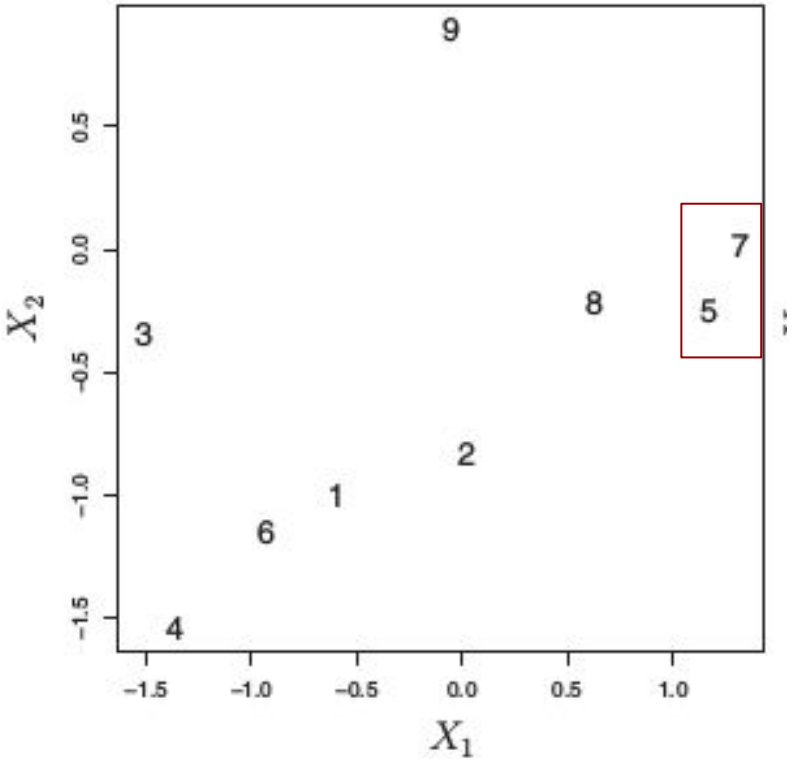


Dendrogram

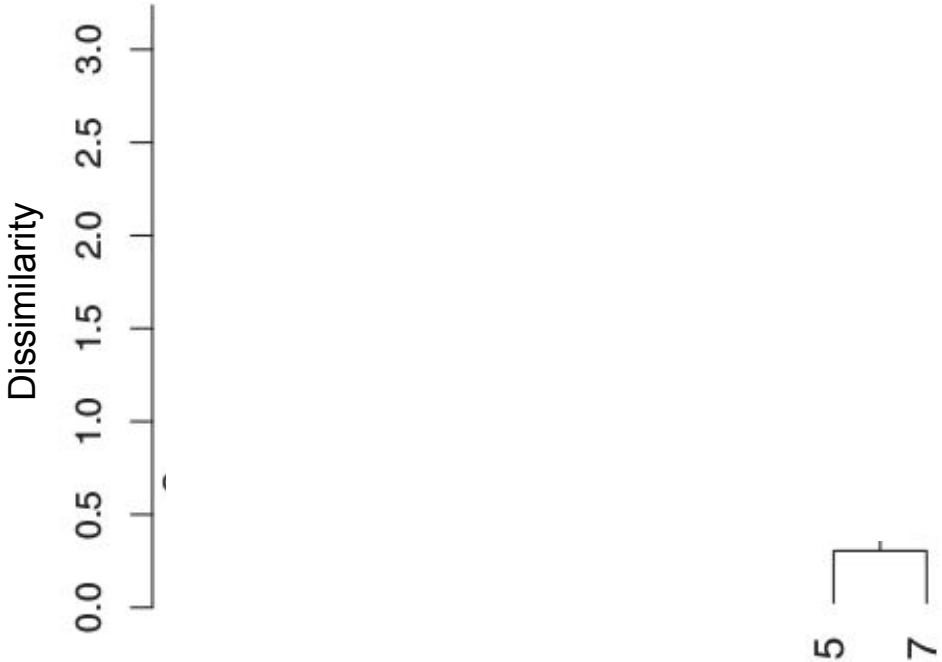


Hierarchical clustering algorithm (visualized)

