CQS Summer Institute: Machine Learning in Data Science

Matthew S. Shotwell, Ph.D.

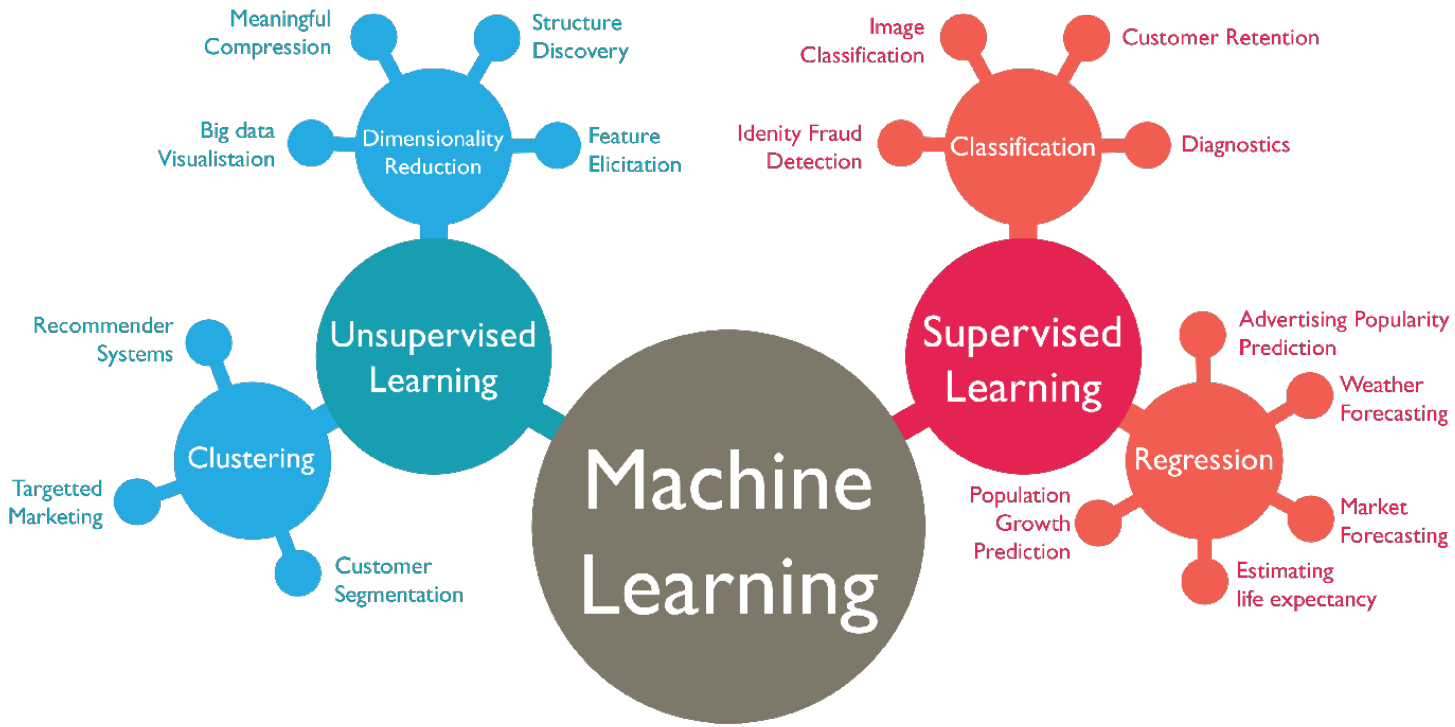
Department of Biostatistics
Vanderbilt University Medical Center
Nashville, TN, USA

August 8, 2019

Course Overview

- Syllabus and R code:
- https://github.com/biostatmatt/cqs-ml-stat-r
- Monday: Intro and Data Management
- Tuesday: Supervised Learning Part 1
- Wednesday: Supervised Learning Part 2
- Thursday: Supervised Learning Part 3
- Friday: Unsupervised Learning

Machine learning



source: https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications

Unsupervised learning

- Learning without a teacher
- Often more challenging than supervised learning
- Often exploratory
- Can't "check your work" as for supervised learning
- Have X but no Y
- Goal to discover structure in distribution of X:
- Dimension reduction: identify low-dimensional regions of the X space with high data density. Example method: principal components analysis
- Cluster analysis: identify convex regions of the X-space that contain high data density. Example method: k-means clustering

Principal components analysis (PCA)

- Useful when large set of correlated variables
- Summarize with fewer variables (*principal components*)
- Small information loss
- PCA refers to process of finding PCs
- Always center data first (mean $= 0$)

What are principal components?

