MTH 3270 Notes 9

7.1 Unsupervised Learning (9)

- Recall that ***unsupervised learning*** is used for identifying groupings and other patterns from observations of **explanatory variables** (X's) when there's ***no* response variable** (Y).

Data Set: USArrests

The USArrests data set (built into R) contains contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas. The four variables are:

Murder	Murder arrests (per 100,000).
Assault	Assault arrests (per 100,000).
UrbanPop	Percent urban population.
Rape	Rape arrests (per 100,000).

```
head(USArrests)

##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

7.1.1 Hierarchical Clustering

- Fig. 1 below shows the result of **hierarchical clustering** of the 50 states using the USArrests data set.

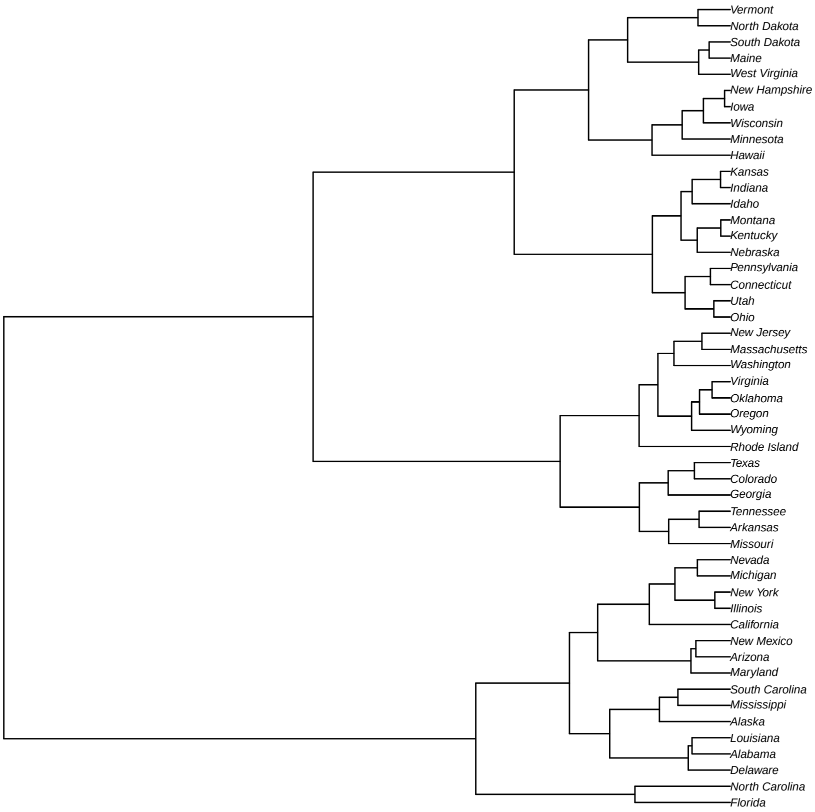
Vermont
North Dakota
South Dakota
Maine
West Virginia
New Hampshire
Iowa
Wisconsin
Minnesota
Hawaii
Kansas
Indiana
Idaho
Montana
Kentucky
Nebraska
Pennsylvania
Connecticut
Utah
Ohio
New Jersey
Massachusetts
Washington
Virginia
Oklahoma
Oregon
Wyoming
Rhode Island
Texas
Colorado
Georgia
Tennessee
Arkansas
Missouri
Nevada
Michigan
New York
Illinois
California
New Mexico
Arizona
Maryland
South Carolina
Mississippi
Alaska
Louisiana
Alabama
Delaware
North Carolina
Florida

Figure 1

- To carry out ***hierarchical clustering***:
  1. Let n denote the number of **rows**, i.e. **observations**, in the data frame, .
  2. Begin with each observation representing a ”singleton” **cluster** (i.e. begin with $\boldsymbol{n}$ **clusters**, each consisting of a single observation).
  3. Merge the two clusters (observations) that are ”closest” (**least dissimilar**) into a single cluster, resulting in $n - 1$ clusters (one of which now has two observations). A measure of **dissimilarity** between clusters is defined below.
  4. At each of the remaining steps, merge the two ”closest” (**least dissimilar**) clusters into a single cluster, producing one less cluster at the next higher level of the tree.
  5. The last ($(n - 1)$st) step produces **one cluster** consisting of **all $\boldsymbol{n}$ observations** in the data frame.

- **Dissimilarity**: If G and H are two **clusters**, and $d_{i,j}$ is the Euclidean distance between observation i in cluster G and observation j in cluster H, three methods of measuring of **dissimilarity** between G and H are:
  1. ***Single linkage*** (or ***nearest-neighbor***): The **dissimilarity** between G and H is the distance between their two **closest** points, i.e.

     $$\text{Dissimilarity}(G,\, H) \;=\; \min(d_{i,j}).$$

  2. ***Complete linkage*** (or ***furthest-neighbor***): The **dissimilarity** between G and H is the distance between their two **farthest** points, i.e.

     $$\text{Dissimilarity}(G,\, H) \;=\; \max(d_{i,j}).$$

  3. ***Group average***: The **dissimilarity** between G and H is the **average** distance between their points, i.e.

     $$\text{Dissimilarity}(G,\, H) \;=\; \text{avg}(d_{i,j}).$$

- **Comment**: Each **distance** ($d_{i,j}$ above) is a Euclidean distance in p-dimensional space, where **each coordinate axis** represents an explanatory (X) variable (column of the data frame). But the variables might be measured on very **different scales**.

  Consider **re-scaling** the variables so that distances along each coordinate axis are comparable and reasonably reflect how different the two observations are. ***Standardizing*** each variable is one possible option.

- These functions, from the `"ape"` package, can be used to carry out **hierarchical clustering**.

