CS/ISAT 344 Intelligent Systems
K-Means Clustering and Classification using R

Objective:

- Use statistical software (R) to find clusters or communities within a collection of categorical and quantitative data.

Instructions:

Preparation:

You should read "Introduction to R" Section 1 (Introduction & Preliminaries) before beginning this lab. A link to this document is available on Canvas.

Part 1. Basic K-Means Clustering

In this part of the lab you will use the K-Means Clustering algorithm within R to create clusterings for a set of 100 random data points. You'll first create a set of random two-dimensional points, then run the clustering algorithm and view the results to see the clusters that were created.

1. Launch R Studio from one of the lab computers. (Alternatively, you can install R and R Studio on a personal computer. Visit http://www.r-project.org/ and https://www.rstudio.com/ for downloads.)

2. Load the R graphics package, which will be used later to visualize the results of clustering.

```
require(graphics)
```

3. Generate a collection of 100 random two-dimensional points. The points will be stored in a matrix with columns labeled "x" and "y".

   First, generate the points:

```
x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
  matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))
```

   then give the columns labels: “x” and “y”.

```
colnames(x) <- c("x", "y")
```

4. Plot the points on an X-Y graph (which will appear in a separate window) and examine the distribution.

```
plot(x)
```

   Can you see a natural clustering of the points in two groups? Try to imagine where the clusters should be created.

5. Apply the K-Means clustering algorithm to cluster the data into 2 groups. K-Means will cluster the points to minimize the average distance of each point to its cluster centroid.

```
cl <- kmeans(x, 2)
```

6. Generate a plot of the two clusters, and identify the centers (which will be marked in the plot).

```
plot(x, col = cl$cluster)

points(cl$centers, col = 1:2, pch = 8, cex=2)
```

7. Repeat steps 5 and 6 several times with a higher number of clusters (the second parameter of kmeans) each time, to see the effect of finer clustering.

Hint: You can repeat prior commands using the up-arrow key.

Part 2. K-means clustering of real data

In this part of the lab you will use the K-Means Clustering algorithm to cluster a larger dataset containing data from bank loan applications.

Two files are provided in a zip file:

- bank-data-no-header.csv : the original application data, in column format
- bank-data-recoded.csv : the same data, recoded to numeric form and normalized

1. First, examine the bank data with a spreadsheet program (Libre Office or Excel) to get a sense of what it contains. The columns in the data, from left to right, are:

   - id
   - age
   - sex
   - region
   - income
   - married
   - children
   - car
   - savings acct
   - current acct
   - mortgage
   - pep (personal equity plan)

2. Load the data and label the columns.

   Before loading the data, use the Session / Set Working Directory option of R Studio to set your working directory to where the unzipped data files are located.

   Load the bank data into a vector named "bank".

   ```
   bank <- read.csv("bank-data-no-header.csv", header=FALSE)
   ```

   Assign column headings.

   ```
   colnames(bank) <- c("id","age","sex","region","income","married",
     "children","car","save_act","current_act","mortgage","pep")
   ```

3. Convert the bank data to matrix form, to enable matrix operations. Each row of data from the file will become one row in the matrix.

   ```
   mbank <- data.matrix(bank)
   ```

4. Apply the K-Means clustering algorithm to cluster the bank data into two groups (later we'll try increasing the number of groups).

   The extra set of parens will cause the results of the algorithm to be displayed. The centroids for each cluster will be listed at the top.

   ```
   (bankcl <- kmeans(mbank,2))
   ```

5. Plot the bank data and add center points for the clusters.

   The plot in this case will be a projection of the data onto two dimensions. The plot is essentially a “shadow” of a multidimensional space on a two-dimensional surface. Since only two dimensions (out of 12) will be visible in the plot, the clusters may not appear cohesive. It's hard to visualize the actual 12-dimensional space!

   ```
   plot(mbank, col = bankcl$cluster)

   points(bankcl$centers, col = 1:2, pch = 8, cex=2)
   ```

6. Repeat steps 4 and 5 with increasing numbers of clusters, and see how the plot changes.

Part 3. Classification of real data using learned clusters.

In this part of the lab you will use learned clusters developed from a subset of the bank data to perform classification of additional data.

- The first 100 rows of the bank data will be used as a training set for the classification algorithm.
- Then, the classification algorithm will classify the remaining 500 rows of the data to match the patterns it learned from the first 100 rows.
- Finally, you will compare the classification results for the 500 rows to the original clustering results (from Part 2) to see how well they match. This will indicate how well the algorithm is able to learn the pattern from the first 100 rows and apply it to the remaining rows.

1. Load the library required for classification.

```
require(class)
```

2. Load the bank data, name it as “bank”, and assign column headings.

```
bank <- read.csv("bank-data-recoded.csv", header=FALSE)

colnames(bank) <- c("age","sex","region","income","married",
  "children","car","save_act","current_act","mortgage","pep")
```

3. Convert the bank data to matrix form for further processing.

```
mbank <- data.matrix(bank)
```

4. This step will split the bank data into two parts. The first 100 rows ("train") will be used to train the classifier to recognize the cluster pattern that was previously established. The remaining 500 rows ("test") will be used to test how well the classifier has learned the pattern.

```
train <- rbind(bank[1:100,])
test <- rbind(bank[101:600,])
```

Pull out the original clusters assigned in Step 2 to the training set (first 100 rows).

```
cl <- bankcl$cluster[1:100]
```

5. Use the classifier algorithm ("knn") to classify the testing set using the pattern learned from the training set. The result vector, “classified”, will contain the class assigned to each row of the testing dataset.

```
classified <- knn(train,test,cl,k=3)
```

6. Finally, compare the original classification (cluster numbers) developed for the test data in Step 2 to the classifications produced by the classifier after it learned the pattern from the training data.

The closeness of the match will indicate how well the classifier has learned the clustering pattern from the training dataset.

```
clustered <- bankcl$cluster[101:600]

match <- matrix(0)
for (n in 1:500) {
  match[n] <- ifelse(clustered[n] == classified[n], "MATCH", "NO MATCH")
}
table(match)
```

Submission:

When you have finished the lab, take a screen shot of your final RStudio window and upload it to Canvas.

To take a screen shot on the MacOS, press Command+Shift+4 (all together), then press the space bar to turn the cursor into a camera, then click on the window to be saved.