CME 250:
Introduction to Machine Learning

Lecture 7:
Unsupervised Learning

Sherrie Wang
sherwang@stanford.edu

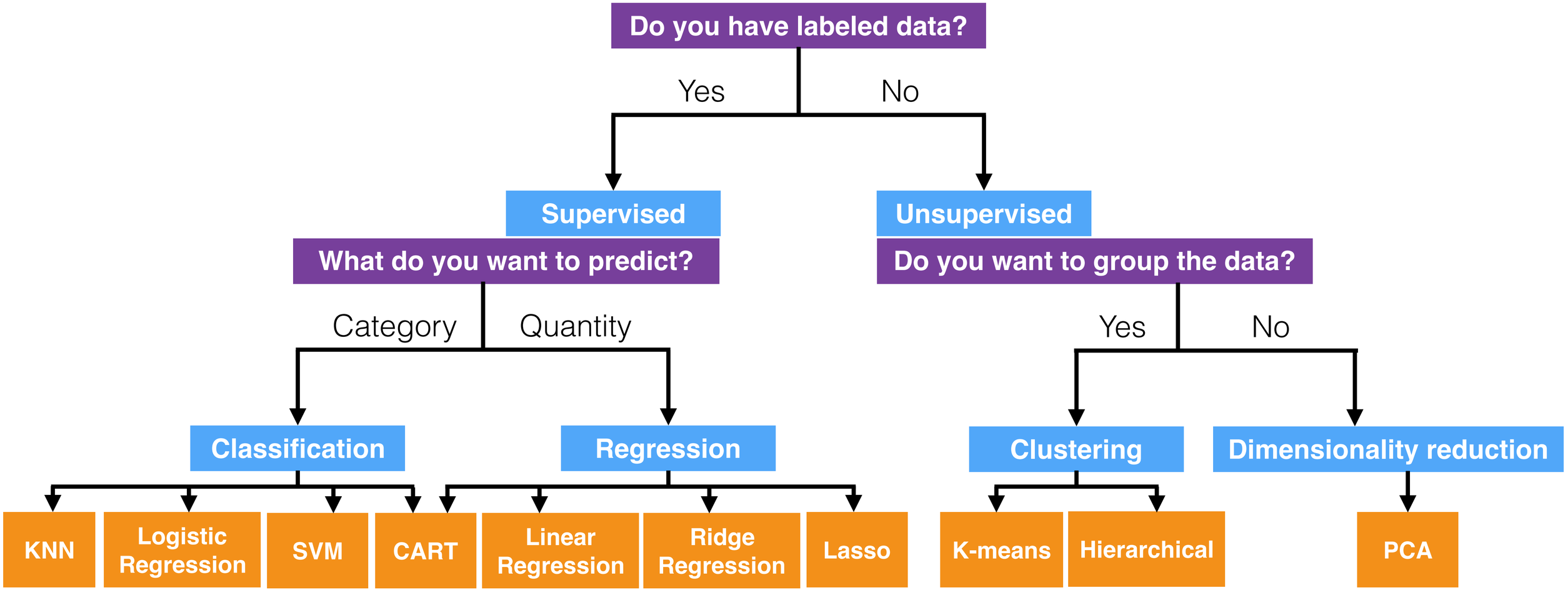
ICME

Agenda

Slides are online at cme250.stanford.edu

- Clustering methods
  - K-means clustering
  - Hierarchical clustering
- Dimensionality reduction
  - PCA

Machine Learning Methods



Unsupervised Learning

Recall: A set of statistical tools for data that only has features/input available, but no response.

In other words, we have X's but no labels y.

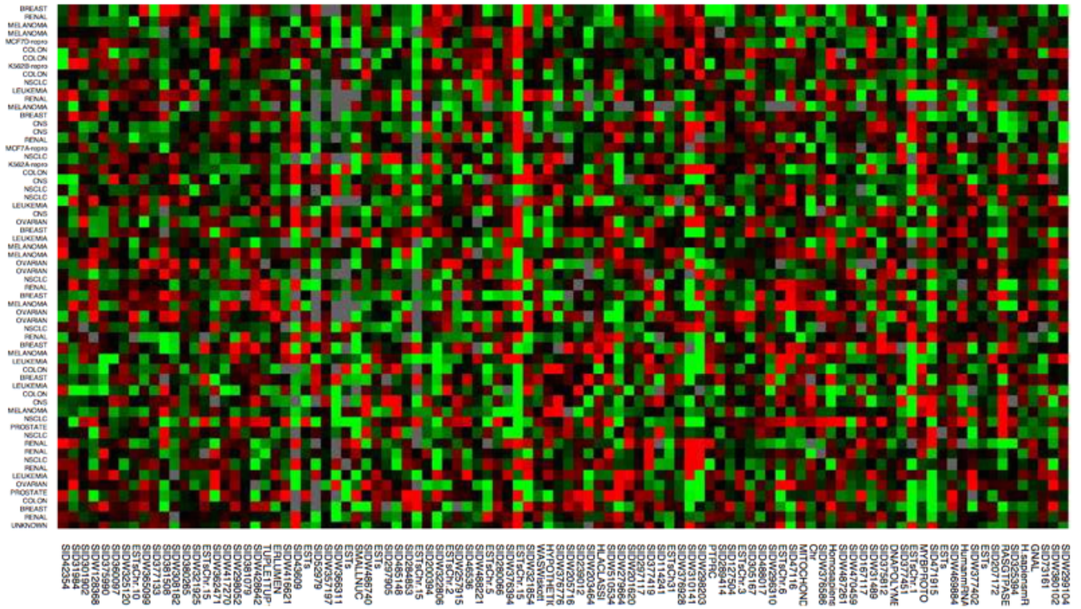
Goal: Discover interesting patterns/properties of the data.

- E.g. for visualizing or interpreting high-dimensional data.

Unsupervised Learning

Example applications:

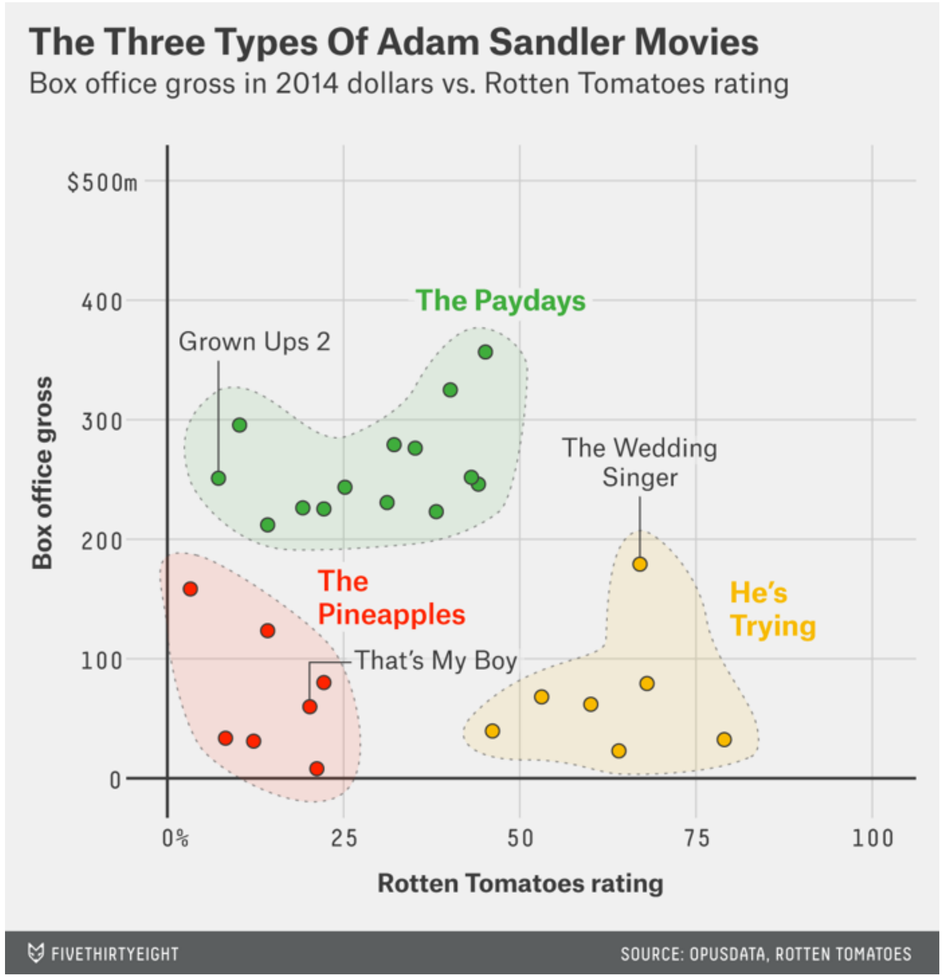
- Given tissue samples from n patients with breast cancer, identify unknown subtypes of breast cancer.
- Gene expression experiments have thousands of variables. Represent the data using a smaller set of features for visualization and interpretation.



Unsupervised Learning

Example applications:

- Document clustering: identify sets of documents about the same topic.
- Given high-dimensional facial images, find a compact representation as inputs for a facial recognition classifier.



Challenges of Unsupervised Learning

Why is unsupervised learning challenging?

- Exploratory data analysis — goal is not always clearly defined
- Difficult to assess performance — “right answer” unknown
- Working with high-dimensional data

Types of Unsupervised Learning

Two approaches:

- **Cluster analysis**

  - For identifying homogenous subgroups of samples

- **Dimensionality reduction**

  - For finding a low-dimensional representation to characterize and visualize the data

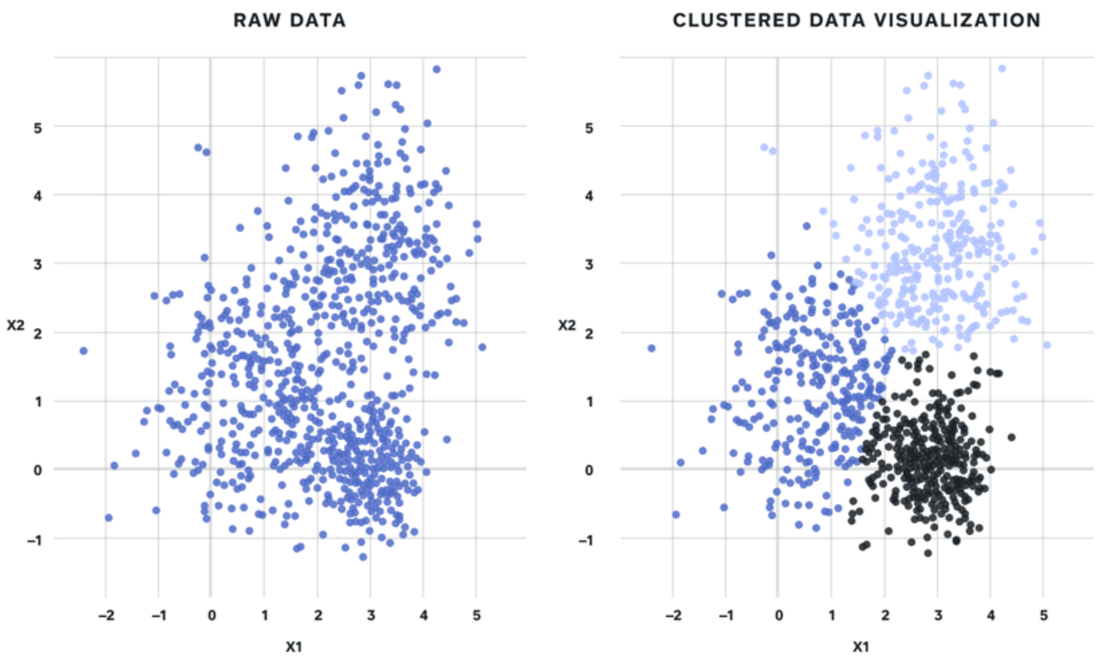
Cluster Analysis

Clustering

A set of methods for finding subgroups within the dataset.