- PCs are linear combinations of the inputs

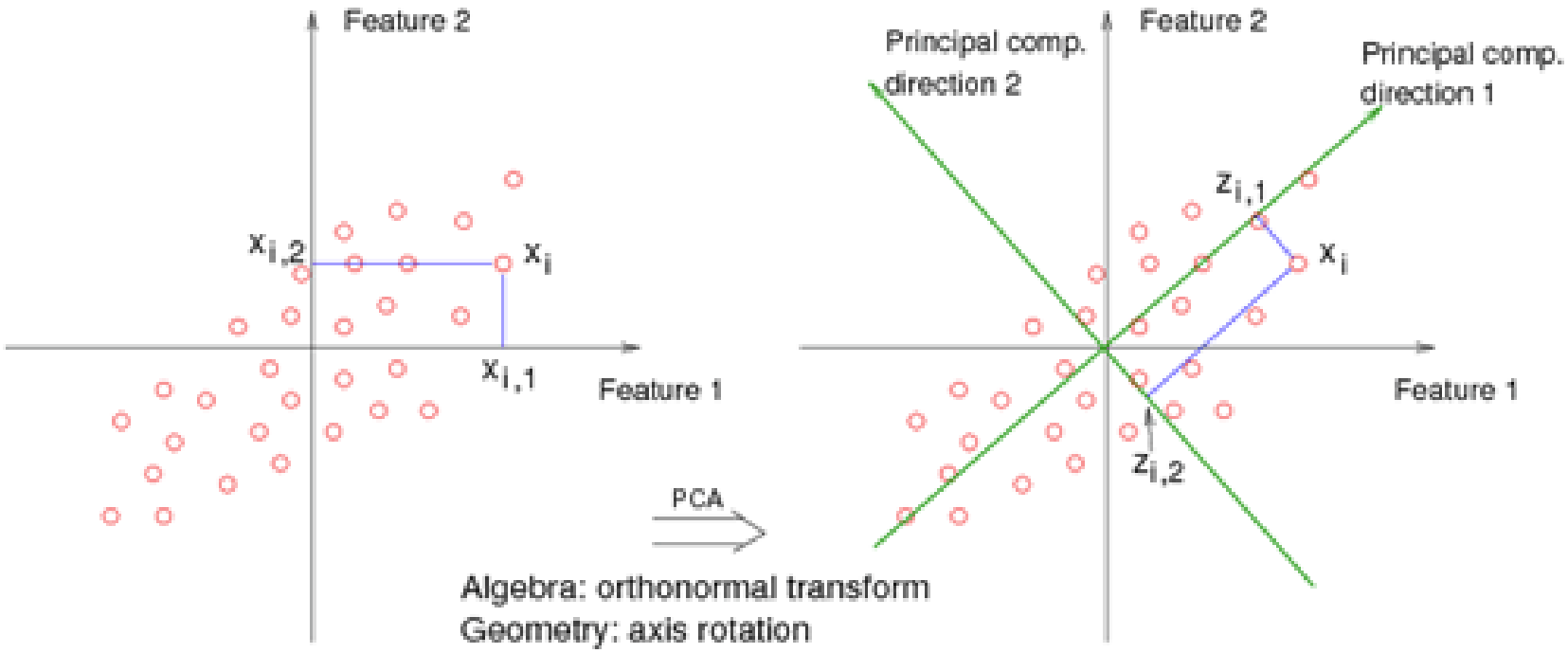
$$Z_1 = \phi_{11} X_1 + \phi_{21} x_2 + \ldots + \phi_{p1} X_p$$

- The first PC (Z_1) is lin. comb. with largest variance
- $\phi_{11}, \ldots, \phi_{p1}$ are the "loadings" of first PC
- Loadings are restricted s.t., $\sum_{j=1}^{p} \phi_{j1}^2 = 1$
- Values of PC in particular data set are called "scores"
- e.g., z_{i1} is score for obs i on PC 1.

What are principal components?

- Second PC Z_2 is lin. comb. of inputs with largest variance and is also uncorrelated with Z_1
- ϕ_2 orthogonal to ϕ_1
- This means PC directions are perpendicular
- Software computes the loading and score vestors (e.g., ϕ_1, z_{i1})

