Topic # 05: Data Analytics

Clustering Methods

Instructor: Prof. Arnab Bisi, Ph.D.

Johns Hopkins Carey Business School

Non-parametric Methods

Session 5:

Agenda

- Part 1: K-Nearest-Neighbors
- Part 2: K-means clusters
- Part 3: Hierarchical clusters
- Break
- Part 4: Working with code
- Part 5 : Course Projects

Non-Parametric Methods

- Parametric methods estimate the value of specific "parameters"
- Many advantages
  - Easy to fit
  - Estimate a small number of values
  - Simple interpretation
- Some disadvantages
  - Strong assumptions are made about the world
  - True relationship may be far from linear or logistic
  - Poor data fit, wrong conclusion
- Non-parametric methods
  - Do not explicitly assume a parametric model
  - Provide more flexible approaches

K-Nearest Neighbors (KNN)

- Given a positive integer K and a test observation x_0
- KNN first identifies the K points in the training data that are closest to x_0
  - Call this set, Set_0
  - Estimate the conditional probability by

$$\Pr\left(Y = j \mid X = x_0\right) = \frac{1}{K} \sum_{i \in Set_0} I\left(y_i = j\right),$$

  - Where $I()$ is an indicator function
  - The estimate of $f(x_0)$ is

$$\hat{f}\left(x_0\right) = \frac{1}{K} \sum_{x_i \in Set_0} y_i$$

  - KNN applies Bayes rule and classifies x_0 in the class with the largest probability

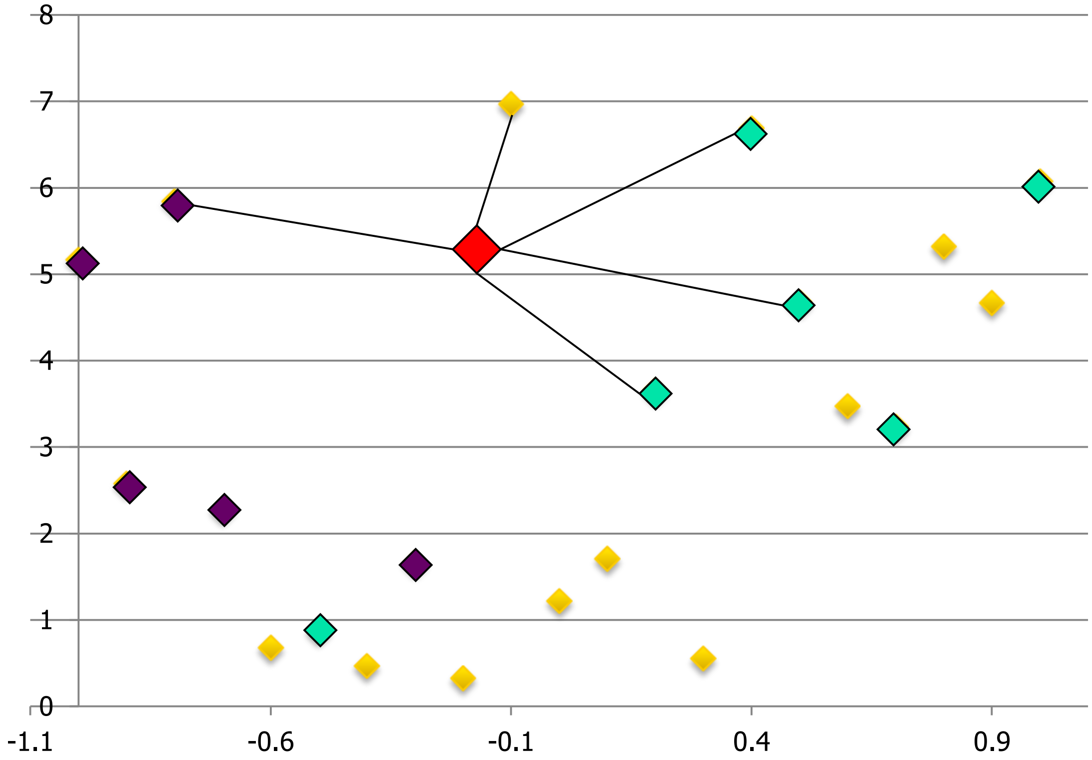
JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Steps in KNN

1. Identify class for each point in a set
2. Find distance between these points and new point x_0

$$D_{1,2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

3. Identify KNN
4. Count number of neighbors in each class
5. Assign new point to class containing highest number of neighbors