```
hclust()        # Carry out hierarchical clustering. Returns an object
                # of class "hclust".
as.phylo()      # Converts an object of class "hclust" into a tree of
                # class "phylo".
plot.phylo()    # Method for the (generic) plot() function that
```

```
                  # plots objects of class "phylo".
```

- The following function will compute **distances** between observations (in p-dimensional space, where p is the number of explanatory variables (X's) in the data set).

```
dist()    # Compute pairwise distances between observations (rows) in
          # a data frame.  Returns an object of class "dist".
```

- For example, to carry out a **hierarchical cluster analysis** using the (built-in) `USArrests` data set, type:

```
library(ape)
arr_dist <- dist(USArrests, method = "euclidean")
arr_clust <- hclust(arr_dist)
arr_tree <- as.phylo(arr_clust)
plot(arr_tree, cex = 0.5)
```

The result is Fig. 1, which is called a ***dendogram***.

Each **node** of the tree is a **cluster** formed by merging the two clusters of its daughter nodes. The steps proceed left to right.

The **leftward position** of a node, relative to the right side of the graph, is **proportional** to the **dissimilarity** between its two daughter nodes. As the steps proceed, more and more dissimilar clusters get merged.

A set of **clusters** (i.e. grouping of observations) is chosen by drawing a vertical line through the **dendogram** – the horizontal lines it crosses are the **clusters**. The left/right position of the vertical line can be used to control **how many clusters** the data set is split into.

Data Set: `wine`

The `wine` data set (from the `"rattle"` package) contains the results of a chemical analysis of wines grown in a specific area of Italy. Three types of wine are represented in the 178 samples, with the results of 13 chemical analyses recorded for each sample. The `Type` variable has been transformed into a categorical variable.

The data contains no missing values and consists of only numeric data, with a three class target variable (Type) for classification. The ten variables are:

`Type`	The type of wine, into one of three classes, 1 (59 obs), 2(71 obs), and 3 (48 obs).
`Alcohol`	Alcohol.
`Malic`	Malic acid.
`Ash`	Ash.
`Alcalinity`	Alcalinity of ash.
`Magnesium`	Magnesium.
`Phenols`	Total phenols.
`Flavanoids`	Flavanoids.
`Proanthocyanins`	Proanthocyanins.
`Color`	Color intensity.
`Hue`	Hue.
`Dilution`	D280/OD315 of diluted wines.
`Proline`	Proline.

Section 7.1 Exercises

Exercise 1 Here's a small data set.