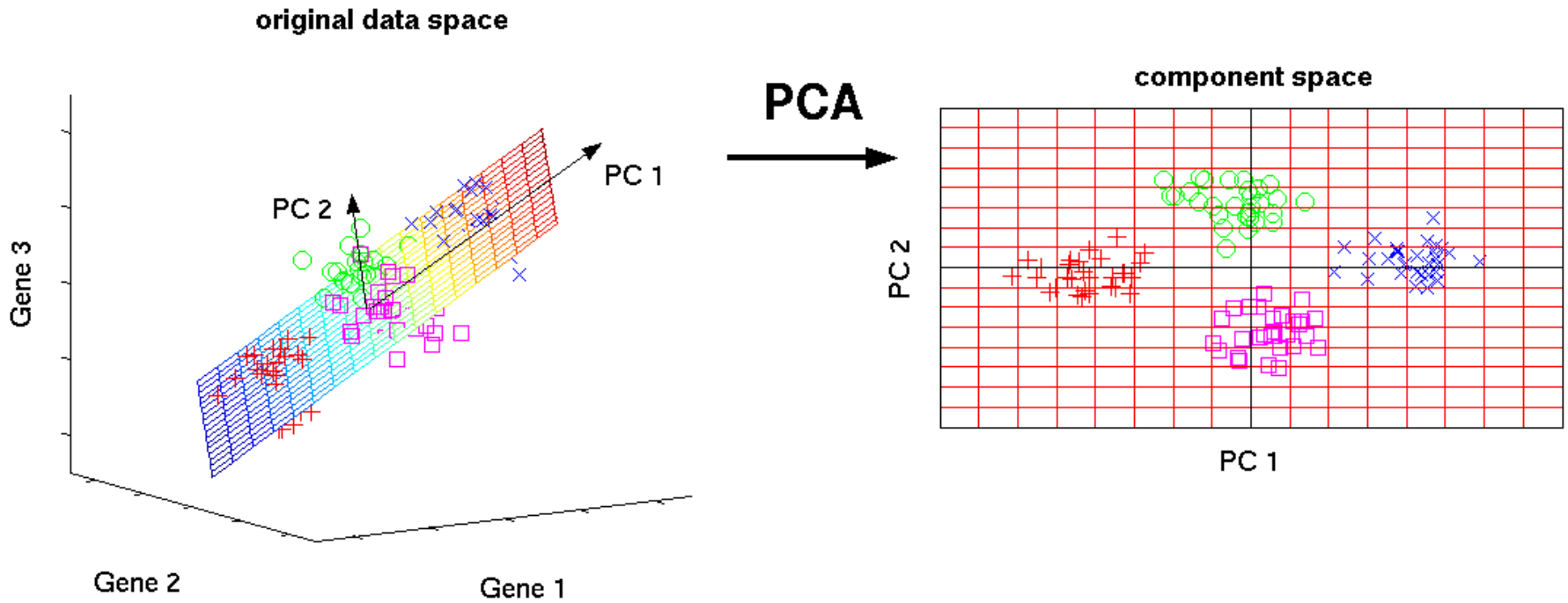
PCA with two inputs



PCA for more than two inputs

- If p inputs, process is repeated to identify p PCs
- Then plot pairs of PC scores for low-dimensional view of data
- E.g., can plot z_{i1} vs z_{i2}, the first two PCs
- This is a "projection" of data onto plane defined by first two PCs

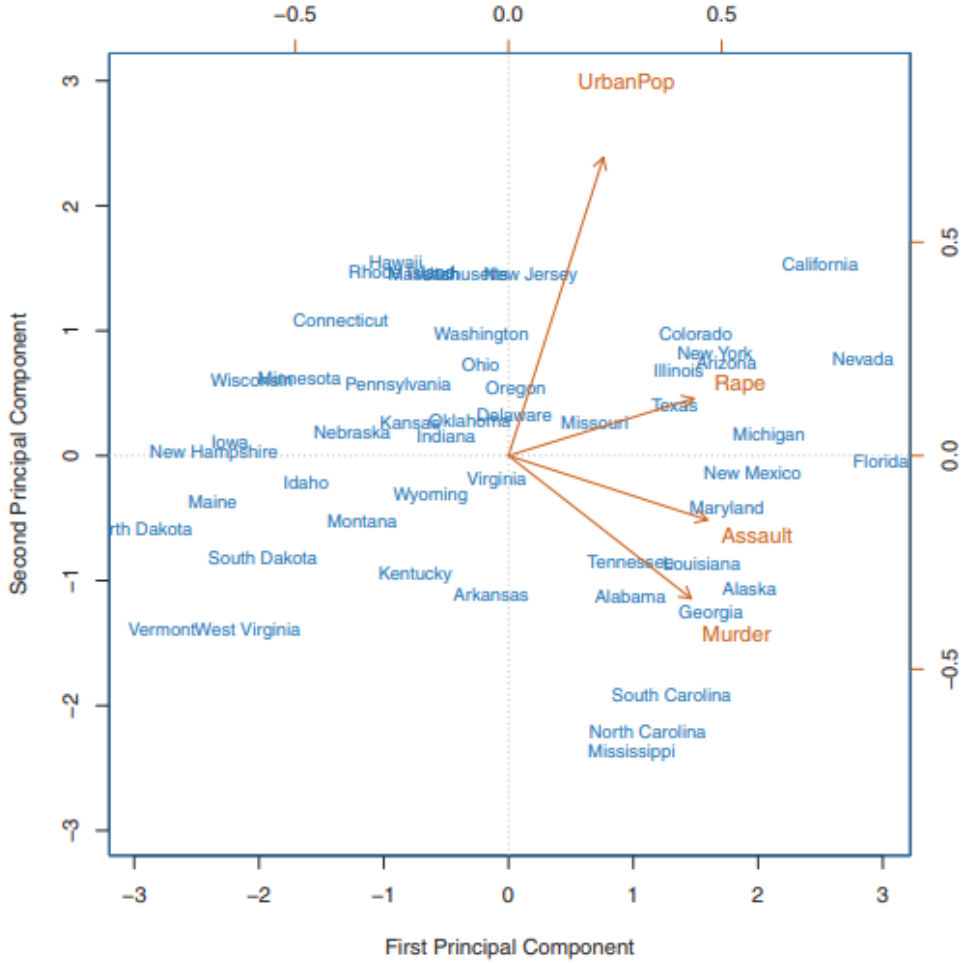
2D projection of 3D data onto first two PCs



Example: USArrests data

- For each state:
- # arrests per 100k for Assault, Murder, and Rape
- % of pop. living in urban area (UrbanPop)
- 50 rows (states), so PC score vectors have length 50
- 4 variables, so 4 PCs and each loading vector has length 4