KNN Example

- We use the knn() function which is part of the class library
  - First we fit a model using the training data
    - This identifies the class of each of your neighbors
  - Then we use the model to make predictions
    - Assign new points to a class
- KNN requires four inputs
  - Data Frame containing the predictors associated with the training data
    - Coordinates of points already assigned to a class
  - A Data Frame containing the predictors associated with the test data for which we wish to make predictions
    - Coordinates of points yet to be assigned to a class
  - A vector (list) containing the class labels for the training observations
    - Which class each assigned point is already in
  - A value for K, which is the number of nearest neighbors to be used by the classifier

KNN Example: Smarket Data

- Sample code
  - > require(ISLR)
  - > Smarket = Smarket
  - > require(class)
  - > knn1.fit=knn(train.data[,c(2:7)], test.data[,c(2:7)], train$Direction, 1)
  - > head(knn1.fit)
  - > mean(knn1.fit == test.data$Direction)

- When k = 1 the prediction is correct 50% of the time

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Exercise

- Use Smarket data set
- Perform KNN on the training data with several values of K (1, 3, 5, 7, 9, etc)
  - Plot the error rate as a function of K
  - Which value of K works best?

- 15 mins

Supervised Learning

- Supervised learning: We're predicting an output variable (or a class for that variable) for which we get to see examples.

- **Data:** n observations including response Y and p features $X_1, X_2, \ldots, X_p$.

- **Goal:** Predict Y using $X_1, X_2, \ldots, X_p$.
  - Regression
  - Classification

Johns Hopkins Carey Business School

Unsupervised Learning

- Unsupervised learning: We're searching for insights about a target for which we do not get to see examples.
- **Data:** n observations only including p features $X_1, X_2, \ldots, X_p$.
- **Goal:** not interested in prediction;
- Discover interesting things;
- Discover subgroups.

Unsupervised Learning

- Unsupervised learning: is often more subjective because there is not a simple goal
- Techniques growing in importance in many fields
  - Subgroups of cancer patients grouped by gene expressions
    - Often easier to obtain unlabeled data from a lab instrument or computer
  - Shoppers grouped based on browsing and purchase history
  - Movies grouped by ratings or reviews

Clustering Methods

- “Show the subgroups in the data”: close to each other.
- Find homogeneity and heterogeneity among the data.
- K-means clustering: partition observations into a pre-specified number of clusters
- Hierarchical clustering: not know number of clusters; end up with a tree-like visual representation
- We always want the same things: low deviance, without overfit.

How do we define close?

- Most important step
  - Garbage in $\Rightarrow$ garbage out
- Distance or similarity
  - Continuous - euclidean distance
  - Continuous - correlation similarity
  - Binary - Manhattan distance
- Pick a distance/similarity that makes sense for your problem
- How do we visualize the grouping?
- How do we interpret the grouping?

Johns Hopkins Carey Business School

Applications of Unsupervised Learning

- Marketing
  - Online shopping sites identify similar shoppers
  - Market segmentation
- Cancer researchers look for subgroups to obtain better understanding of diseases
- Search engines display same search results to users with similar search patterns

Overview of K-Means Clustering

- K-means clustering is a simple and elegant approach for partitioning a data set into K distinct, non-overlapping clusters
- To perform K-means clustering, we must first specify the desired number of clusters K
- Then, the K-means algorithm assigns each observation to exactly one of the K clusters.

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Overview of K-Means Clustering

- A partitioning approach
  - Fix a number of clusters
  - Get "centroid" of each cluster
  - Assign things to closest centroid
  - Recalculate centroids