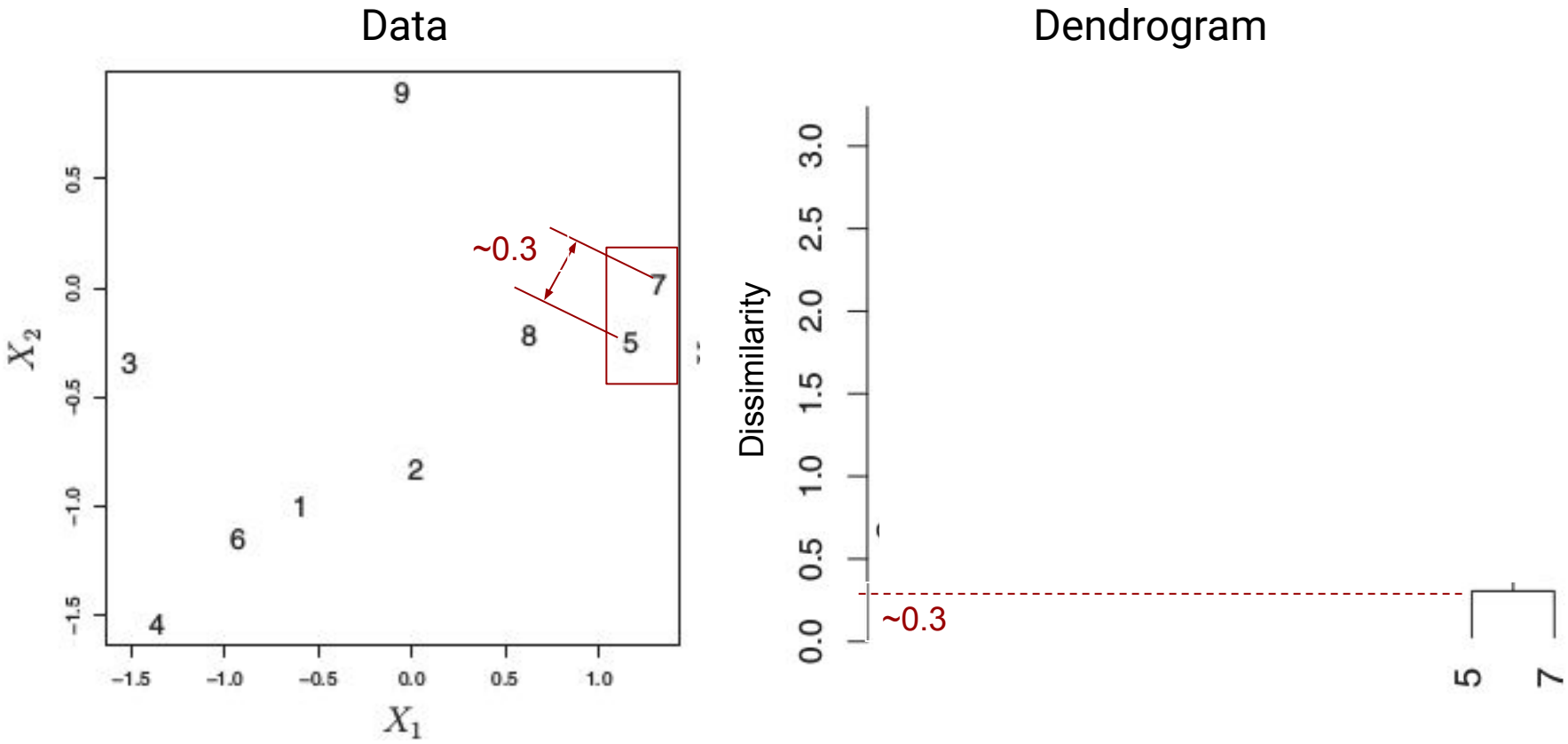
Data



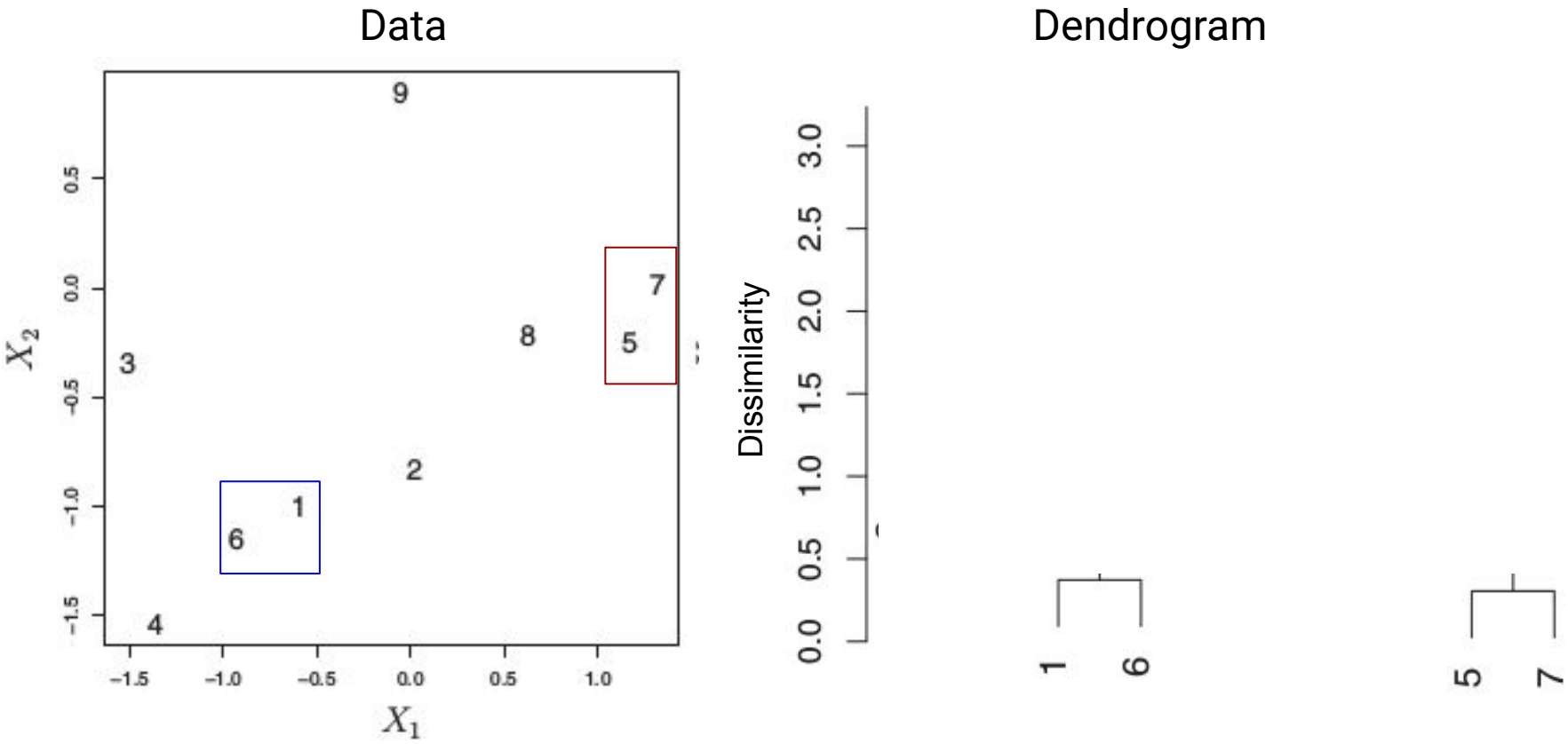