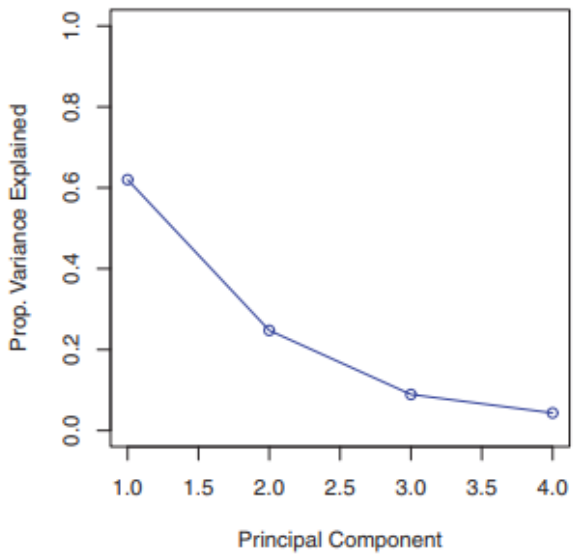
USArrests visualization: *biplot* of PC1 vs PC2

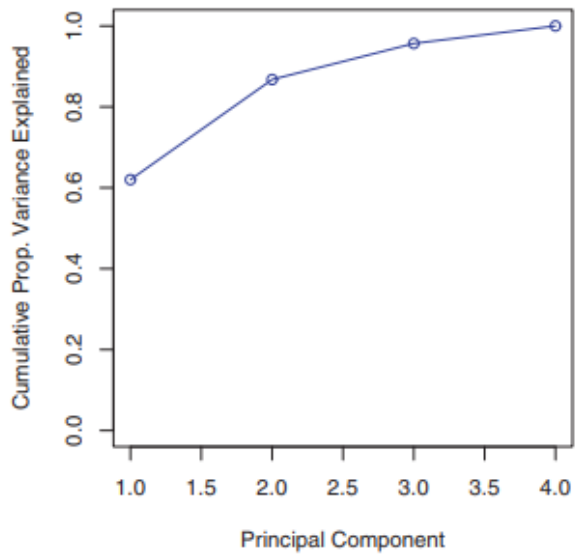


- First PC overall measure of crime
- Second PC measure of urbanization
- CA high crime and urbanization
- GA moderate crime and low urbanization

PCA for dimension reduction

- Because first few PCs contain most of information, can omit high PCs. In other words, first $m < p$ PCs provide best m-dimensional approximation of data. Can quantify loss of information by computing proportion of variance explained by each PC. Plot of these data are called *scree plots*.
- How to decide how many? Elbow rule.





PCA in R: principal-components.R

Clustering

- Objective: find subgroups of similar inputs
- Must define *similar* or *different*
- Similarity is domain specific
- Many different clustering methods
- Best known: k-means and hierarchical clustering

k-means clustering

- Partition data into k non-overlapping clusters
- Must specify k
- k-means algorithm creates k clusters with smallest total within-cluster variance
- Thus Euclidean distance measures similarity
- This difficult combinatorial problem, but iterative algorithm does pretty good

k-means algorithm

Algorithm 10.1 *K-Means Clustering*

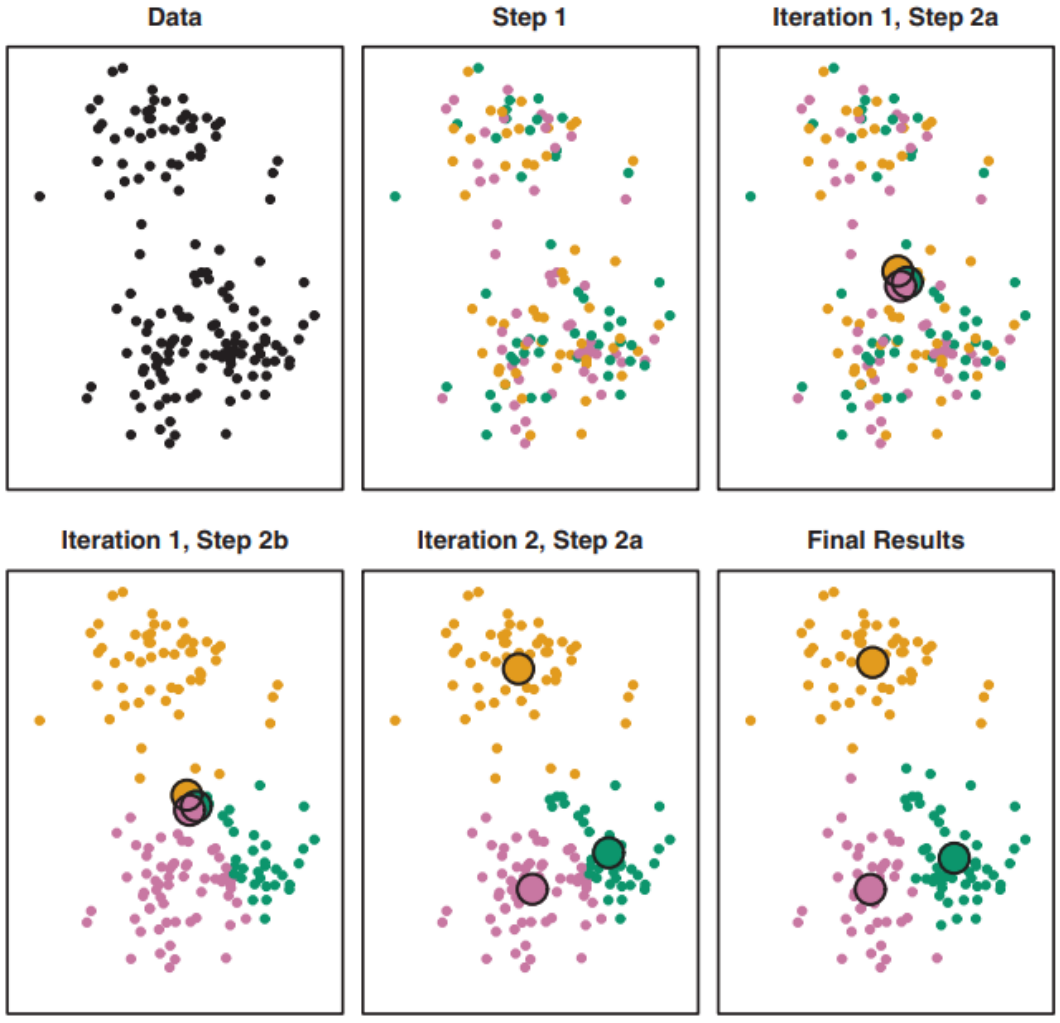
1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