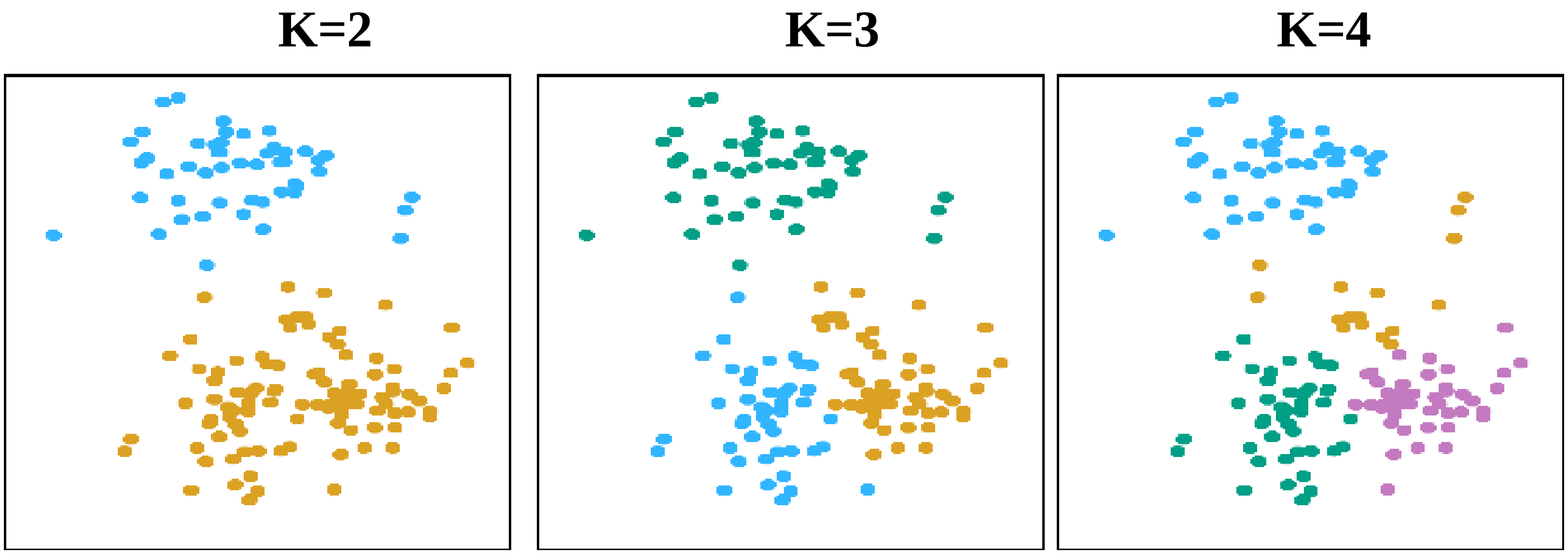
- Requires
  - A defined distance metric
  - A number of clusters
  - An initial guess as to cluster centroids

- Produces
  - Final estimate of cluster centroids
  - An assignment of each point to clusters

Johns Hopkins Carey Business School

Clustering Illustration

Simulated data in two-dimensional space: clustered into K classes by the *K*-means clustering algorithm.



Johns Hopkins Carey Business School

Best K Clusters

Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. The sets satisfy two properties:

- $C_1 \bigcup C_2 \bigcup \ldots \bigcup C_K = \{1, 2, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.
- $C_k \bigcap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are nonoverlapping: no observation belongs to more than one cluster.

The idea behind K-means clustering is that a good clustering is one for which the within-cluster variation is as **small** as possible.

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Within-Cluster Variation

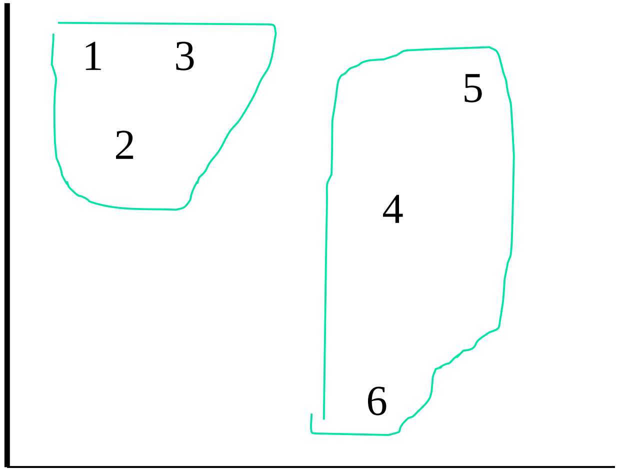
- The within-cluster variation for cluster C_k is a measure $W(C_k)$ of the amount by which the observations within a cluster differ from each other.
- Using squared Euclidean distance, we define

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

- $|C_k|$ denotes the number of observations in cluster k
- The best K-means clustering,

$$\min_{C_1, C_2, \ldots, C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Within-Cluster Variation