Dendrogram



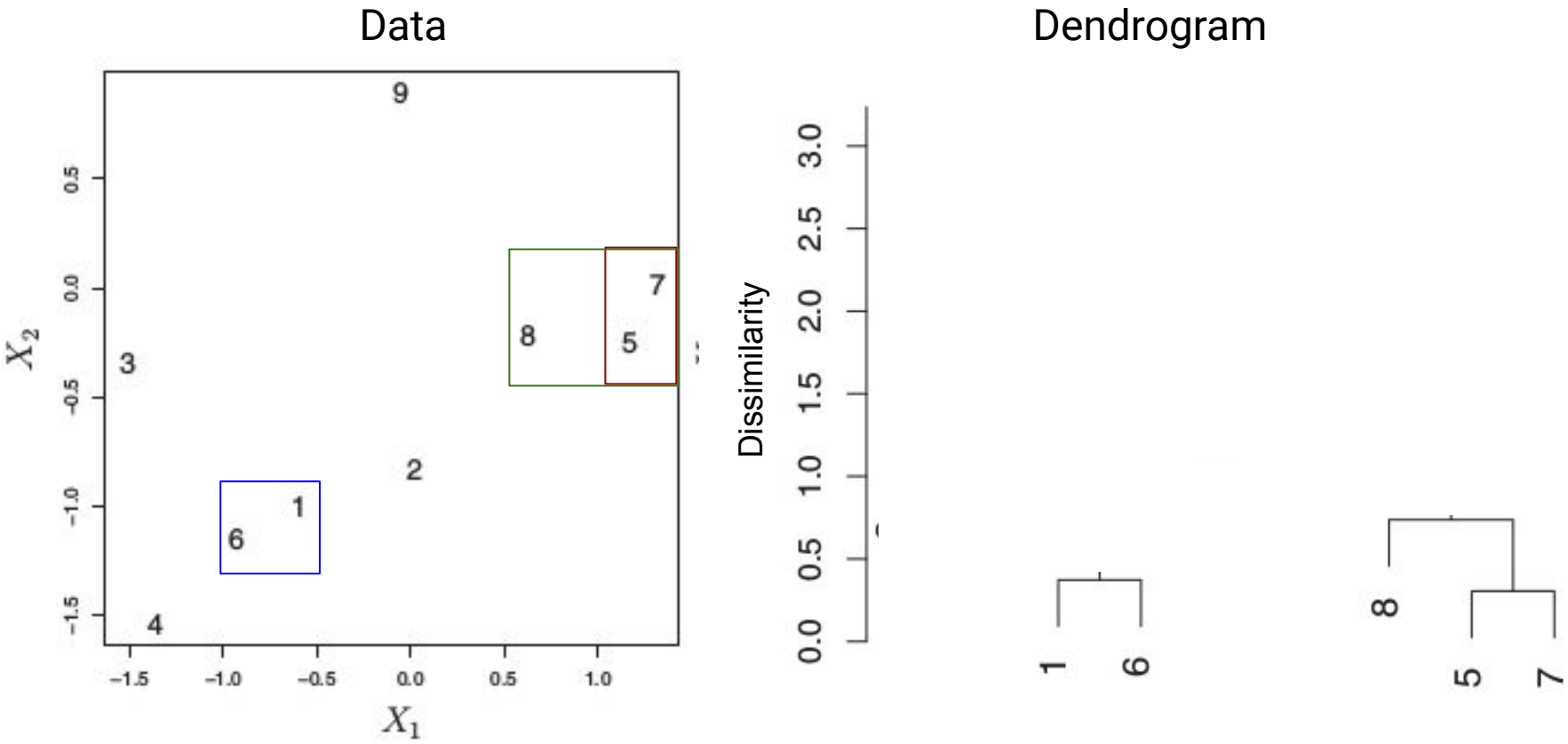
Hierarchical clustering algorithm (visualized)