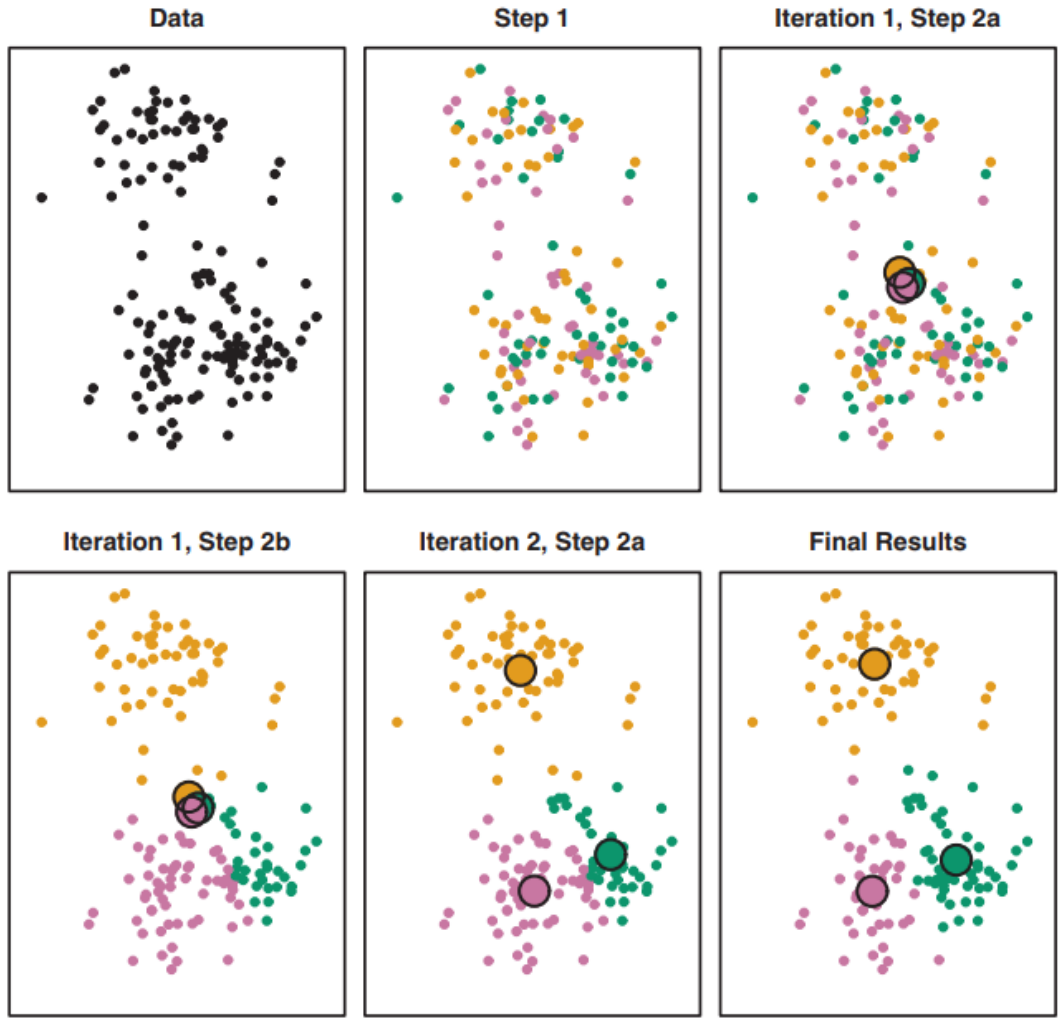
   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