- Objective: minimize Within-cluster variation
  = total within sum of squares (tot.withinss)
- tot.withinss = $(D_{1,2}^2 + D_{1,3}^2 + D_{2,3}^2)/3 + (D_{4,5}^2 + D_{4,6}^2 + D_{5,6}^2)/3$
- betweenss = $\{(D_{1,4}^2 + D_{1,5}^2 + D_{1,6}^2) + (D_{2,4}^2 + D_{2,5}^2 + D_{2,6}^2) + (D_{3,4}^2 + D_{3,5}^2 + D_{3,6}^2)\}/9$
- totss = tot.withinss + betweenss

K-means Algorithm in R

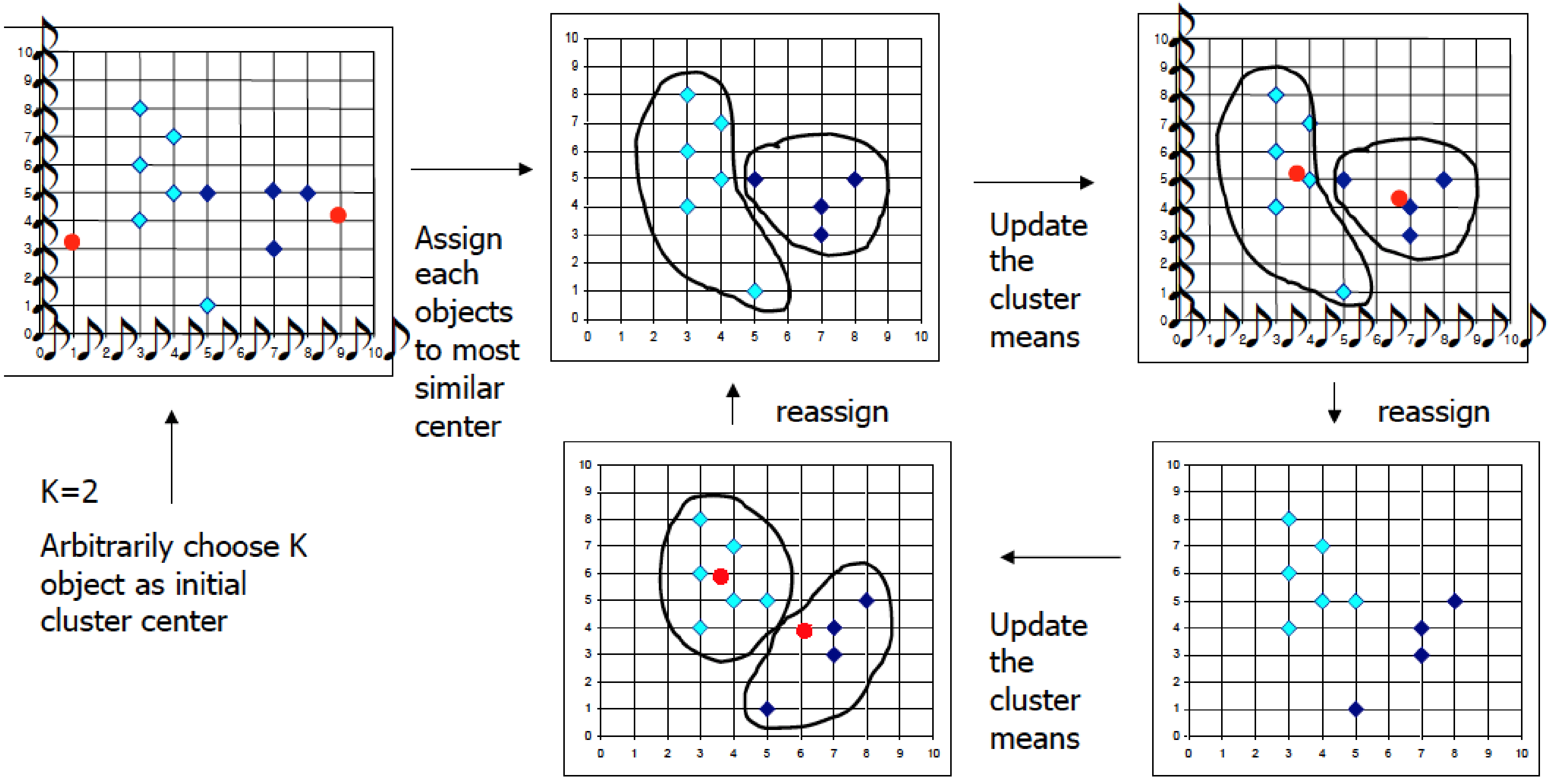
- The K-means method is as follows:
  - Randomly select K rows (points) in the data set. These are treated as initial centroids
  - Assign each point to the closest centroid, where closest is defined using Euclidean distance.
  - Calculate new centroids after all points are assigned
    - The *k*th cluster centroid is the vector of the *p* feature means for the observations in the *k*th cluster.
  - Iterate until the cluster assignments stop changing
- It does not necessarily obtain the global optimum. It is important to run the algorithm multiple times from different random initial configurations.