```
my.data <- data.frame(X1 = c(3, 5, 4, 7),
                      X2 = c(6, 4, 9, 9),
                      X3 = c(1, 7, 2, 1))
rownames(my.data) <- c("Obs1", "Obs2", "Obs3", "Obs4")
my.data

##      X1 X2 X3
## Obs1  3  6  1
## Obs2  5  4  7
## Obs3  4  9  2
## Obs4  7  9  1
```

Compute the pairwise **distances** between observations (rows) in `my.data`:

```
my.data_dist <- dist(my.data, method = "euclidean")
my.data_dist
```

a) What's the distance (in a 3-dimensional space whose coordinates are `X1`, `X2`, and `X3`) between `Obs1` and `Obs2`?

b) Which two observations are ”closest” (**least dissimilar**) to each other?

c) Which two observations would be merged in the **first step** of a **hierarchical clustering** procedure?

Exercise 2 Compute the pairwise **distances** between states in the `USArrests` data set:

```r
arr_dist <- dist(USArrests, method = "euclidean")
arr_dist
```

What's the distance (in a 4-dimensional space whose coordinates are `Murder`, `Assault`, `UrbanPop`, and `Rape`) between **Florida** and **Alabama**.

Exercise 3 The `"rattle"` package contains a data set named `wine` (described above):

```r
# install.packages("rattle")
library(rattle)
head(wine)
```

The first column (`Type`) is a categorical variable, so it shouldn't be included in the cluster analysis:

```r
library(dplyr)

# The Type column gets removed:
wine2 <- select(wine, -Type)
```

Use `dist` to compute the **distances** between wines (in 13-dimensional space):

```r
wine_dist <- dist(wine2, method = "euclidean")
wine_dist
```

Use `wine_dist` to carry out a **hierarchical cluster analysis** on the `wines` data set (excluding `Type`), and produce the **dendogram**. **Report your R command(s)**.

7.1.2 *K* Means Clustering

- Another method of identifying **clusters** (**groupings**) of observations (when there's no response variable Y) is ***k means clustering***.

  Unlike *hierarchical clustering*, ***k* means clustering** requires knowing *in advance* the number of clusters (groups) ***k*** into which the set observations in a data set will be partitioned.

- To carry out ***k means clustering***:

  1. ”Guess” the **centers** of the ***k*** **clusters** (i.e. the ***cluster means***), either subjectively or randomly. The definition of ”**cluster centers**” will be given later.
  2. Given a current set of **cluster centers**, **assign** each observation to the **closest cluster center**. Each observation will now be in one of the k clusters.

3. For a given set of **assignments** of observations to **clusters**, compute the **centers** of these **clusters**. These new centers may have shifted a bit from their previous positions.
4. Repeat Steps 2 and 3 until **assignments** to clusters **don't change**, in which case the cluster centers won't change either.

- Consider the task of identifying $\boldsymbol{k = 3}$ clusters (groups) in the data shown and plotted below.