k-means algorithm

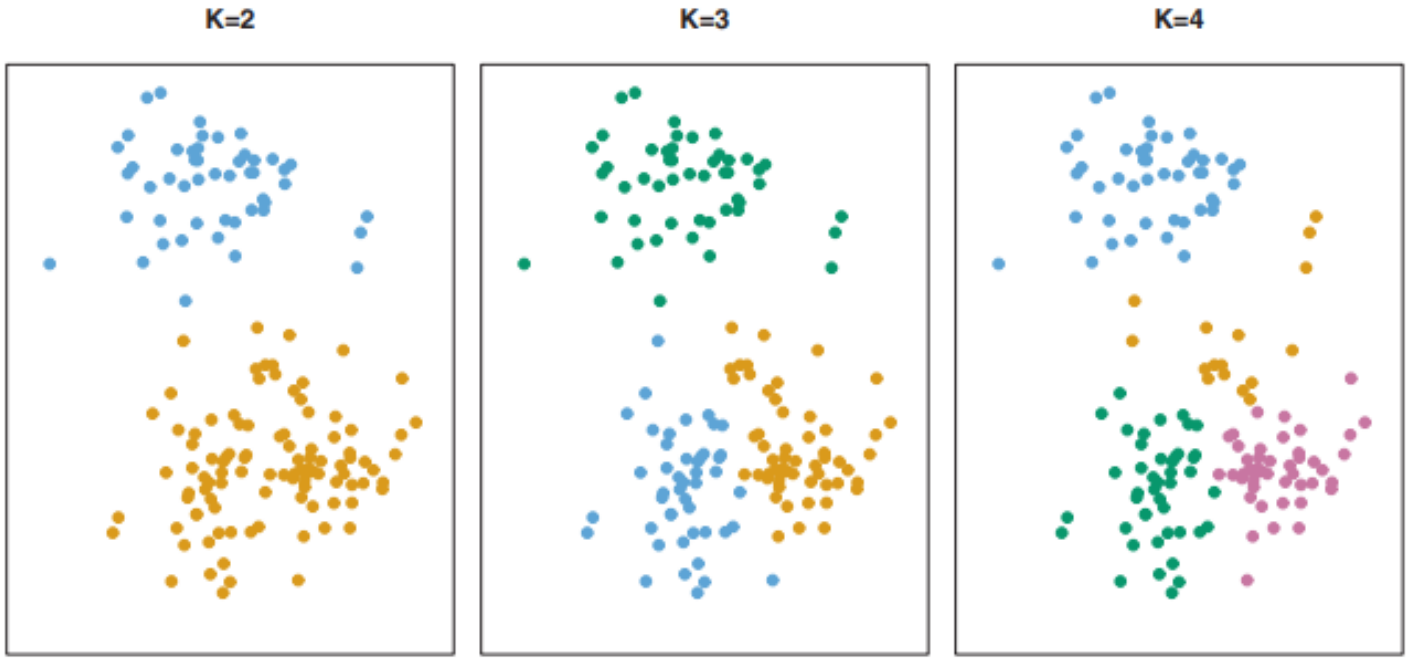


k-means starting values



Need to do k-means several times, pick best

k-means value of k



k-means value of k

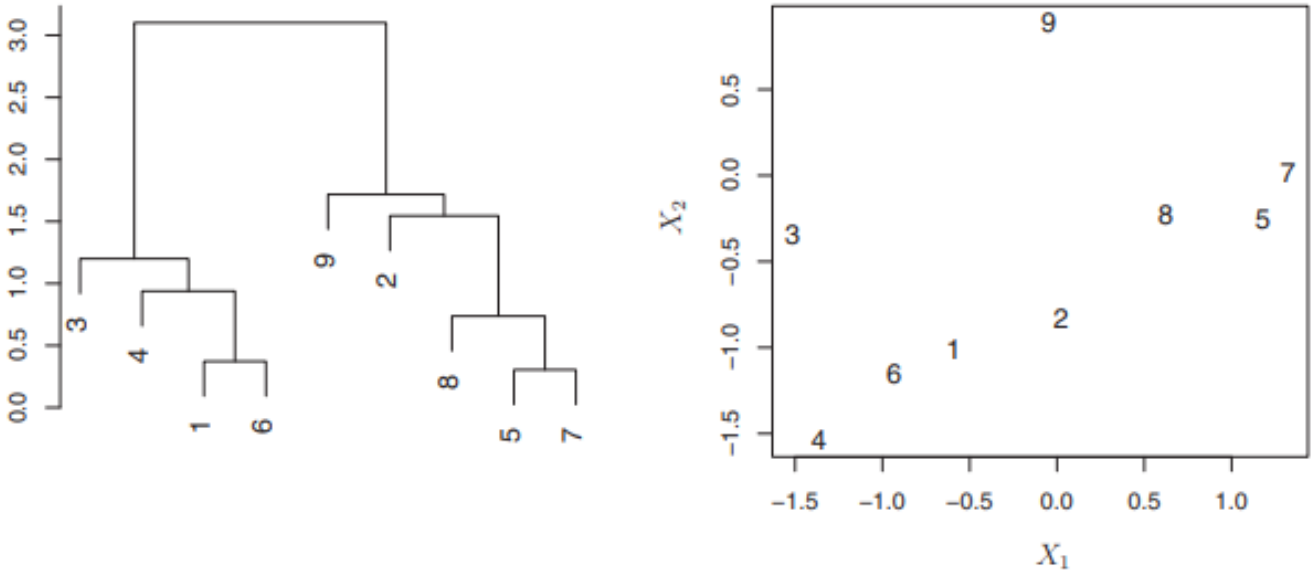
- As k increases total within-cluster variance decreases
- Use domain-specific considerations
- Use the elbow rule if no outside info