- Algorithm AS 136: A K-Means Clustering Algorithm. Hartigan JA, Wong MA. Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol 28, No 1, pp 100-108.

K-means Algorithm in R

The *K-Means* Clustering Method

- Example



JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Example in R: Simulated Data

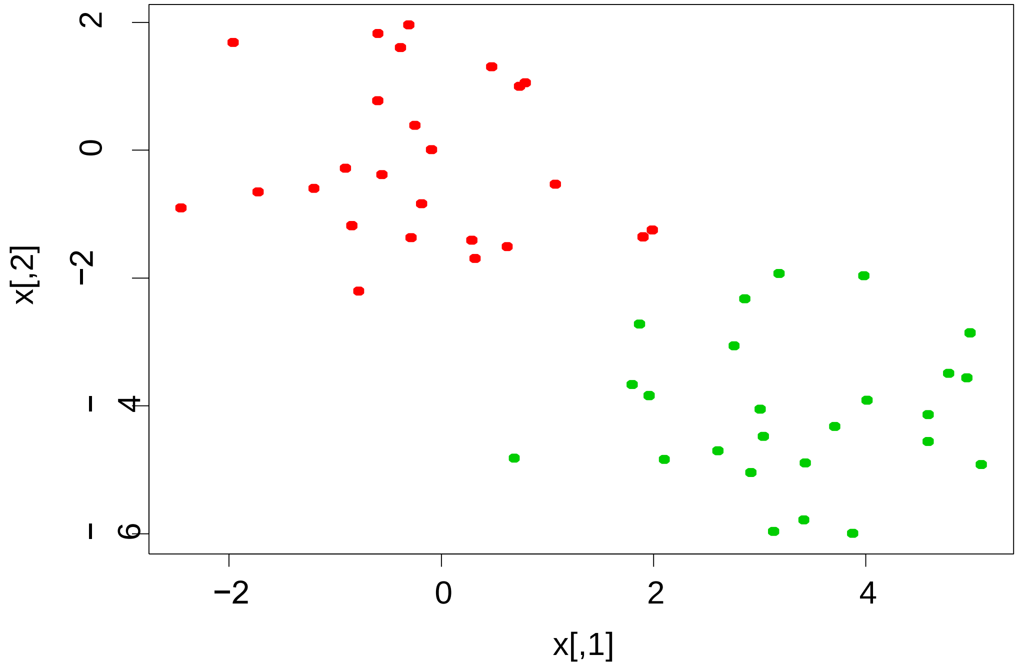
- Function kmeans() performs K-means clustering in R. Example with simulated data
- `> set.seed(2)`
- `> x=matrix(rnorm(50*2),ncol=2)`
- `> plot(x)`
- `> x[1:25,1]=x[1:25,1]+3`
- `> x[1:25,2]=x[1:25,2]-4`
- `> plot(x)`
- `> km.out=kmeans(x,2,nstart=20)`
- `> names(km.out)`
- `> km.out$cluster`

- `> plot(x,col=km.out$cluster+1,pch=20,lwd=3)`
- `> km.out$tot.withinss`

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

K-Means Clustering

Simulated data in two-dimensional space: clustered into K=2 classes by the *K*-means clustering algorithm.



Example in R

- Change number of clusters (centers)
  - > set.seed(4)
  - > km.out3 = kmeans(x, centers = 3, nstart = 20)
  - > km.out3
  - > plot(x, col = km.out3$cluster + 1, pch = 10, lwd = 5)
  - >km.out3$tot.withinss

- Recommend running K-means clustering with a large value of nstart, such as 20 or 50

Summary: Kmeans Clustering

- K-means requires a number of clusters
  - Pick by eye/intuition
  - Pick by cross validation/information theory, etc. Determining the number of clusters

- K-means is not deterministic
- Different # of clusters
- Different number of iterations

Exercise

- Use wine.csv data set
- Cluster the data set into 2 groups
  - Hint: only use the numerical data
  - How do the groups compare to the colors?