```
my.x1 <- c(5.2, 4.6, 5.9, 6.8, 10.5, 10.7, 8.6, 10.5, 14.1, 16.4, 14.3, 12.4)
my.x2 <- c(3.6, 4.7, 2.2, 4.5, 7.2, 7.3, 7.1, 9.9, 6.3, 4.2, 6.2, 3.3)
my.data <- data.frame(x1 = my.x1, x2 = my.x2)
```

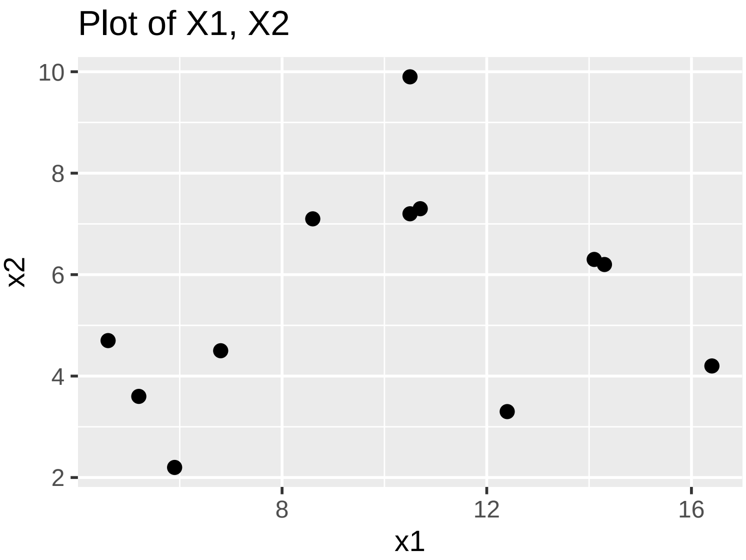

Figure 2

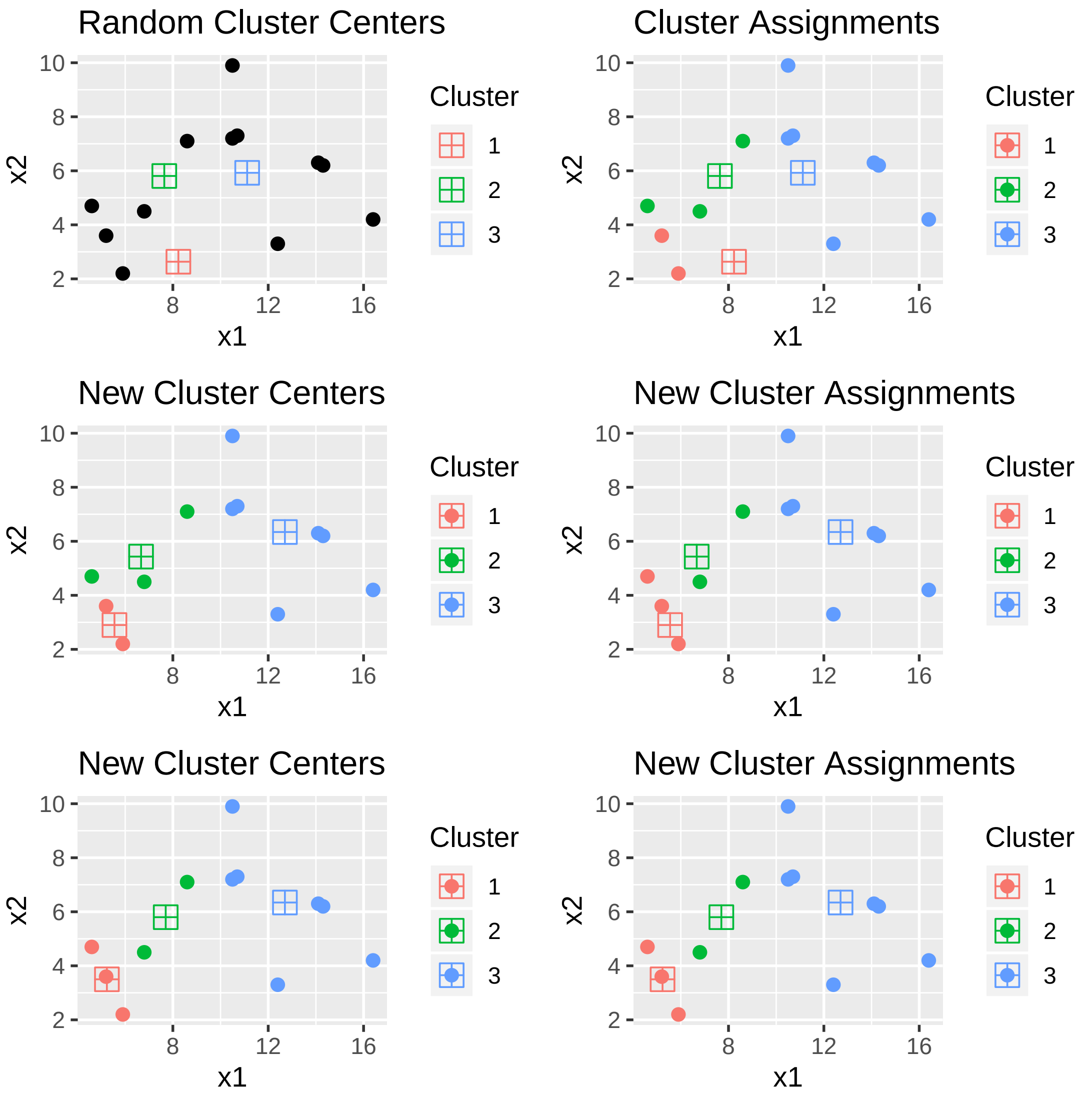

Figure 3

- The **coordinates** of the ***cluster centers*** are obtained by **averaging** each variable for observations in the cluster.

  For example, suppose these data are **one** of ***k*** **clusters**:

```
my.data <- data.frame(X1 = c(3, 5, 4, 7),
                      X2 = c(6, 4, 9, 9),
                      X3 = c(1, 7, 2, 1))
rownames(my.data) <- c("Obs1", "Obs2", "Obs3", "Obs4")
my.data
```

```
##      X1 X2 X3
## Obs1  3  6  1
## Obs2  5  4  7
## Obs3  4  9  2
## Obs4  7  9  1
```

Then the **cluster center** would be a point in a **3-dimensional** (`X1`, `X2`, `X3`) coordinate system having coordinates:

```
colMeans(my.data)

##   X1   X2   X3 
## 4.75 7.00 2.75
```

- The function below, from the `"mclust"` package, can be used to carry out ***k* means clustering**.