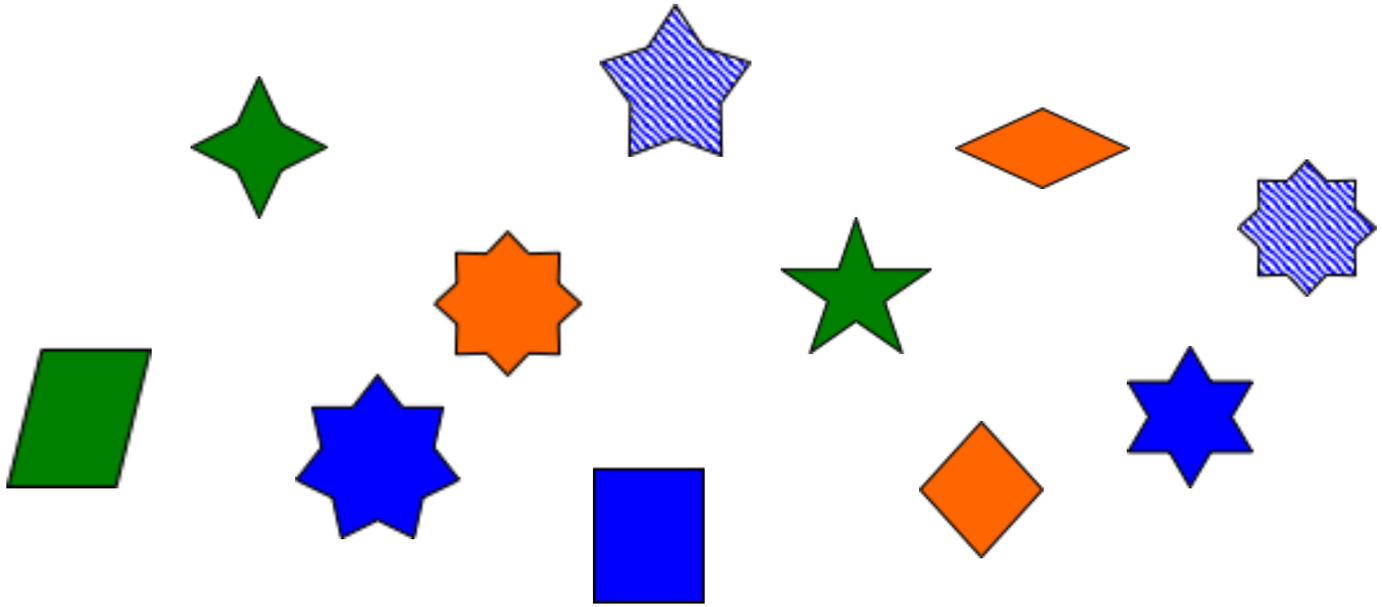
- Observations should share common characteristics within the same group, but differ across groups.

- Groupings are determined from attributes of the data itself — differs from classification.

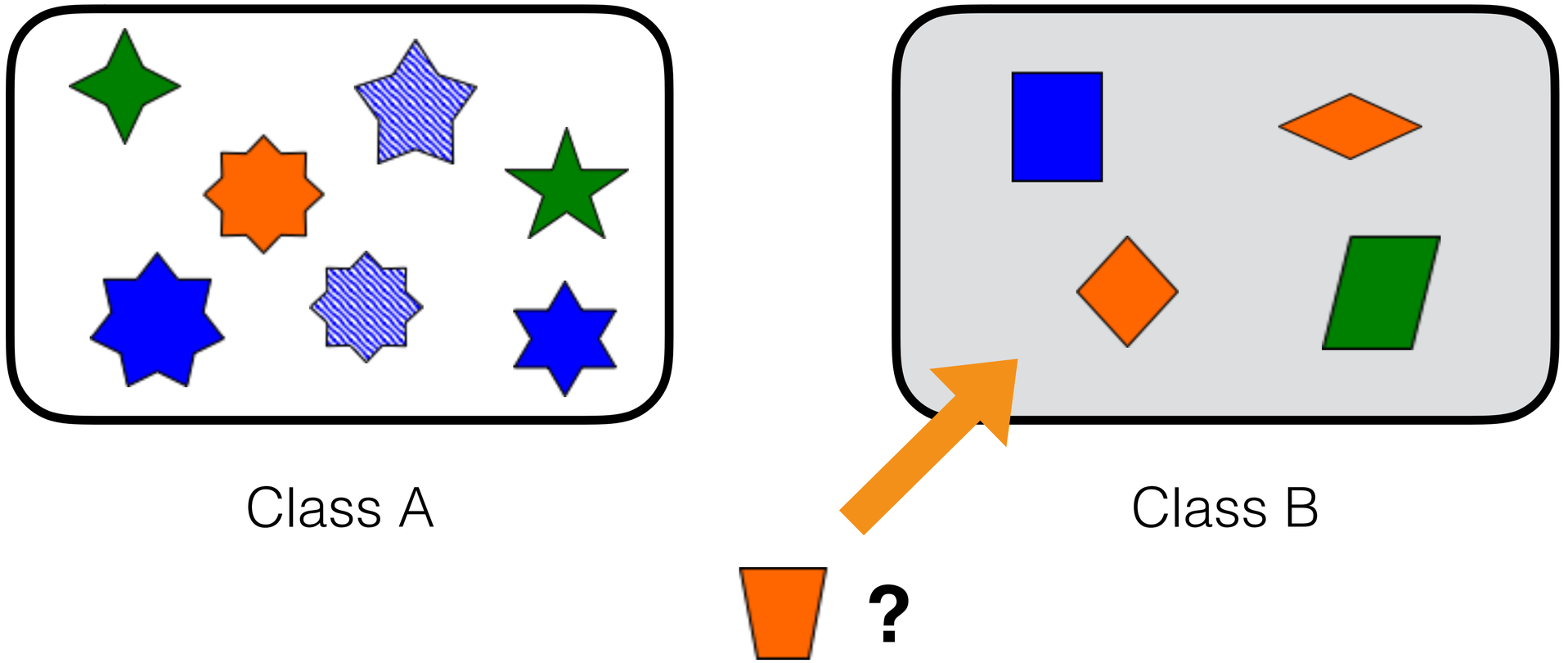


https://medium.com/square-corner-blog/so-you-have-some-clusters-now-what-abfd297a575b