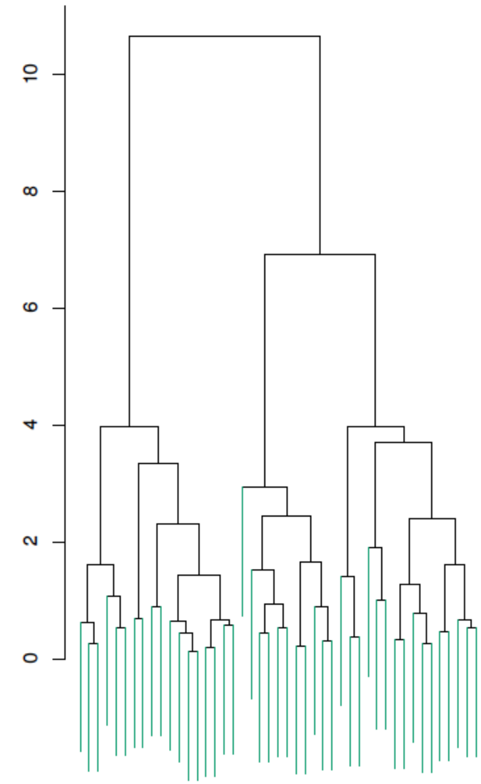
- 15 mins

Hierarchical Clustering

- Disadvantage of K-means clustering: it requires that we pre-specify the number of clusters K
  - Using unsupervised approaches it is hard to know what the proper number K is
- Hierarchical Clustering is an alternative: does not require K
- Advantage of Hierarchical Clustering: it results in an attractive tree-based representation of the observations, called a dendrogram

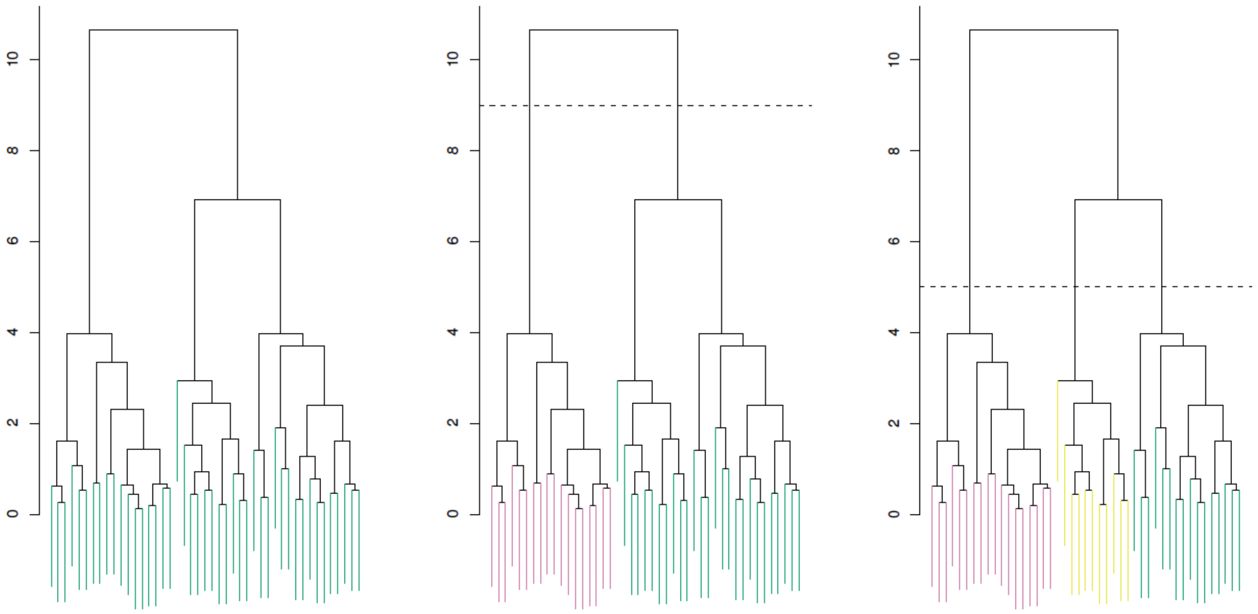
Hierarchical Clustering

- An agglomerative approach
  - Find two closest points
  - Put them together
    - Treat this as one point
  - Repeat until back to a single set



- Requires
  - A definition of distance
  - A merging approach
- Produces
  - A tree showing how close things are to each other

Hierarchical Clustering: Example



- The leaves at the bottom of the dendrogram represent the individual units
- Leaves are combined to form small branches
  - Vertical length of a branch is the increase in tot.withinss
- Small branches are combined into larger branches, until one reaches the trunk or the root
- Where to cut: use your eyes

Hierarchical Clustering: Algorithm

Algorithm

(1) Begin with n observations and a measure (such as Euclidean distance) of all the 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.