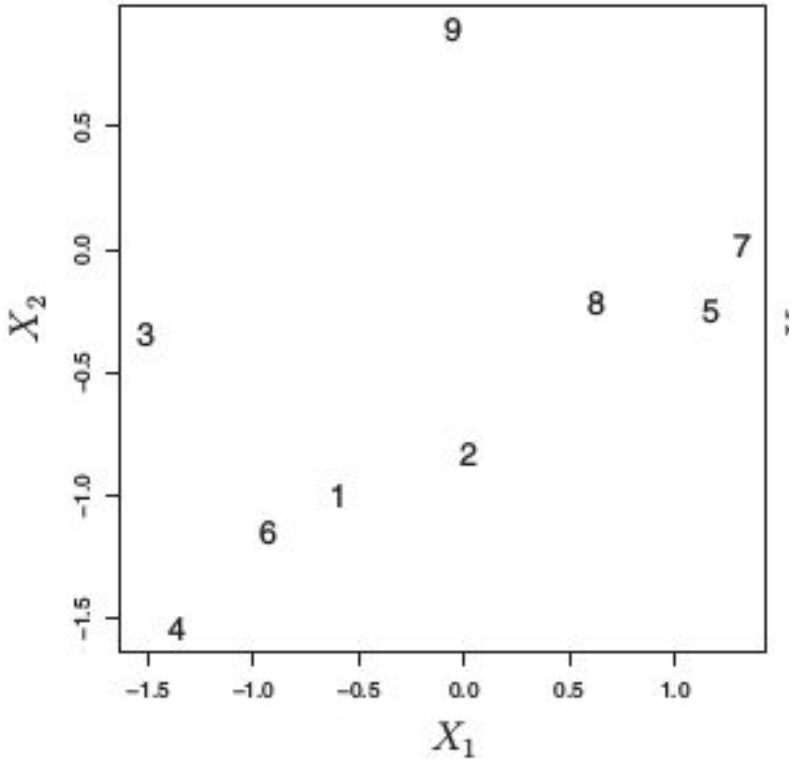
Hierarchical clustering algorithm (visualized)