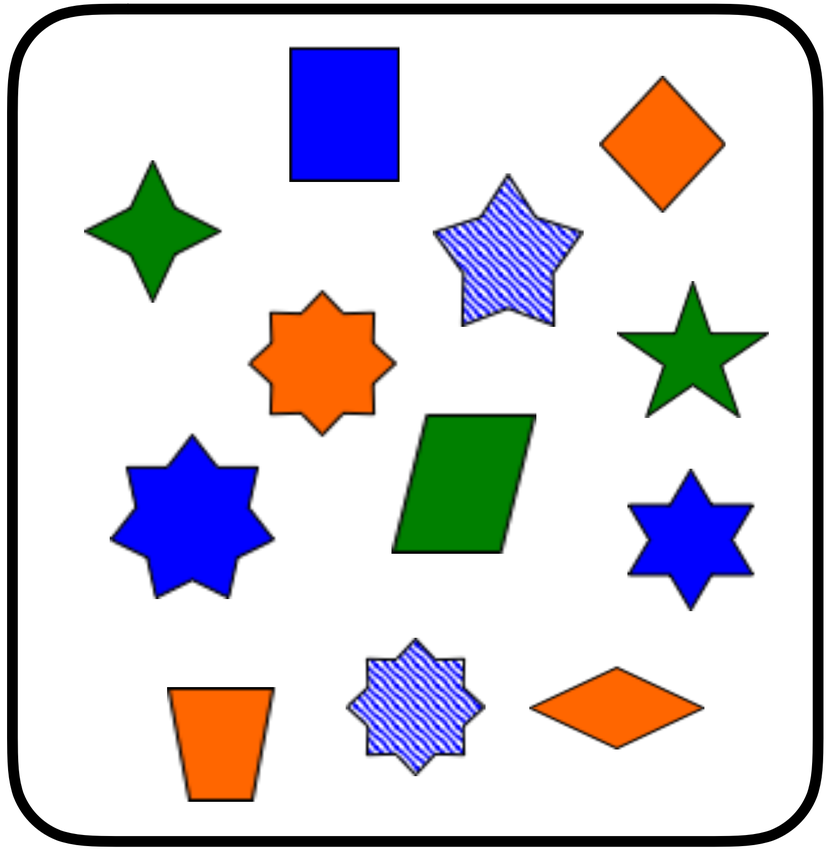
Clustering vs. Classification



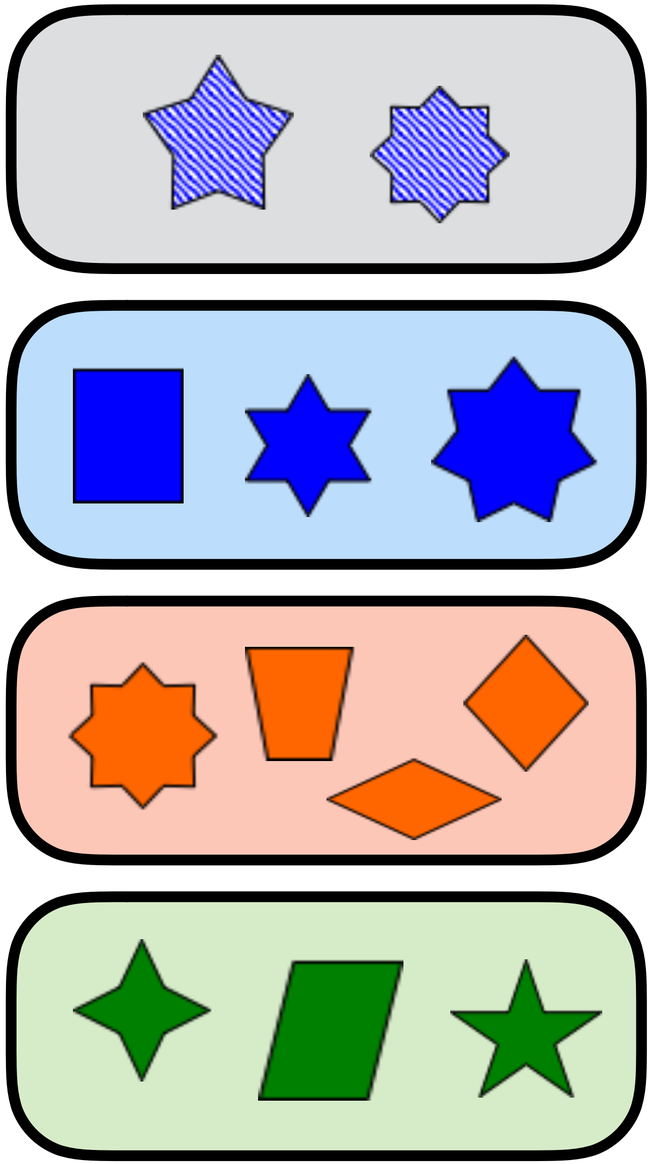
Classification



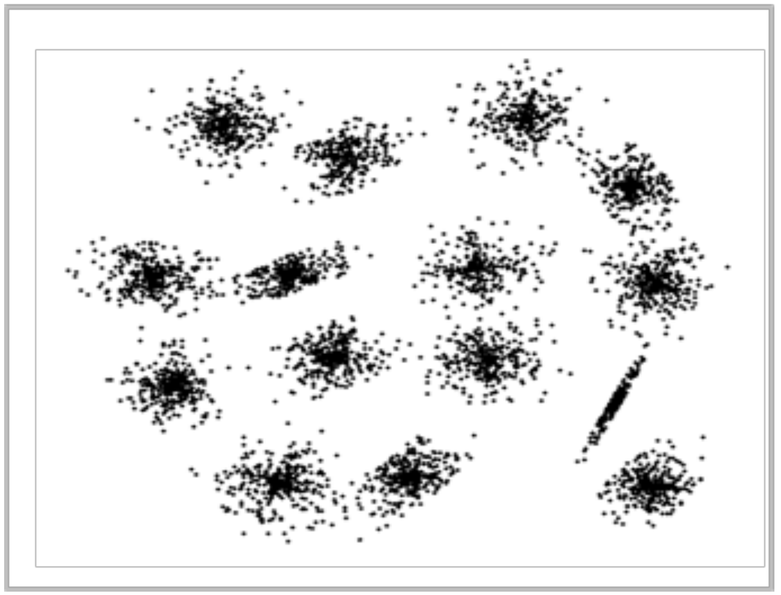
Clustering



Dataset



Clustering



http://cs.joensuu.fi/sipu/datasets/

Types of Clustering

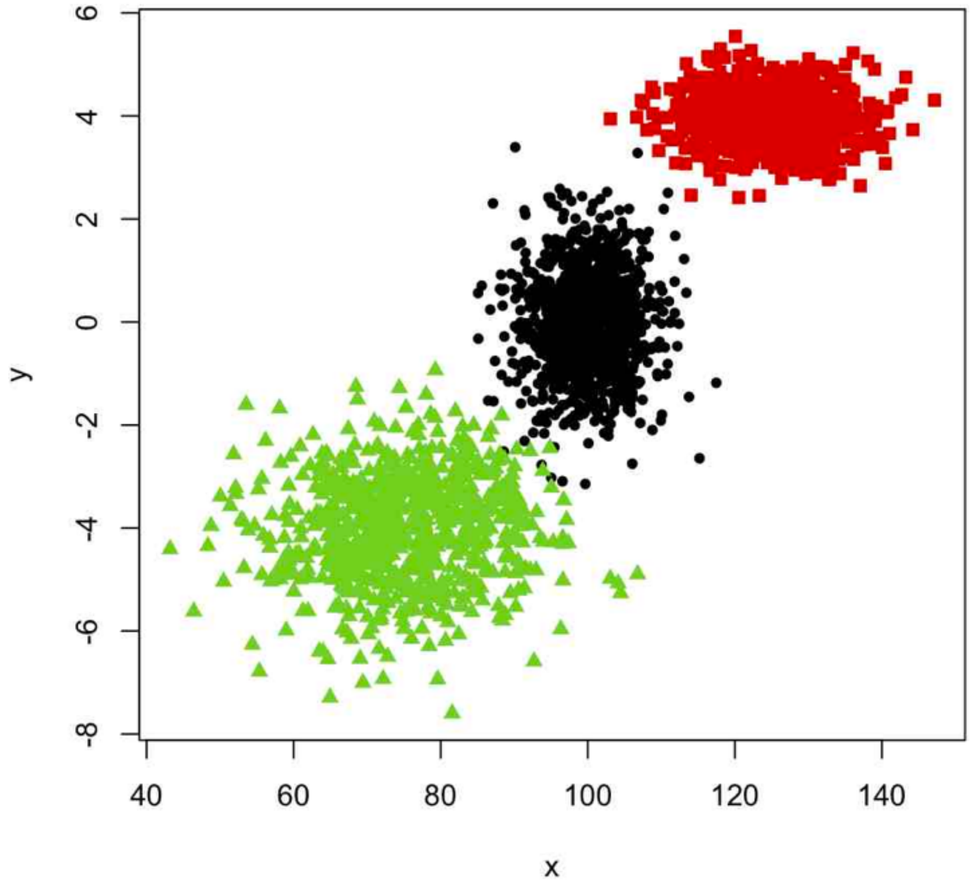
- **Centroid-based clustering**
- **Hierarchical clustering**
- **Model-based clustering**
  - Each cluster is represented by a parametric distribution
  - Dataset is a mixture of distributions
- **Hard vs. soft/fuzzy clustering**
  - Hard: observations divided into distinct clusters
  - Soft: observations may belong to more than one cluster

K-means Clustering

Groups data into K clusters that satisfy two properties.

1. Each observation belongs to at least one of the K clusters.

2. Clusters are non-overlapping. No observation belongs to more than one cluster.



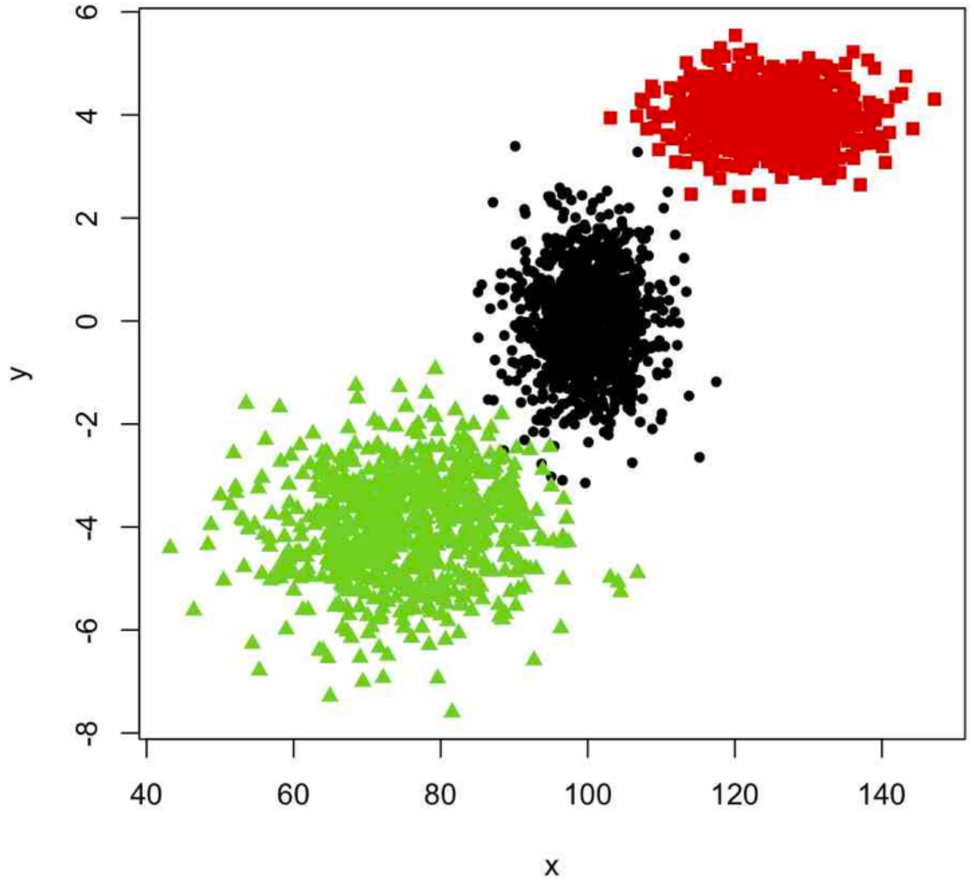
K-means Clustering

A good clustering is one for which the *within-cluster variation* is as small as possible.

Denote each cluster by C_k, and let $W(C_k)$ be a measure of the within-cluster variation.

K-means aims to solve

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



K-means Clustering

How to measure within-cluster variation?

The most common choice is squared Euclidean distance.

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

Which means overall we solve

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

K-means Clustering

It turns out that this optimization problem is difficult to solve, as it is discrete and there are nearly K^n ways to split n samples into K clusters.

In practice, use an iterative algorithm that finds a local minimum to this optimization.

K-means Clustering Algorithm

1. Initialize each observation to a cluster by randomly assigning a cluster, from 1 to K, to each observation.

2. Iterate until the cluster assignments stop changing:

    a. For each of the K clusters, compute the cluster centroid. The k-th cluster centroid is the vector of the p feature means for the observations in the k-th cluster.

    b. Assign each observation to the cluster whose centroid is closest (using Euclidean distance as the metric).

K-means Clustering Iterations

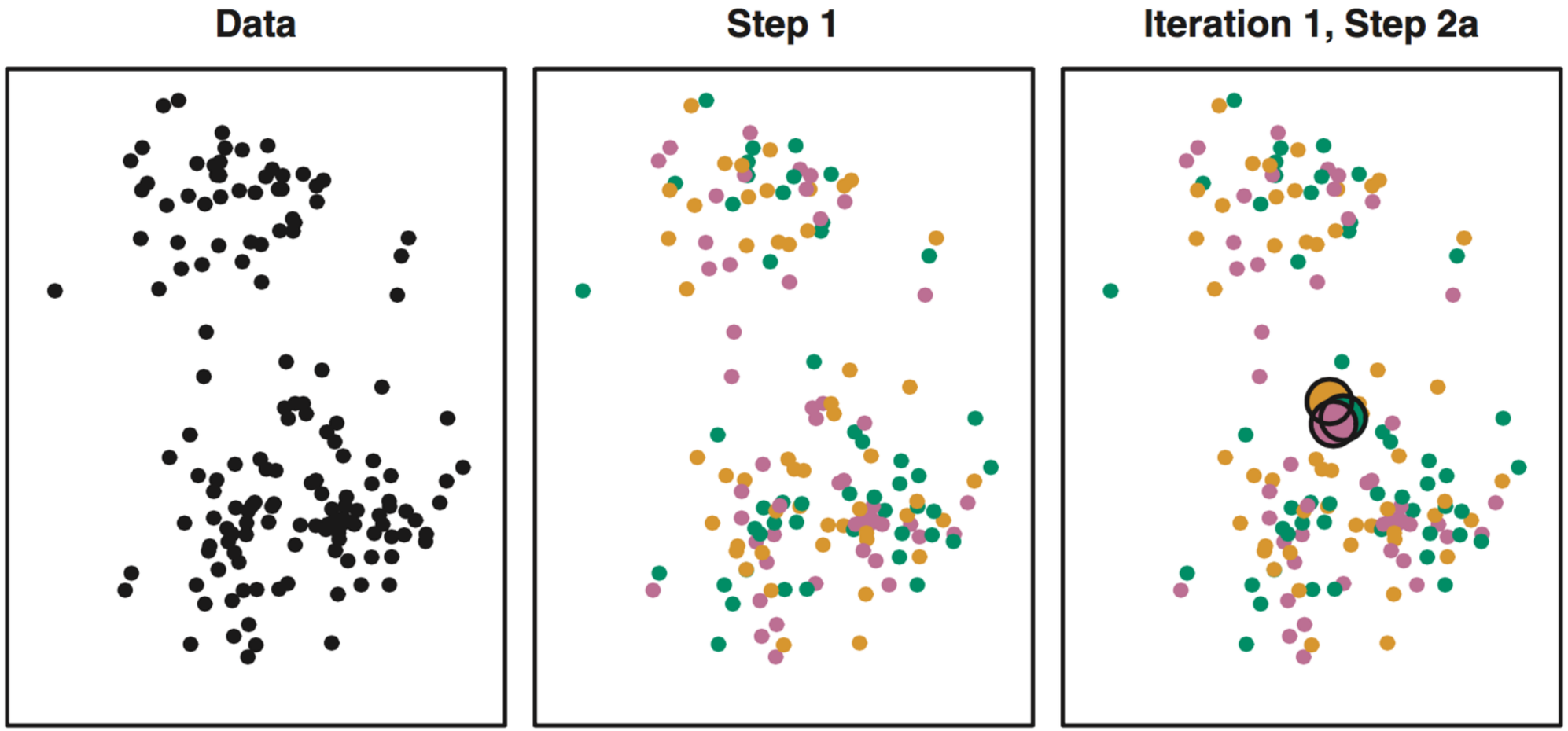


FIGURE 10.6, ISL (8th printing 2017)