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

Hierarchical Clustering: **R**

Function hclust() implements Hierarchical Clustering in **R** Example:

use Euclidean distance to measure dissimilarity:
dist()

```
>  set.seed(2)
>  x=matrix(rnorm(50*2),ncol=2)
>  x[1:25,1]=x[1:25,1]+3
>  x[1:25,2]=x[1:25,2]-4
> hc.complete=hclust(dist(x),method="complete")
> hc.average=hclust(dist(x),method="average")
> hc.single=hclust(dist(x),method="single")
> hc.centroid=hclust(dist(x),method="centroid")

> par(mfrow=c(1,4))
> plot(hc.complete)
```

Use function cutree() to determine the cluster labels

```
> cutree(hc.complete,3)
```

Clustering: scaling

- In addition to carefully selecting the dissimilarity measure used, one must consider whether or not the variables should be scaled
- Example 1: a shopper buys 10 pairs of socks per year and one computer
  - High-frequency purchases tend to have a much larger effect
  - After scaling, each variable will be given equal weight
  - **R** code: function scale()
- Example 2: data shows age in years and income in dollars
  - Since income is on a much broader scale, it dominates the distance calculations
  - In many settings age is more important

Hierarchical Clustering

- Consider the USArrests data, which is part of the base **R** package. We will now perform hierarchical clustering on the states.
- Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.
- Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

Hierarchical Clustering

- Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one.
- (Hint: use function `scale()` to standardize the data)
- What effect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed? Provide a justification for your answer.

Hierarchical Clustering: R

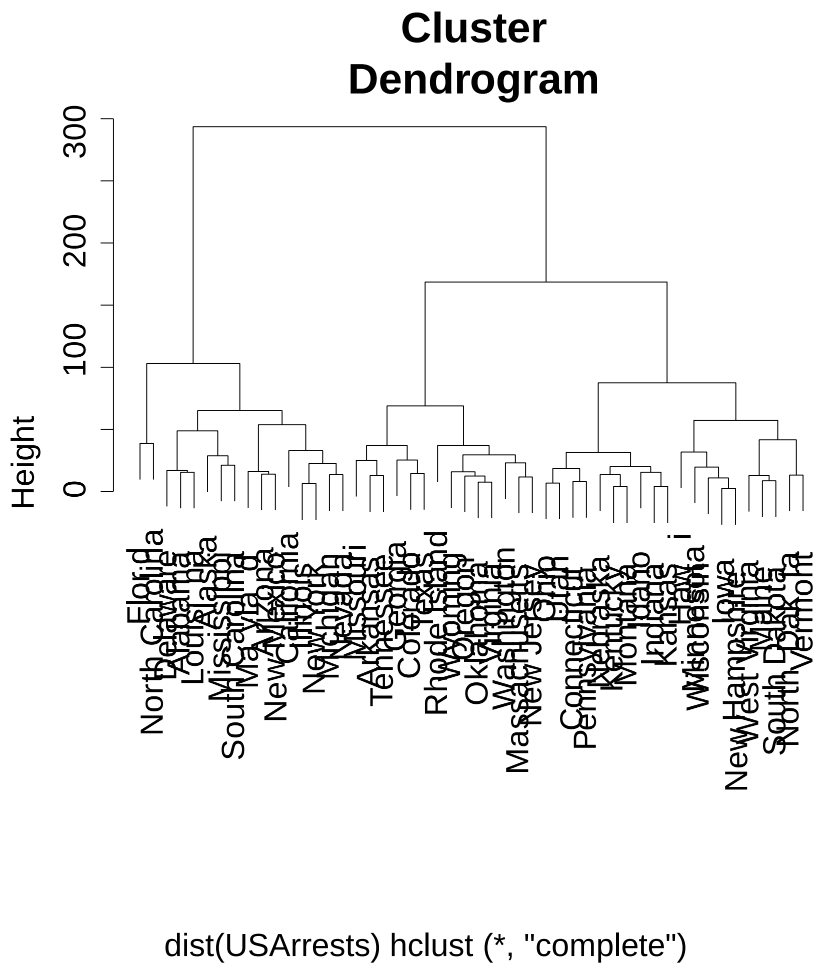
Sample code:

```
states=rownames(USArrests)
arrest.reasons=colnames(USArrests)
hc.complete=hclust(dist(USArrests),method="complete")
plot(hc.complete)
cutree(hc.complete,3)
```