k-means clustering in R: clustering.R

Hierarchical clustering

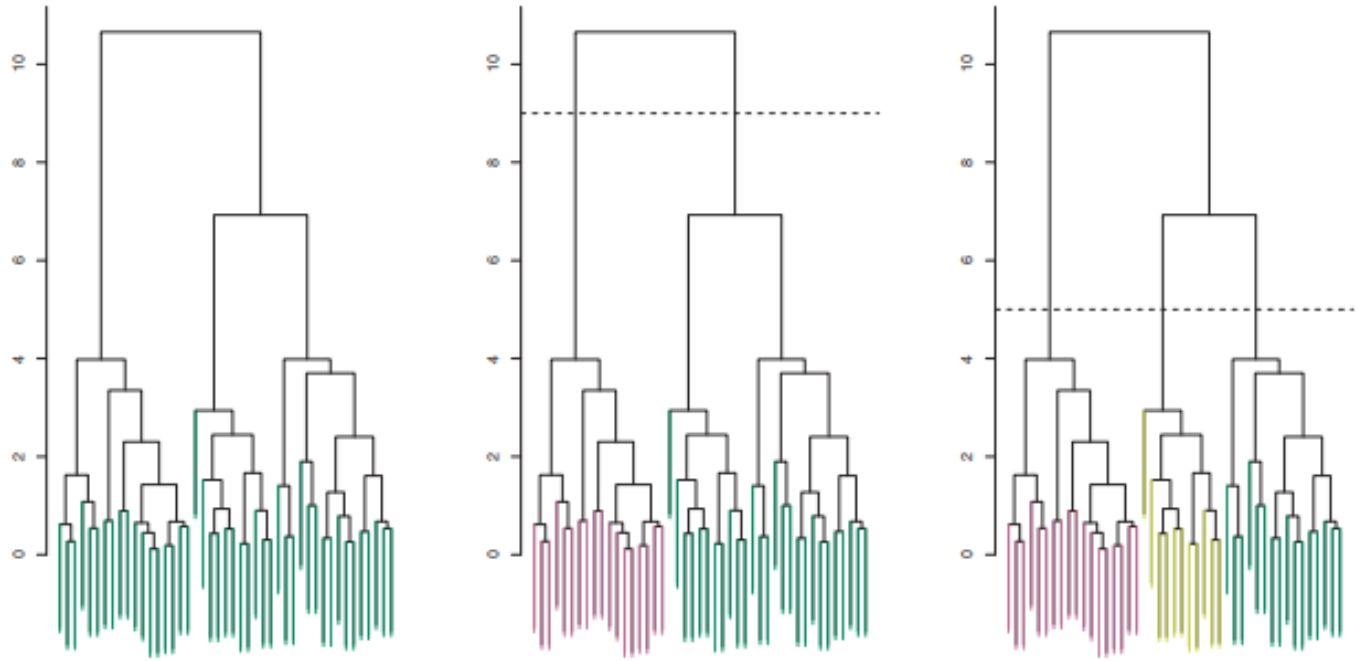
- Generates a sequence of clusters
- Split or merging clusters at each step
- Does not require prespecification of k
- Has tree-based representation: *dendrogram*
- Top-down and bottom-up (agglomerative) versions
- Top-down: start with 1 big cluster, split until N clusters
- Bottom-up: start with N clusters, merge until 1 cluster
- Bottom-up most common

Hierarchical clustering: dendrogram



- *Leaf* at bottom is single observation
- Most similar observations merged to form cluster
- Most similar clusters merged to form new clusters
- Vertical axis is dissimilarity of merged clusters

Hierarchical clustering: dendrogram clusters

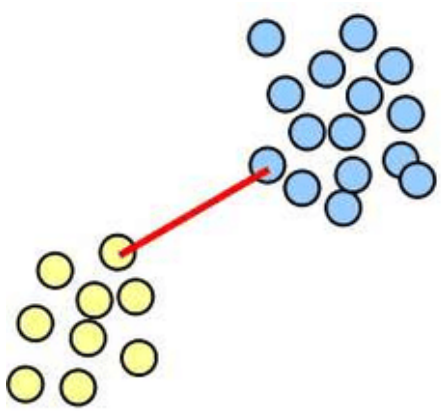


- *Cut* the dendrogram to make clusters
- Where you cut determines k
- Can apply elbow rule to dendrogram

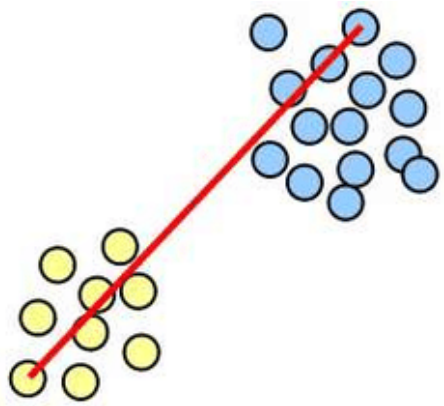
Hierarchical clustering: dissimilarity and linkage