```
kmeans()        # Carry out k means clustering. Returns an object
                # of class "kmeans".
```

- For example, to carry out a ***k* means cluster analysis** using the (built-in) `USArrests` data set, type:

```
library(mclust)
# Set seed for random selection of initial cluster centers.
set.seed(25)
arr_clust <- kmeans(USArrests, centers = 3)
arr_clust

## K-means clustering with 3 clusters of sizes 14, 20, 16
## 
## Cluster means:
##      Murder  Assault UrbanPop     Rape
## 1  8.214286 173.2857 70.64286 22.84286
## 2  4.270000  87.5500 59.75000 14.39000
## 3 11.812500 272.5625 68.31250 28.37500
## 
## Clustering vector:
##        Alabama         Alaska        Arizona       Arkansas 
##              3              3              3              1 
##     California       Colorado    Connecticut       Delaware 
##              3              1              2              3 
##        Florida        Georgia         Hawaii          Idaho 
##              3              1              2              2 
##       Illinois        Indiana           Iowa         Kansas 
##              3              2              2              2 
##       Kentucky      Louisiana          Maine       Maryland
##              2              3              2              3
##  Massachusetts       Michigan      Minnesota    Mississippi
##              1              3              2              3
##       Missouri        Montana       Nebraska         Nevada
##              1              2              2              3
##  New Hampshire     New Jersey     New Mexico       New York
##              2              1              3              3
## North Carolina   North Dakota           Ohio       Oklahoma
##              3              2              2              1
##         Oregon   Pennsylvania   Rhode Island South Carolina
##              1              2              1              3
##   South Dakota      Tennessee          Texas           Utah
##              2              1              1              2
##        Vermont       Virginia     Washington  West Virginia
##              2              1              1              2
##      Wisconsin        Wyoming
##              2              1
##
## Within cluster sum of squares by cluster:
## [1]  9136.643 19263.760 19563.863
##  (between_SS / total_SS =  86.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"
## [4] "withinss"     "tot.withinss" "betweenss"
## [7] "size"         "iter"         "ifault"
```

Note that there are **three** clusters containing **14**, **20**, and **16** states each.

Section 7.1 Exercises

Exercise 4 Here are the data from above containing the variables `x1` and `x2` shown in Figs. 2 and 3:

```
my.x1 <- c(5.2, 4.6, 5.9, 6.8, 10.5, 10.7, 8.6, 10.5, 14.1, 16.4, 14.3, 12.4)
my.x2 <- c(3.6, 4.7, 2.2, 4.5, 7.2, 7.3, 7.1, 9.9, 6.3, 4.2, 6.2, 3.3)
my.data <- data.frame(x1 = my.x1, x2 = my.x2)
```

Carry out a ***k* means cluster analysis** on `my.data`, with $\boldsymbol{k = 3}$:

```
# So that everyone has the same randomly selected
# starting cluster centers:
set.seed(27)

my_clust <- kmeans(my.data, centers = 3)
my_clust
```

How many wines are in each of the three clusters (groups)?

Exercise 5 Recall that the `"rattle"` package contains a data set named `wine` (see description above):

```
# install.packages("rattle")
library(rattle)
head(wine)
```

The first column (`Type`) is a categorical variable, so it shouldn't be included in the cluster analysis:

```
# For select():
library(dplyr)

# The Type column gets removed:
wine2 <- select(wine, -Type)
```

Carry out a ***k* means cluster analysis** on the `wine2` data set, with $\boldsymbol{k = 3}$:

```
# So that everyone has the same randomly selected
# starting cluster centers:
set.seed(20)

wine_clust <- kmeans(wine2, centers = 3)
wine_clust
```

How many wines are in each of the three clusters (groups)?