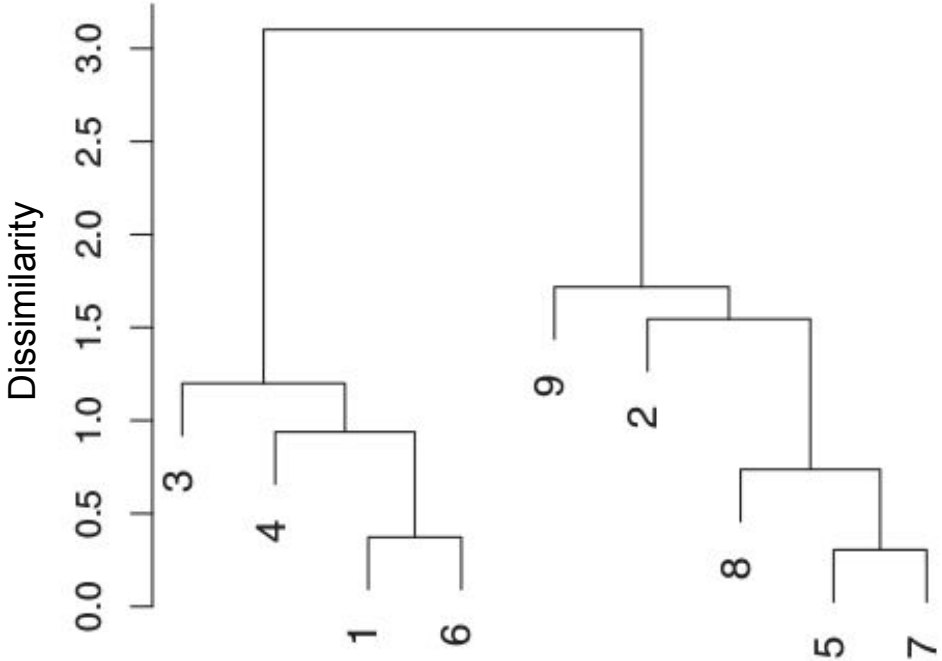


Hierarchical clustering algorithm (visualized)



Hierarchical clustering algorithm



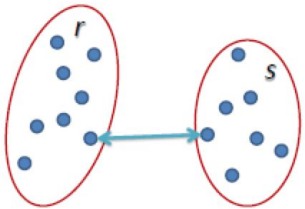


Measures of (dis)similarity between groups

Single Linkage In single linkage hierarchical clustering, the distance between two clusters is defined as the *shortest* distance between two points in each cluster.

"Nearest neighbor"

Drawback: Chaining - several clusters may be joined to just because of a few close cases

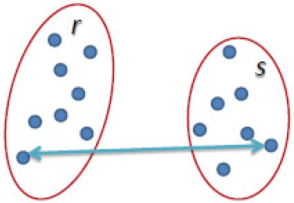


$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

Complete Linkage In complete linkage hierarchical clustering, the distance between two clusters is defined as the *longest* distance between two points in each cluster.

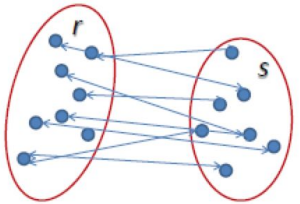
"Farthest neighbor"

Drawback: Cluster outliers prevent otherwise close clusters from merging.