K-means Clustering Iterations

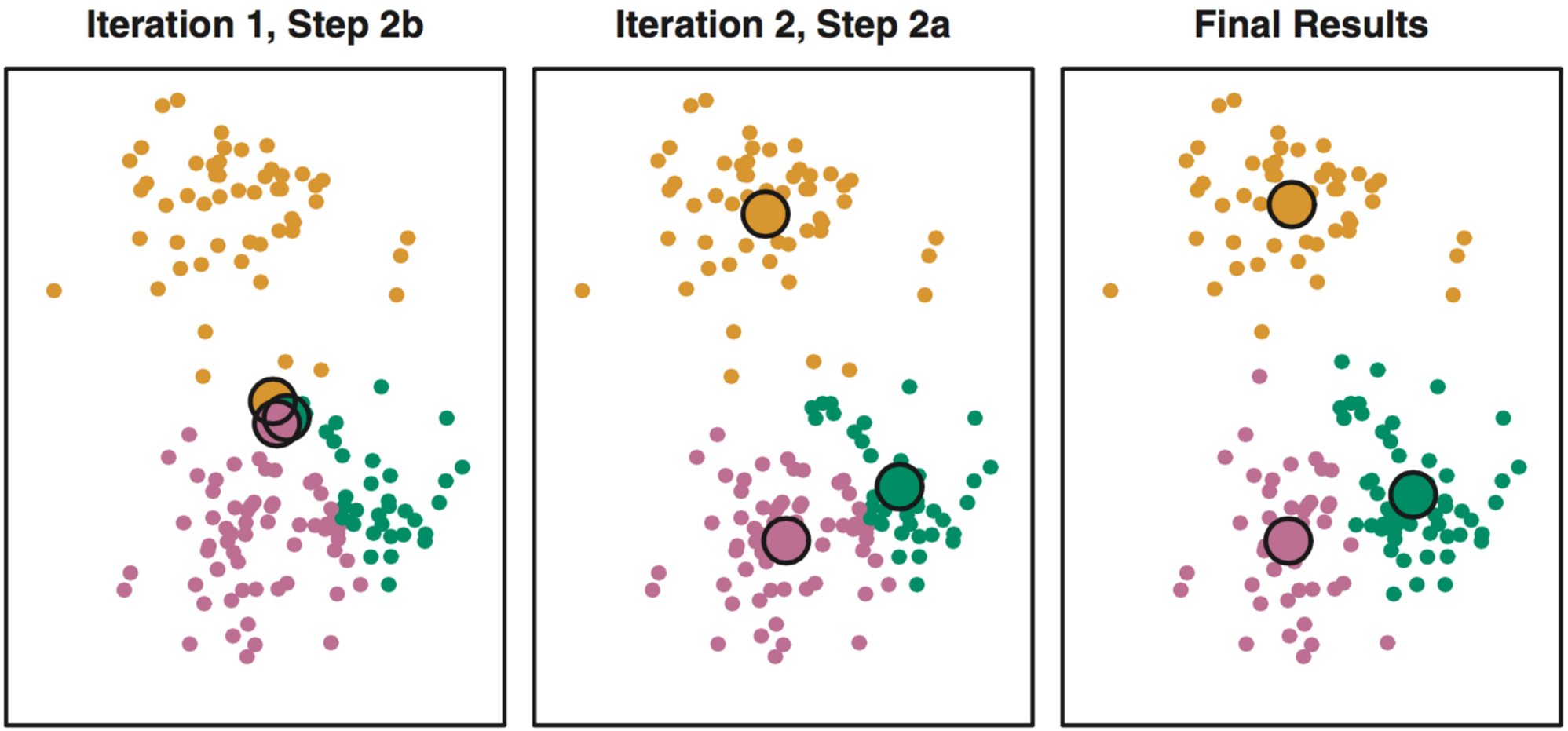
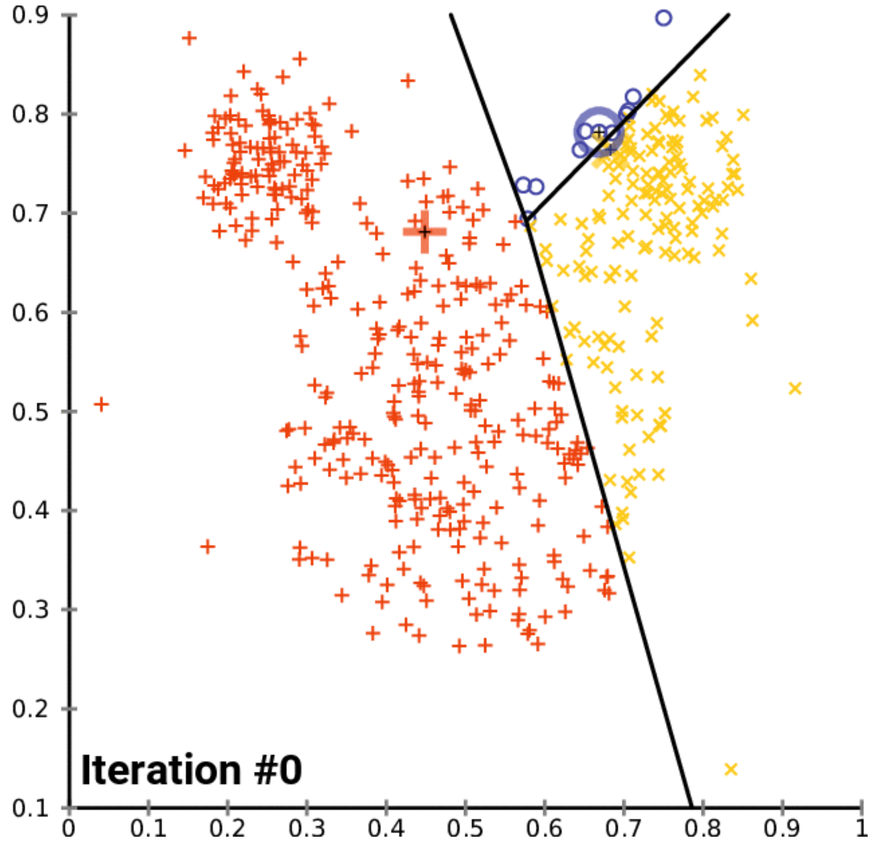


FIGURE 10.6, ISL (8th printing 2017)

K-means Clustering Animation



K-means Clustering Properties

It can be shown that the value of the objective function will never increase at each iteration of k-means.

Since the algorithm finds local minima, however, it will result in different clusters with different initializations.

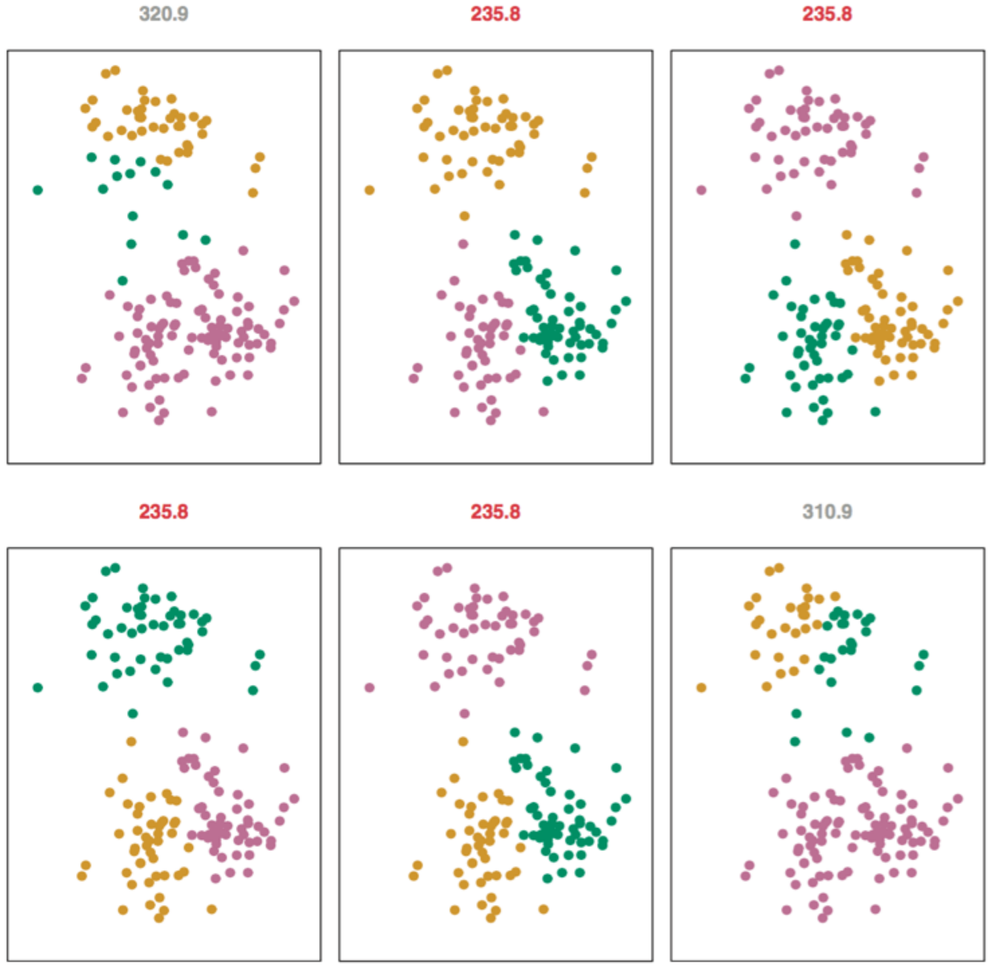


FIGURE 10.7, ISL (8th printing 2017)