- Must define dissimilarity between observations
- Euclidean distance is common
- Must define dissimilarity or *linkage* between clusters:
- *Complete* - Maximum intercluster dissimilarity
- *Single* - Minimal intercluster dissimilarity
- *Average* - Mean intercluster dissimilarity
- *Centroid* - Dissimilarity between centroids
- Different dissimilarity and linkage results in different dendrograms
- Can try a variety and see if patterns consistently emerge

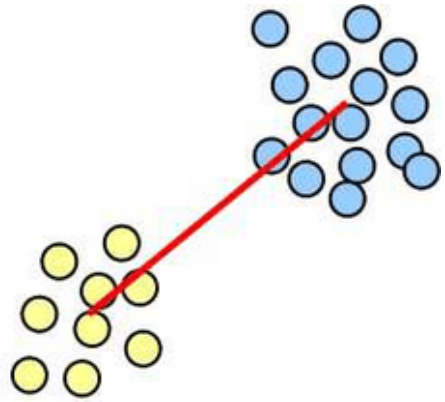
Hierarchical clustering: linkage



single-link

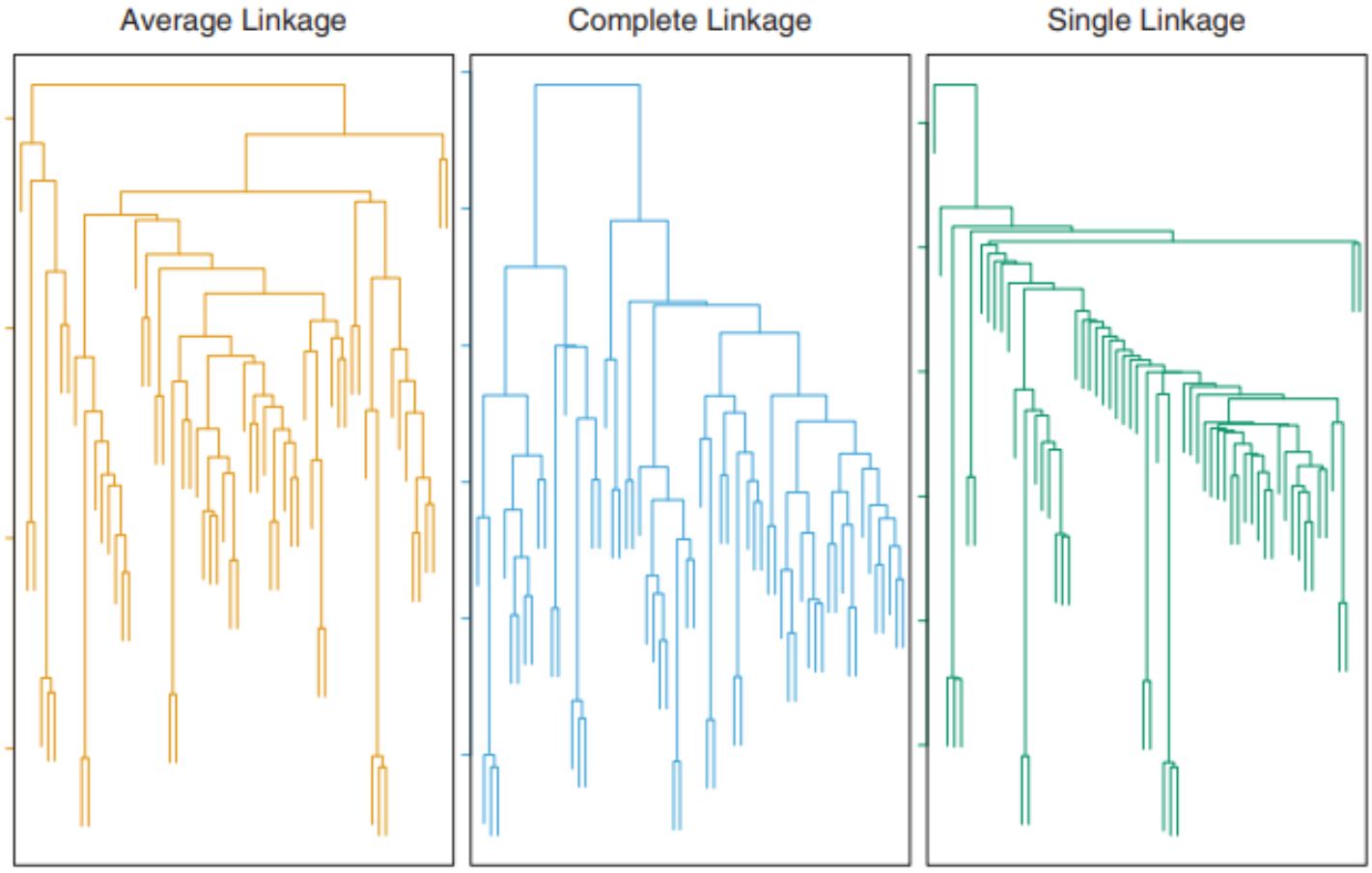


complete-link



average-link

Hierarchical clustering: linkage



Hierarchical clustering: bottom-up algorithm

Algorithm 10.2 *Hierarchical Clustering*

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.

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

Hierarchical clustering in R: clustering.R