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Average Linkage In average linkage hierarchical clustering, the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

Average and complete are most common

http://www.saedsayad.com/clustering_hierarchical.htm

Choice of linkage matters

(clusters based on same data)

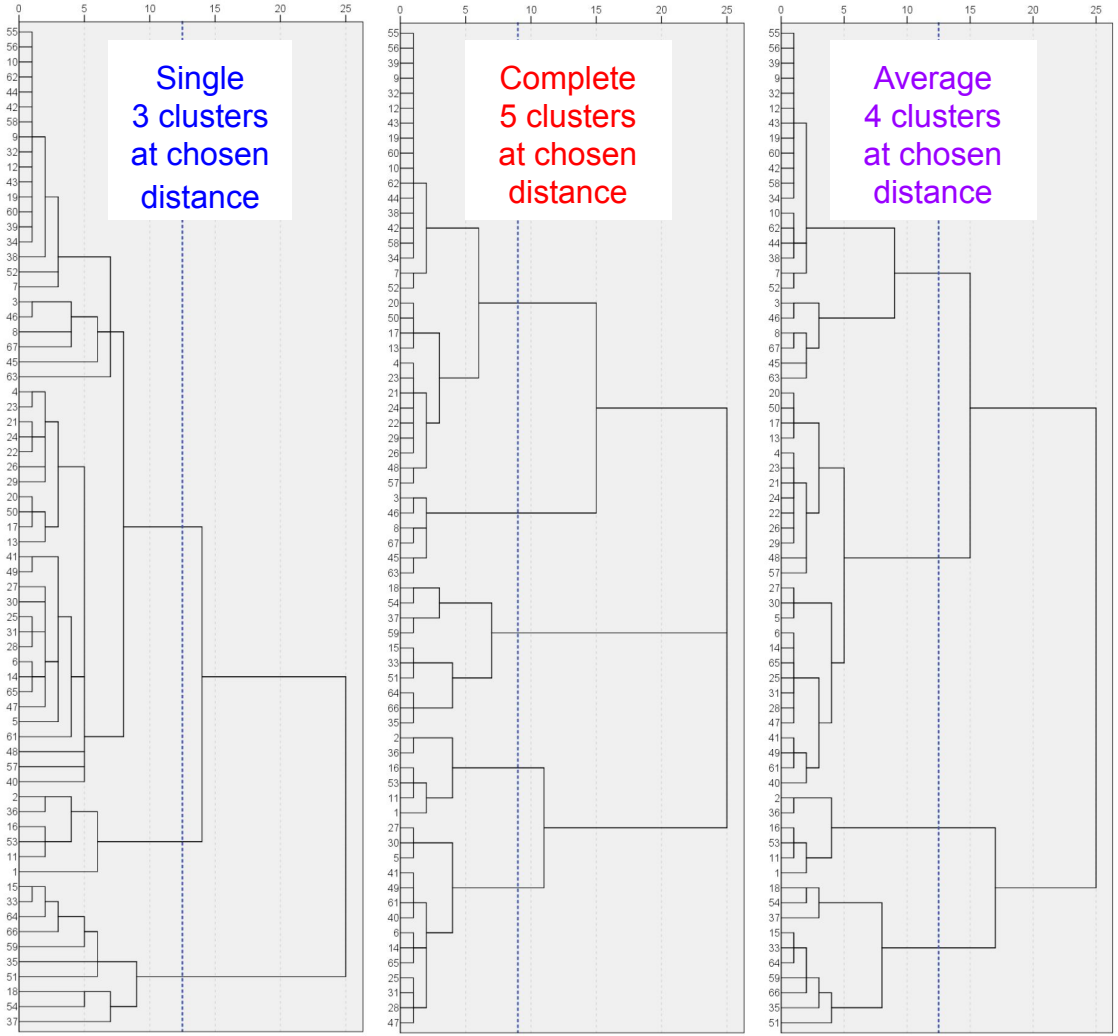
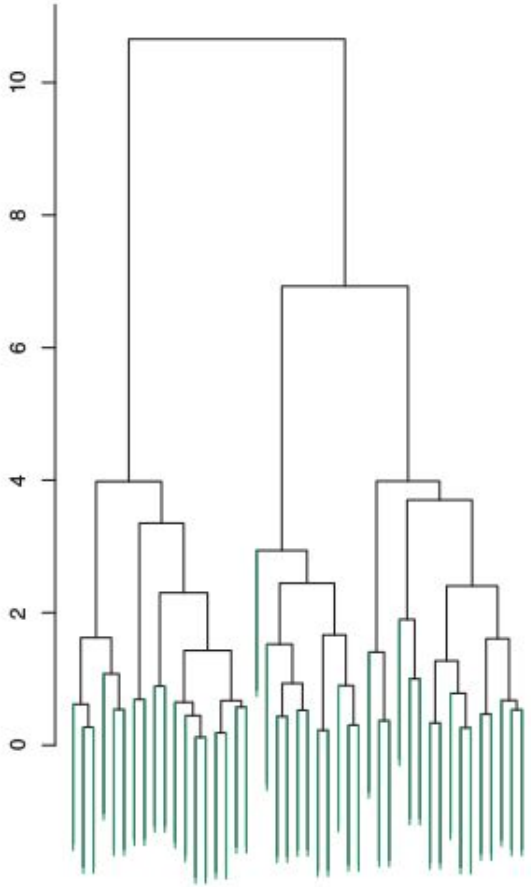
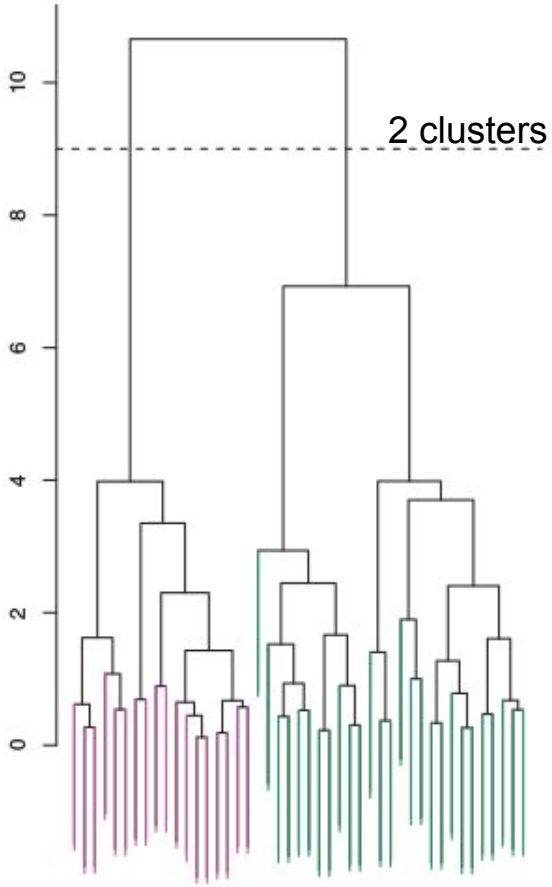