K-means Pros and Cons

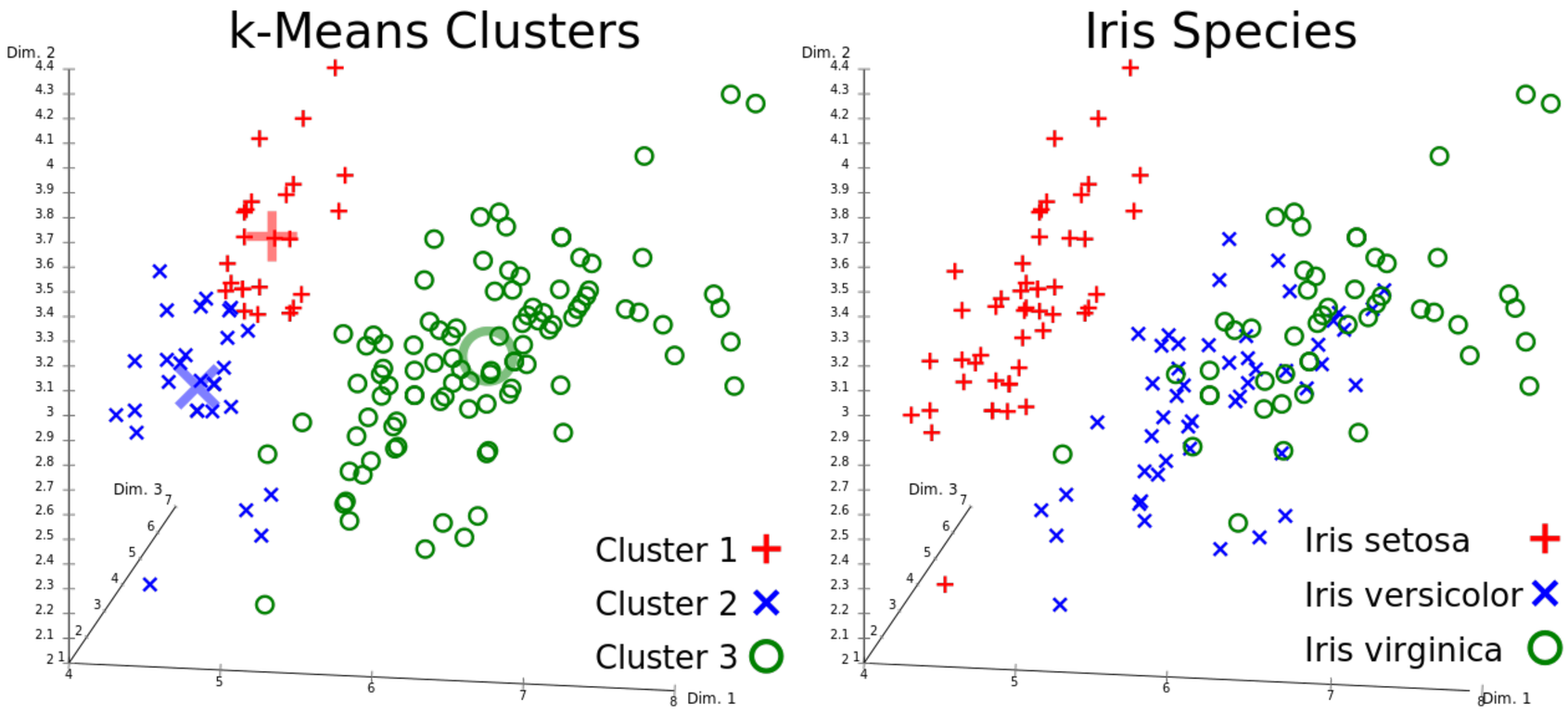
Pros:

- Easy to implement and understand

Cons:

- Not robust to data perturbations and different initializations
- Treats each feature equally, not robust to noise features or different scales of features — looks for in spherical clusters in feature space
- Need to define K before running algorithm

Another K-means Example

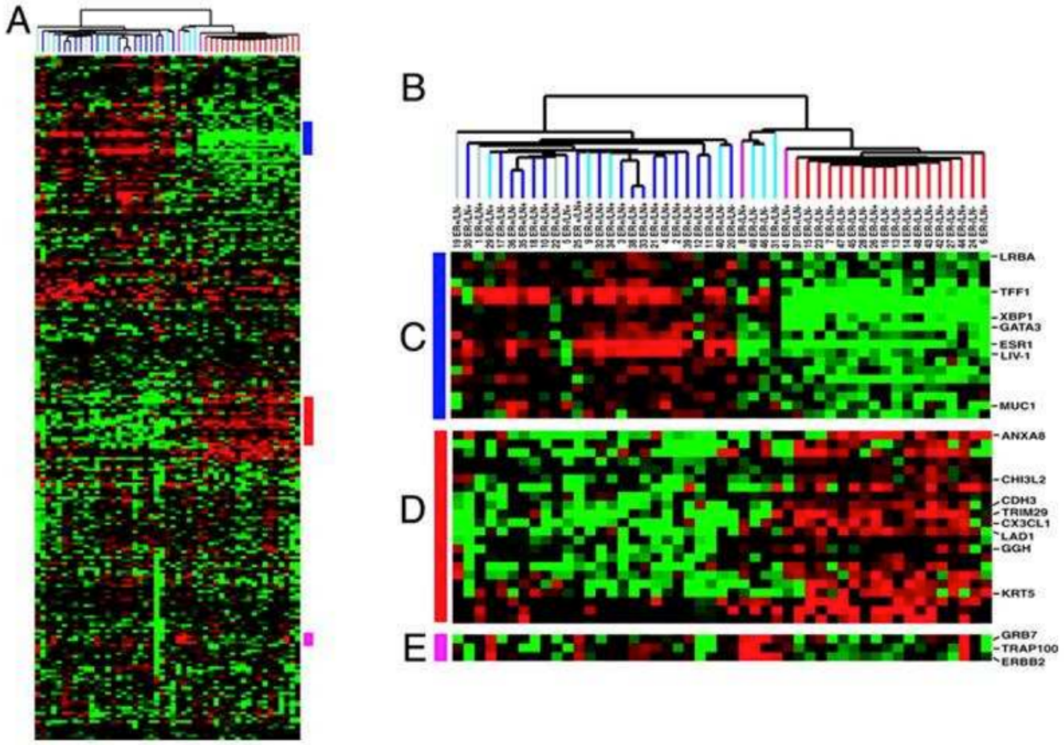


Hierarchical Clustering

Cluster based on distances between observations.

Represented as a tree hierarchy (*dendrogram*) rather than a partition of data.

Does not require committing to a choice of K.



Sørlie, Therese, et al. (2003) "Repeated observation of breast tumor subtypes in independent gene expression data sets," PNAS.

Dendrograms

Each leaf in a dendrogram is a sample/ observation.

As we move up the dendrogram, observations that are similar to each other begin to *fuse* into branches.

Branches then fuse into bigger branches.

Observations that fuse later (near the top of the tree, or root) are more different than observations that fuse earlier.

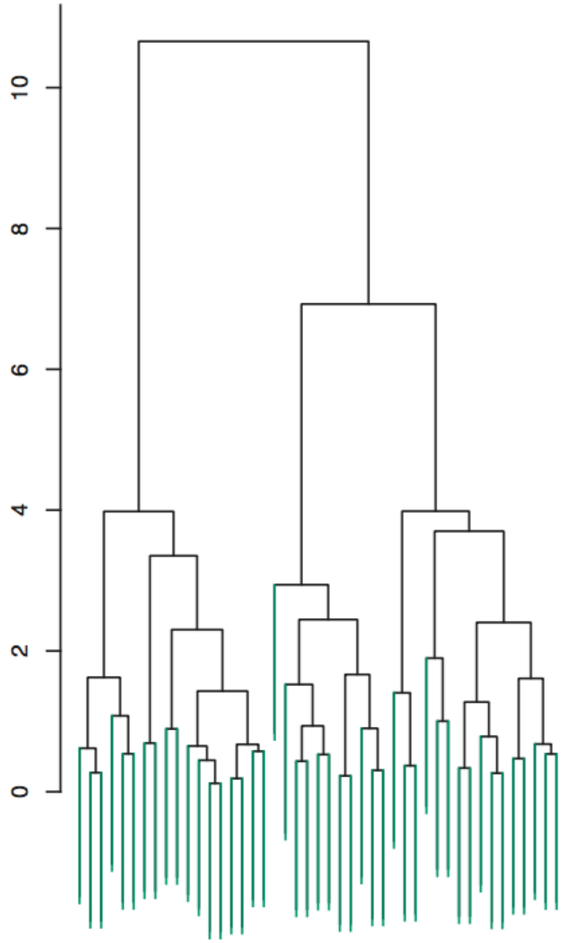


FIGURE 10.9, ISL (8th printing 2017)

Dendrograms

Note that the horizontal distance between observations on a dendrogram is not the appropriate assessment of observation similarity. Instead, look at *vertical* axis where branches are first fused.

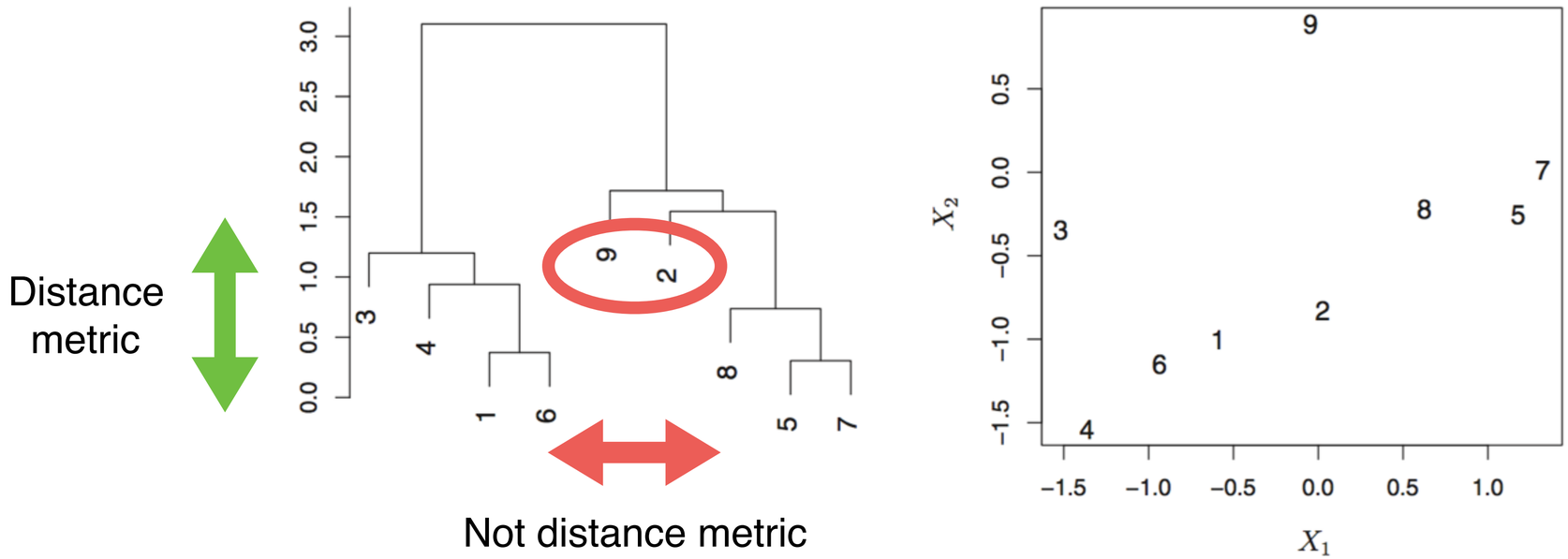


FIGURE 10.10, ISL (8th printing 2017)

Obtaining Clusters

Clusters are created by making a horizontal cut across the dendrogram. Clusters are the separate trees below the cut.