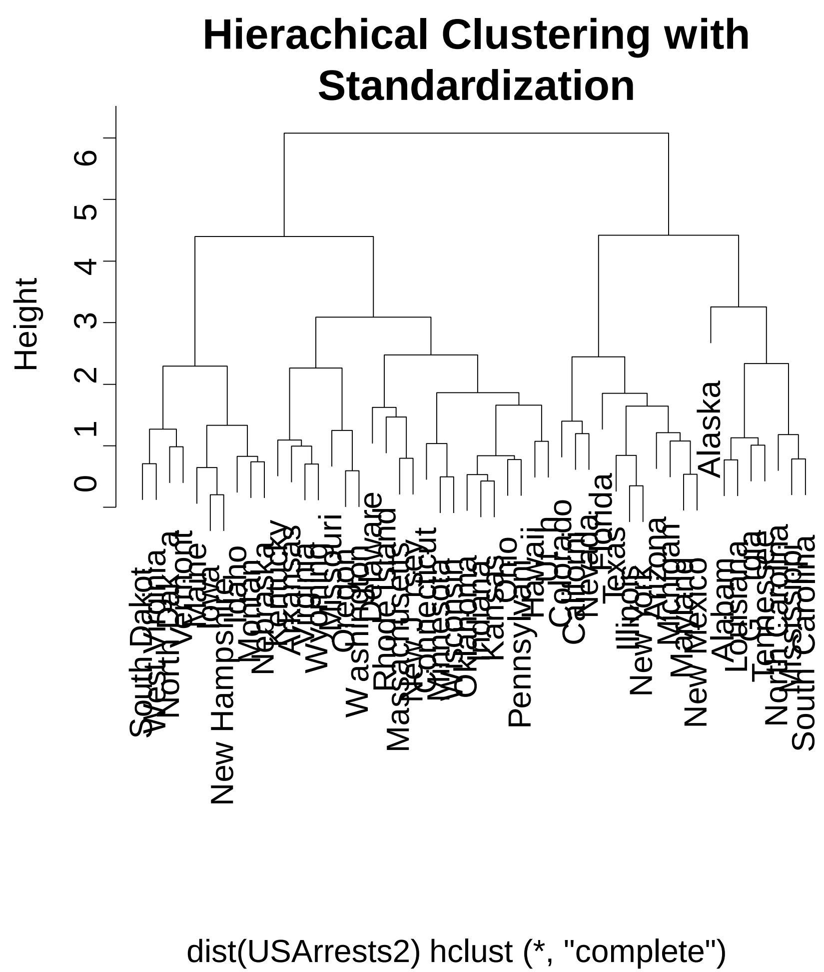
Sample code:

```
USArrests2=scale(USArrests)
hc.complete.2=hclust(dist(USArrests2),method="complete")
plot(hc.complete.2  ,main="Hierachical  Clustering
with  Standardization")
```

Hierarchical Clustering: R



Hierarchical Clustering: standardization



Course Projects

- Instructions
- Data Sets
- Examples

Course Project

- Groups 4-6 Students
- Analyze data
  - Collection
  - Hypothesis
  - Analysis
  - Findings

- Acquire data set
  - Work
  - Prior course
  - Research
  - Public sites
  - UCI Data Repository
  - World Bank
  - US Government Open Data

JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Project Deliverables

- Written Report
- No more than 6 pages
  - Double spaced
  - Appendices do not count
  - Assume reader does not know R
- Data Set
  - csv file
- R-Code
- Presentation File
  - 3-4 speakers/group
  - Can use slides (ppt)

Project Deliverables

- Presentation
  - 12 minutes per group
  - Focus on questions, analysis, and results
    - Not lecture on code
  - 3 or 4 speakers
- Milestones
  - Identify data set by Module 6
  - Submission in Module 7
- Sample projects posted

Questions, Comments?

Let's move to the Code.