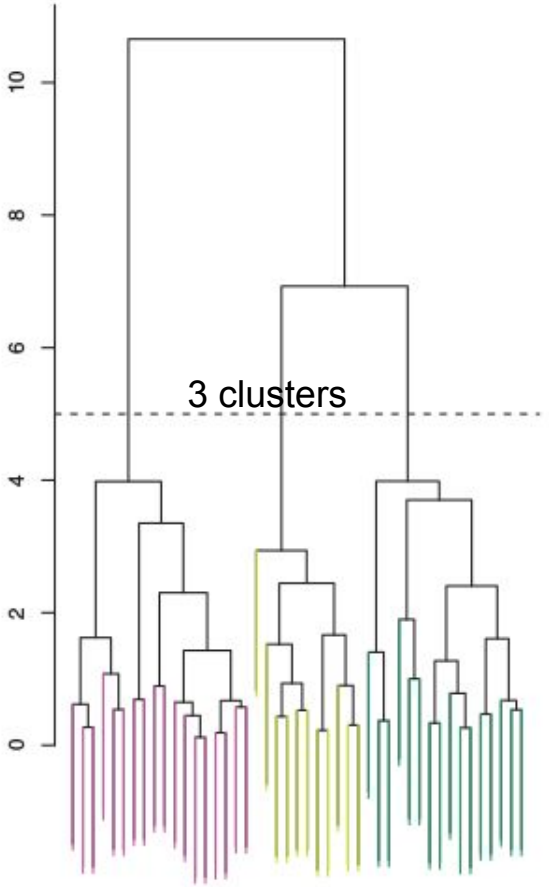
Chart from :

Yim, O. and K.T. Ramdeen, “Hierarchical Cluster Analysis: Comparison of Three Linkage Measures and Application to Psychological Data” The Quantitative Methods for Psychology, vol. 11, no. 1, 2015.

Choice of clusters - where you make the cut







Recap: Learning Objectives

galvanize

- Introduce Unsupervised Learning, compare to Supervised
- Intro to clustering
- Use cases for clustering algorithms
- K-means algorithm:
  - How to code
  - Centroid initialization
  - Stopping Criteria
  - Evaluating the algorithm (metrics)
  - How to choose k
- Hierarchical Clustering
  - Dendrogram