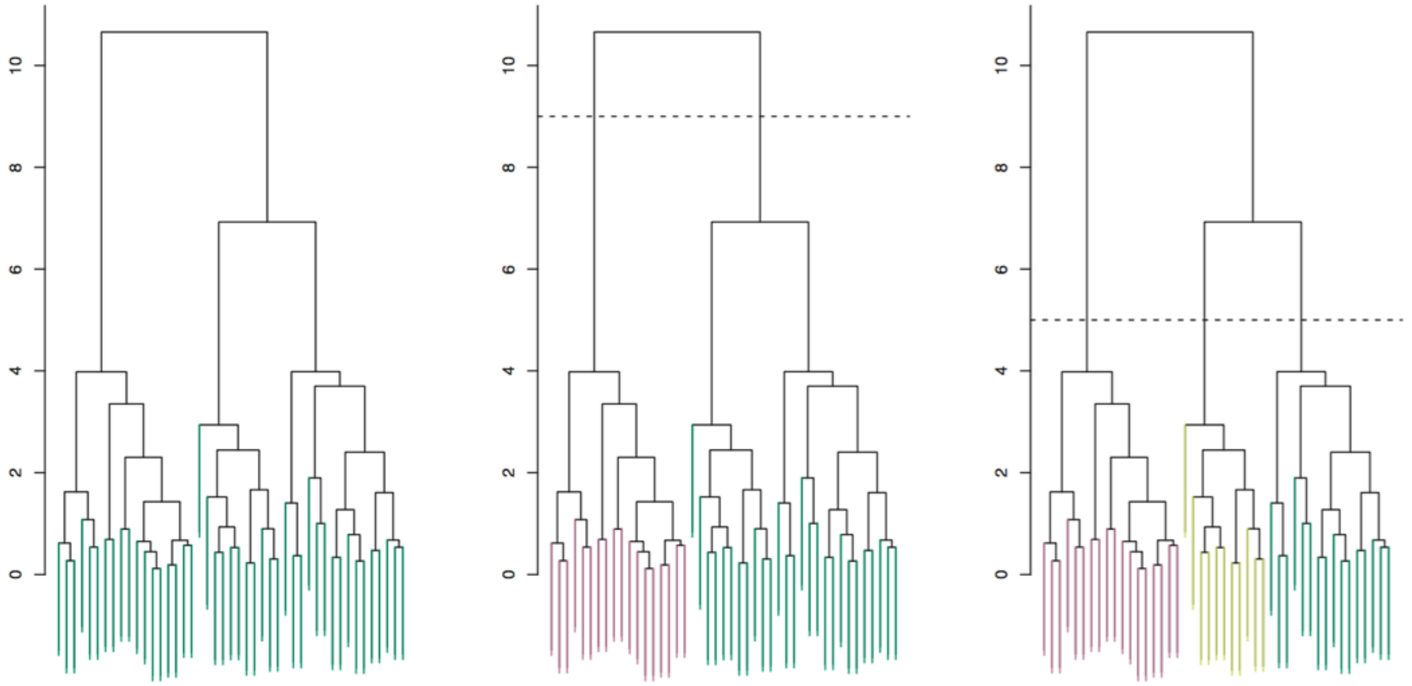


FIGURE 10.9, ISL (8th printing 2017)

Building a Dendrogram

A dendrogram is most commonly built using a *bottom-up* or *agglomerative* algorithm.

We start at the leaves and group observations until we reach the root containing the entire dataset.

Like in k-means, we need a measure of similarity. Again, the most common is Euclidean distance.

Hierarchical Clustering Algorithm

1. Initialize each observation to its own cluster.

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

   a. Examine all pairwise *inter-cluster similarities* among the i clusters and identify the pair of clusters that are most similar. Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion occurs.

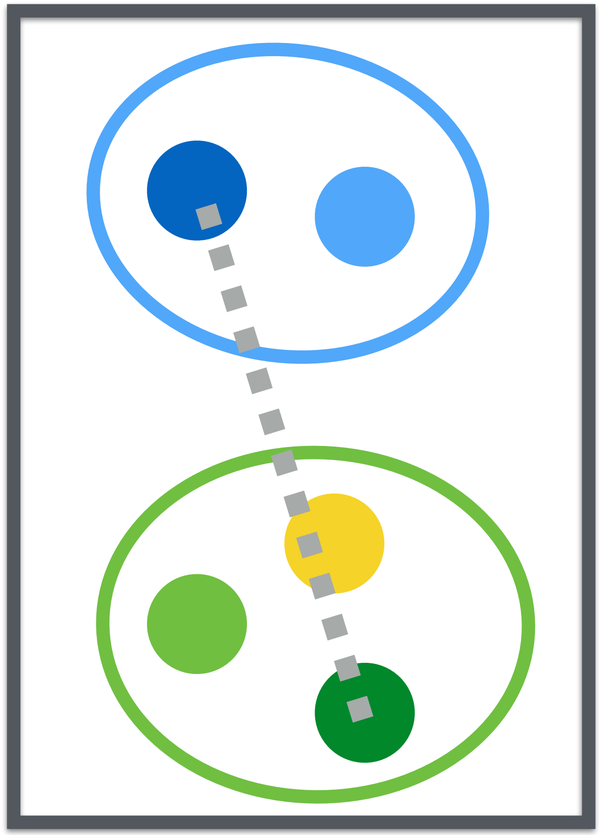
   b. Compute the new pairwise inter-cluster similarities among the $i\text{-}1$ remaining clusters.

Distance Between Groups

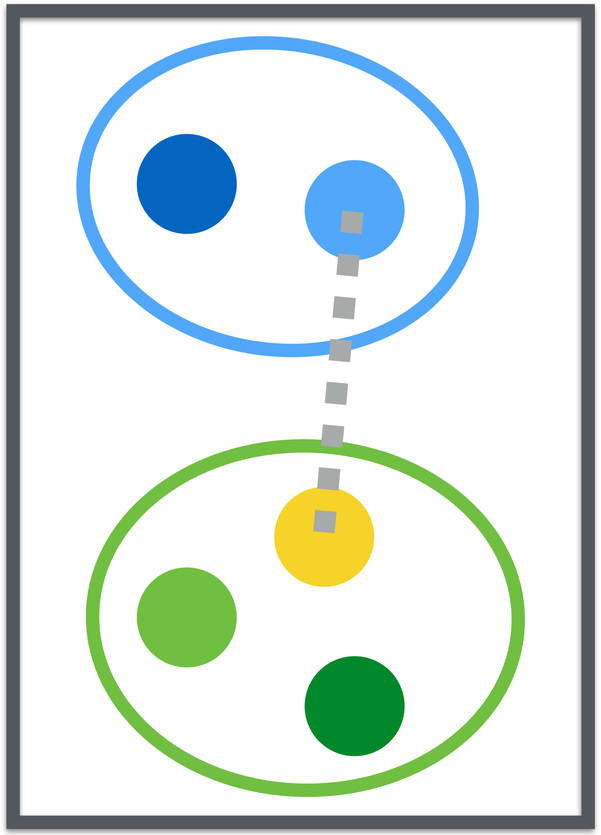
It's easy to compute Euclidean distance between two observations. What is the distance or similarity between two groups or clusters of observations?

Linkage: defines the dissimilarity between two groups of observations. Most common types are *complete*, *average*, *single*, and *centroid*.

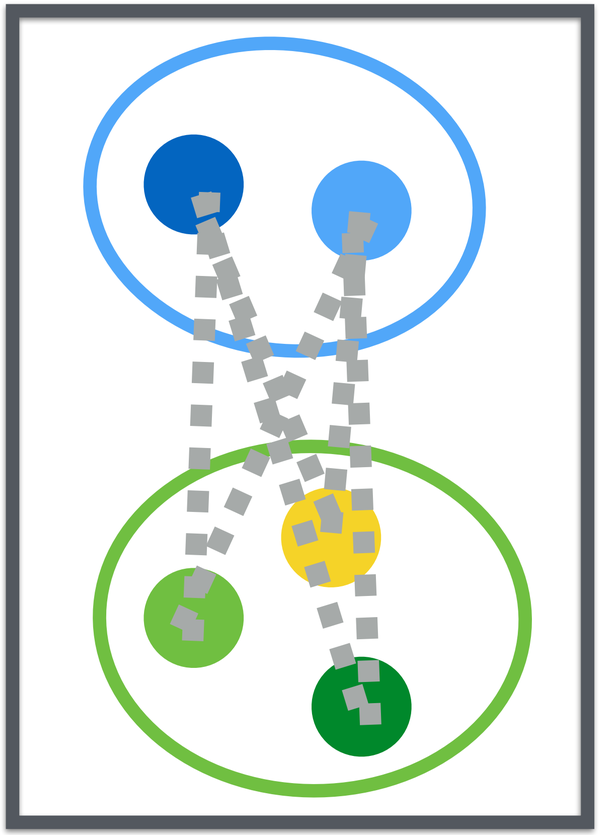
Types of Linkage



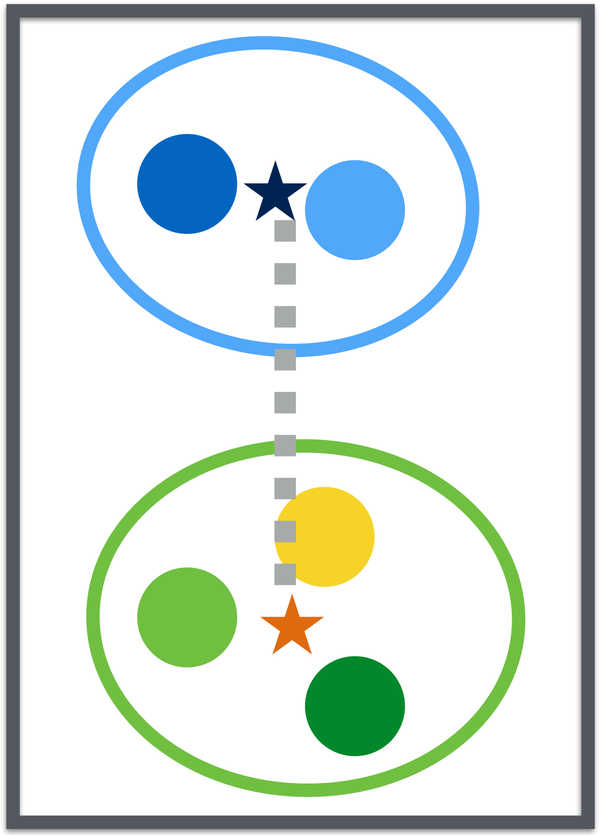
Complete linkage



Single linkage



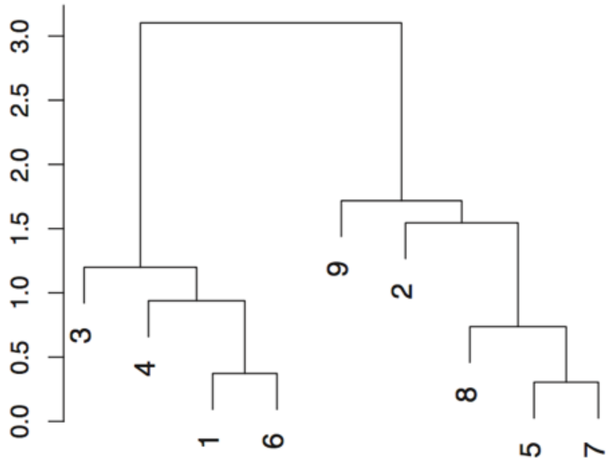
Average linkage



Centroid linkage

Hierarchical Clustering Example

Illustration of the first few steps of the hierarchical clustering algorithm, with complete linkage and Euclidean distance.



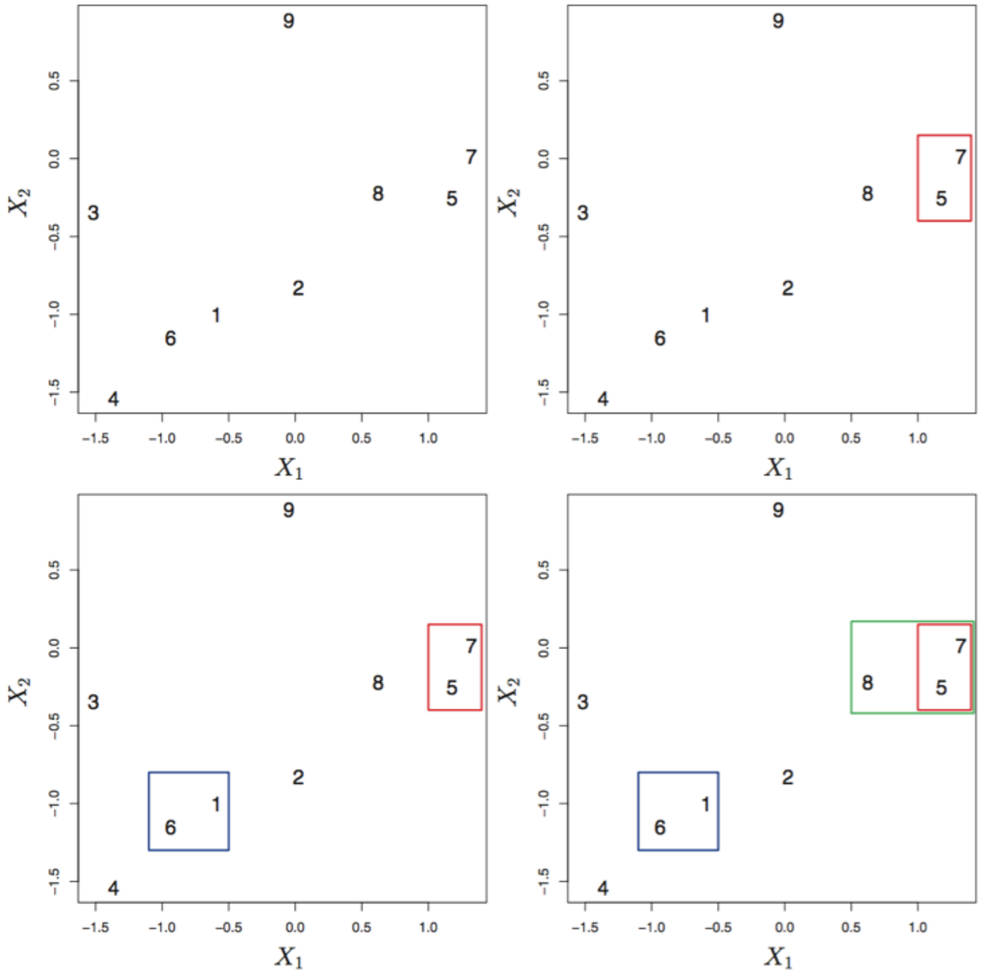


FIGURE 10.11, ISL (8th printing 2017)

Different Linkage, Different Dendrogram

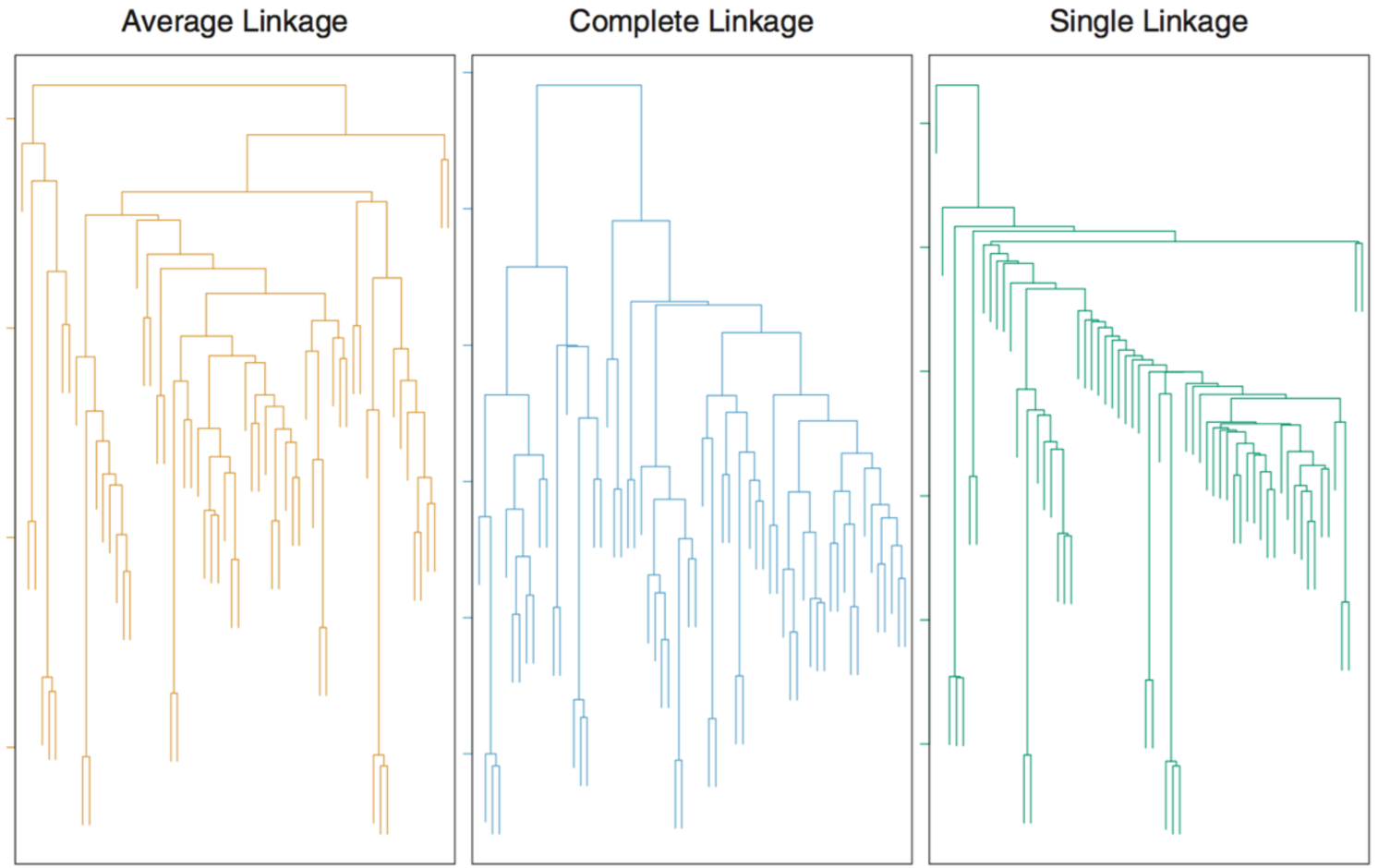


FIGURE 10.12, ISL (8th printing 2017)

Hierarchical Clustering Pros and Cons

Pros:

- Don't have to choose a value of K (number of clusters) before running algorithm

Cons:

- Do have to pick where to cut the dendrogram to obtain clusters
- Sensitive to similarity measure and type of linkage used

Dimensionality Reduction

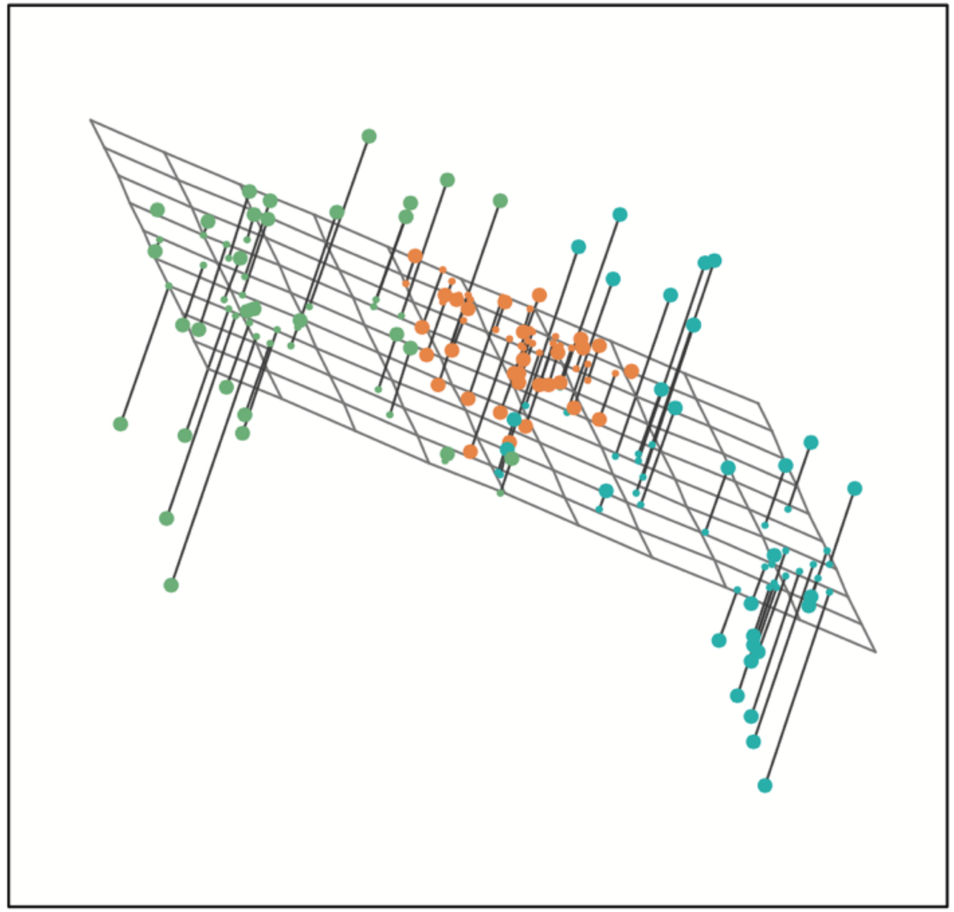
Dimensionality Reduction

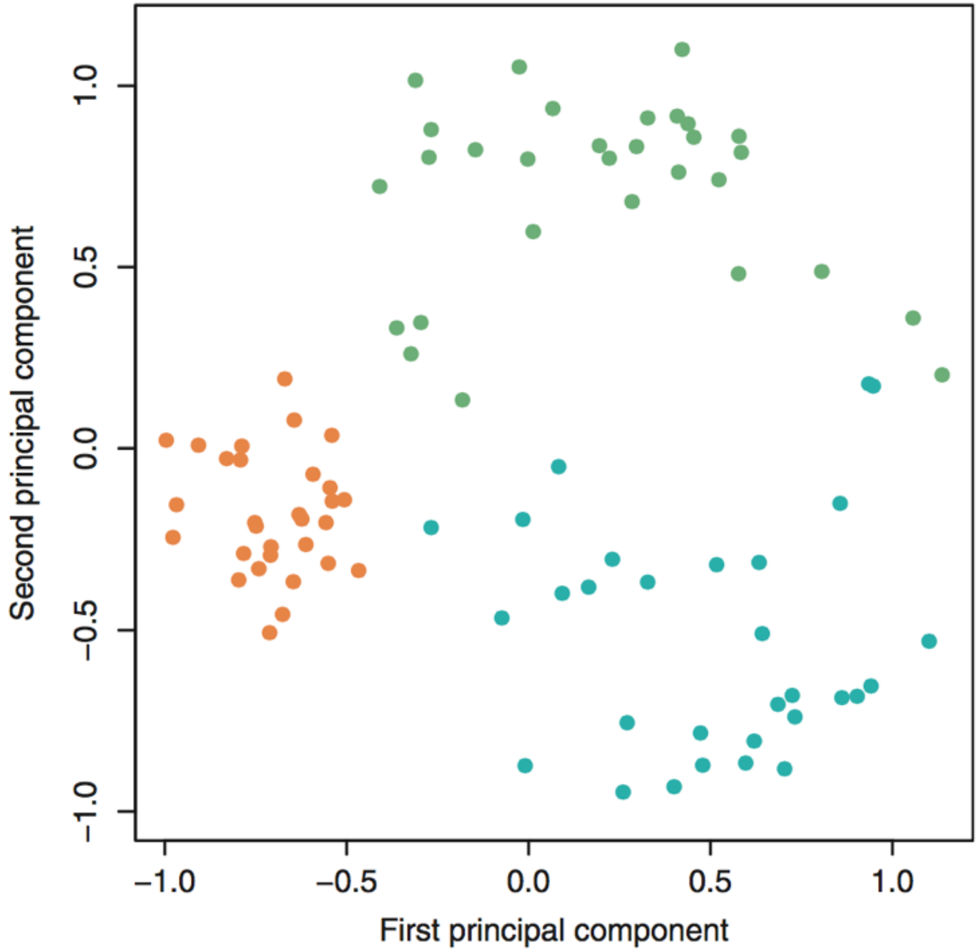
Recall the curse of dimensionality when working in high dimensions.

Dimensionality reduction is the process of reducing the number of features under consideration.

We already saw some examples of this in the lasso and forward/backward selection algorithms. These methods reduce dimensionality by selecting a subset of features. However, they do so using supervision — knowing a response y that is of interest.

Dimensionality Reduction





Principal Component Analysis

Look for a low-dimensional representation of the dataset that contains as much *variation* in the dataset as possible.

E.g. for plotting our data and gaining intuition, if we can obtain a 2D representation of the data, then we can plot the observations in this low-dimensional space.

Note that you want to *center* the data and make the scales of features comparable before performing PCA. E.g. if one feature is in kilometers and another in meters, the one in kilometers may appear to have lower variance when in fact this is due to scaling.

Principal Components

The first principal component of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

that has the largest variance. "Normalized" refers to $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

We refer to $\phi_{11}, \ldots, \phi_{p1}$ as the *loadings* of the first principal component.

The loadings make up the first principal component vector.

$$\phi_1 = (\phi_{11} \ \phi_{21} \ \ldots \ \phi_{p1})^T$$

Principal Components

The first principal component loading vector solves the optimization problem

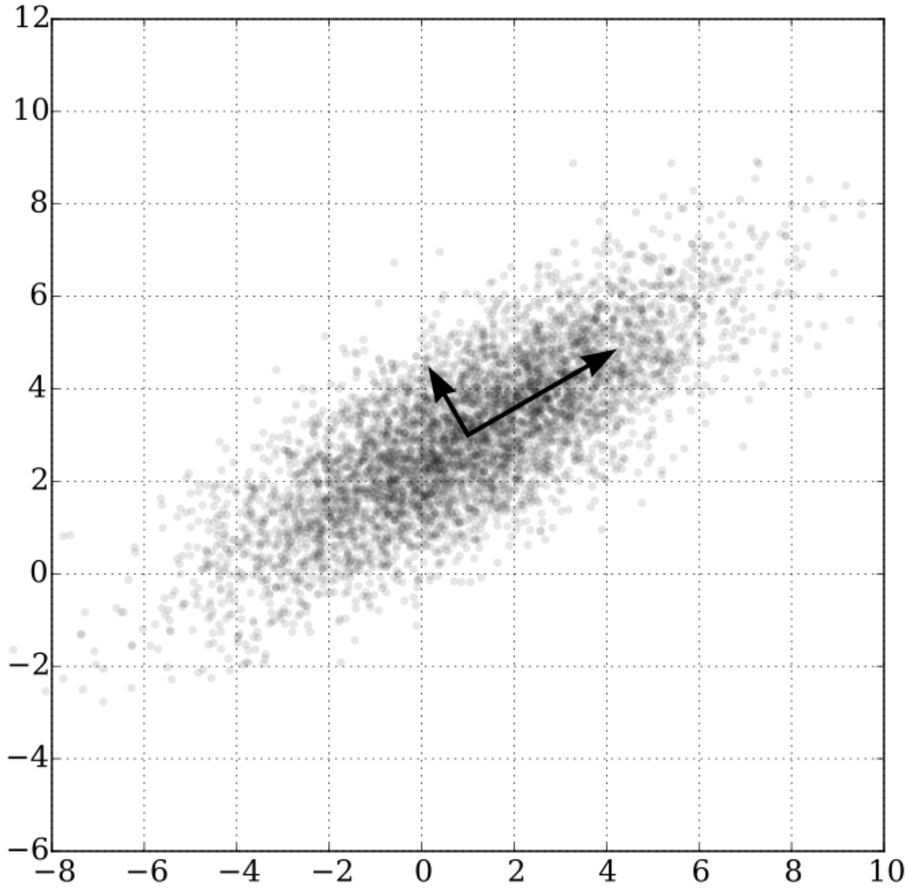
$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Principal Components

The second principal component Z_2 is the linear combination of features that has maximal variance out of all linear combinations that are are uncorrelated with Z_1.

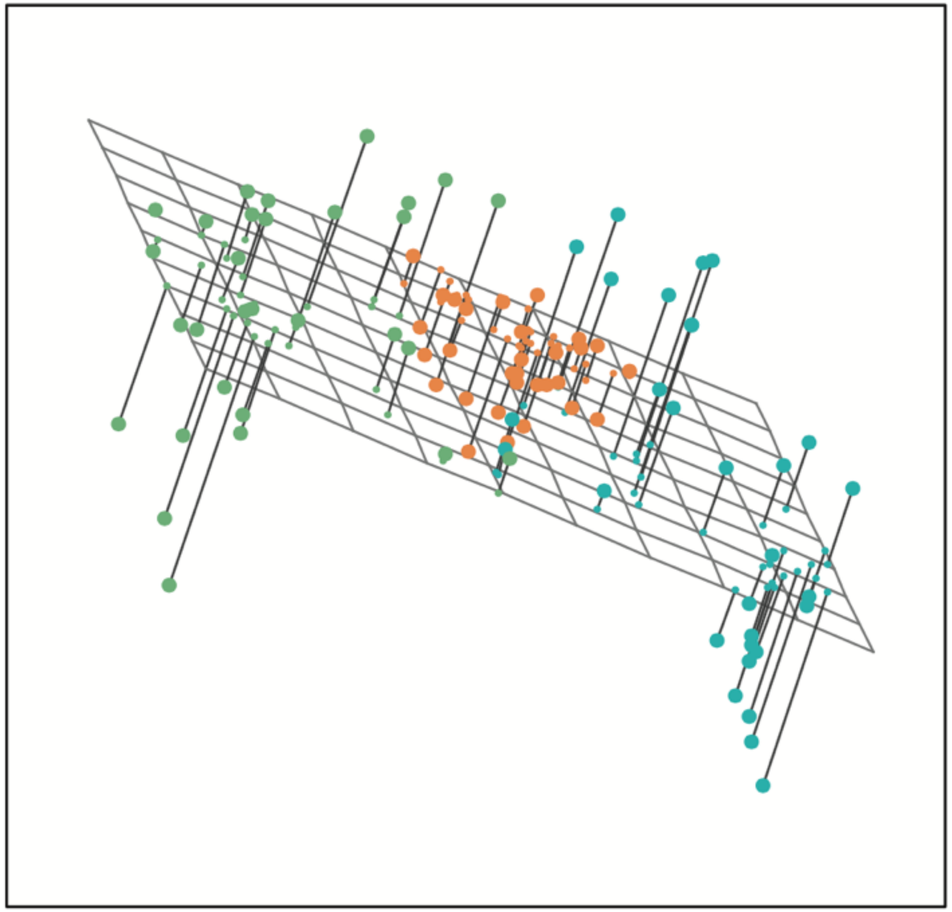
Constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction of ϕ_2 to be orthogonal to ϕ_1 .

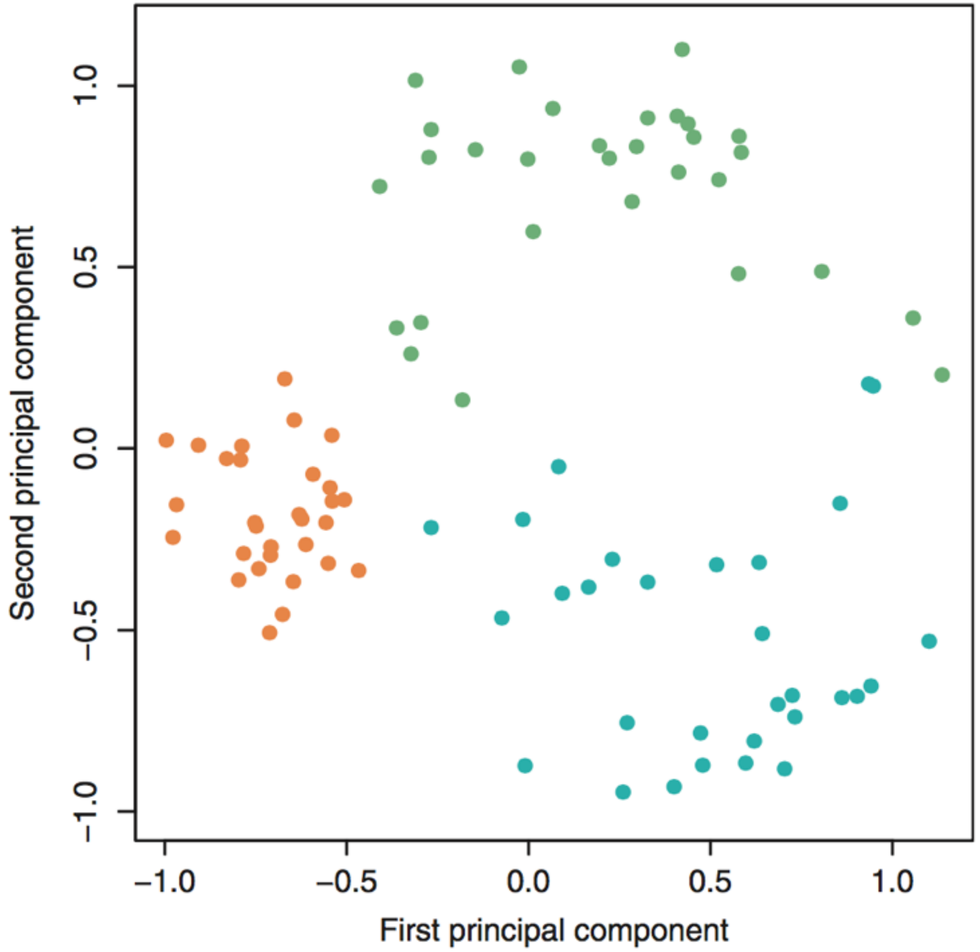
Principal Component Analysis



First two principal axes of this Gaussian dataset.

Principal Component Analysis





Principal Components

Equivalently, find eigenvectors with the largest eigenvalues of the *sample covariance matrix*.

$$X^T X = V D^2 V^T$$

By the singular value decomposition (SVD),

$$X = U D V^T$$

Principal Components

Equivalently, find eigenvectors with the largest eigenvalues of the *sample covariance matrix*.

$$X^T X = V D^2 V^T$$

By the singular value decomposition (SVD),

$$X = U D V^T$$

The right singular vectors are the loadings, or principal axes, of the data.

Principal Components

Equivalently, find eigenvectors with the largest eigenvalues of the *sample covariance matrix*.

$$X^T X = V D^2 V^T$$

By the singular value decomposition (SVD),

$$X = U D V^T$$

UD is the full principal components decomposition of X, aka the Z's on previous slides.

How many principal components?

Scree plot