Homework 12 Answers

Jessica Kraker

Possible Solutions to Selected Questions

Important: This document contains R code solutions and example answers for problems posed in the DS740: Data Mining homework. We are intentionally sharing this document with learners *who have already completed the associated homework*. We want you to be able to review and troubleshoot your code, as well as ask questions to prepare you for later assessments. *By using this document, you are accepting responsibility* **not** *to share it with anyone else, including other students in the course or the program who might not have completed the homework yet*. By upholding this agreement, you are helping us use this tool with learners in future terms.

From Problem 1: Clustering Methods

After loading in the data from the *wine.csv* file, store the 13 numeric variables in a data frame **x**.

Possible Code:

```
wine <- read.csv("wine.csv")
```

```
# Load packages here
library(dplyr)
library(ggformula)
library(ggdendro)
```

Question

Compute the means and standard deviations for all the variables. Compare the means and standard deviations between the thirteen variables, using these values to explain why it is a good idea to standardize the variables before clustering. Include at least one numeric computation to support your explanation.

Possible Answer: *Since the means and standard deviations show that the magnitudes and spreads of the variables are very different (in some cases, by several orders of magnitude), we should scale the variables to count equally in distance measures.*

Standardize the numeric variables in x and store the results in x.scale.

Possible Code:

```
x = as.matrix(wine)
x.scale=scale(x)
```

Hierarchical Clustering

Using Euclidean distance with **x.scale**, fit the hierarchical model using complete linkage. Produce a dendrogram of all the clusters; List an appropriate "height" (corresponding to the value of the distance measure) on the dendrogram for complete linkage that would produce three clusters.

Possible Code:

```
n = dim(x.scale)[1]
dist.x.scale = dist(x.scale, method="euclidean")
hc.complete = hclust(dist.x.scale,method="complete")
linktype = "Complete Linkage, Euclidean Distance, p=13"

# distance at which merge via complete linkage occurs
hc.4321 = hc.complete$height[(n-4):(n-1)]
hc.avg = (hc.complete$height[(n-3):(n-1)]+hc.complete$height[(n-4):(n-2)])/2

dend.form = as.dendrogram(hc.complete)
dend.merge
```

Question

Using Euclidean distance with **x.scale**, fit the hierarchical model using each of single linkage and average linkage, as well as complete linkage.

Possible Code:

```
hc.single = hclust(dist.x.scale,method="single")
ggdendrogram(as.dendrogram(hc.single), rotate = F,labels=F) +
  labs(title = "Single Linkage, Euclidean Distance, p=13")

hc.average = hclust(dist.x.scale,method="average")
ggdendrogram(as.dendrogram(hc.average), rotate = F,labels=F) +
  labs(title = "Average Linkage, Euclidean Distance, p=13")
```

Question

Using the linkage method you selected to best designate three types of wine, for the split of the data in three clusters, make a plot of *Alcohol* versus *Dilution* marked by the clusters (using three different colors and/or symbols, along with a legend).

Possible Code Answer

```
nclust=3; membHier = cutree(hc.complete,k=nclust)
colused = c("turquoise3", "red", "black")[membHier]; pchused = c(3,8,16)[membHier]
wine %>%
  ggplot( aes(x=Dilution, y=Alcohol))  +
  geom_point(color=colused, shape=pchused) +
  labs(title = "3 clusters joined by complete linkage")
```

Nonhierarchical Clustering

Now we consider using nonhierarchical (K-means) clustering to split the data into clusters.

Question

For K-means clustering, use multiple initial random splits to produce $K = 5$, 4, 3, and 2 clusters. Use tables or plots to investigate the clustering memberships across various initial splits, for each value of K. Which number(s) of clusters seem to produce very consistent cluster memberships (matching more than 95% of memberships between nearly all initial splits) across different initial splits? Select all K that apply.

Possible Code Answer

```r
nclust=3
memb1 = kmeans(x.scale,nclust)$cluster
memb2 = kmeans(x.scale,nclust)$cluster
tablematch <- table(memb1,memb2); tablematch

matchtotal <- sum(apply(tablematch,2,max))
matchtotal/n
```

Final Nonhierarchical Clustering

Starting with `set.seed(12)` to set the initial split, use nonhierarchical (K-means) clustering to determine cluster membership for three clusters (corresponding to the three types of wine). How many wine samples are in each cluster?

Possible Code Answer

```r
nclust=3
set.seed(12)
membNon = kmeans(x.scale,nclust)$cluster
table(membNon)
```

Question

For splitting into three clusters, compare the cluster membership of hierarchical clustering (using the linkage method you selected when creating three clusters to designate three types of wine) to the cluster membership of K-means clustering (using the cluster membership from the previous question). What proportion of the cluster memberships match between the hierarchical and nonhierarchical clustering methods?

Possible Code Answer

```r
tableout = table(membHier,membNon); tableout
matchtotal <- sum(apply(tableout,2,max))
matchtotal/n
```

From Problem 2: PCA methods

Load in the data from the **wine.csv** file. Store the 13 numeric variables in a data frame **x**.

We wish to use PCA to identify which variables are most meaningful for describing this dataset. Use the `prcomp` function, with `scale=T`, to find the principal components.

Possible Code

```
pc.info = prcomp(x, scale=T)
```

Questions

Look at the loadings for the first principal component.

What is the loading for the variable *Alcohol*?

Which variable appears to contribute the **least** to the first principal component?

What is the PVE for the first principal component?

How many principal components would need to be used to explain about 80% of the variability in the data?

Possible Code

```
pc.info$rotation[1,1] #Loadings

pc.info$rotation[,1]

CumulativePVE <- summary(pc.info)$importance[3,]; CumulativePVE
```

Scores

On a biplot of the data, wine sample #159 appears to be an outlier in the space of principal components 1 and 2. What are the principal component 1 and 2 score values (that is, the coordinates in the space of principal components 1 and 2) for wine sample #159?

Possible Code

```
pc1scores = pc.info$x[,1] # first principal component score vector

pc2scores = pc.info$x[,2] # second principal component score vector

cbind(pc1scores,pc2scores)[159,]
```

From Problem 3: Gene Expression Application

The goal is to distinguish between healthy and diseased tissue samples.

Data preparation - apply the following commands to properly prepare the data:

```
genes = read.csv("GeneExpression.csv",header=F)
genesNew = t(genes); dim(genesNew)

genesNew = scale(genesNew)
```

Question

Based on the goal of the study, explain why it makes sense to split the data into only two clusters.

Possible Answer: *Since we are aiming to distinguish into two groups, healthy and diseased, it makes sense to try to cluster the tissue samples into two clusters, aiming to correspond to those two types.*

Question

Use hierarchical clustering with **Euclidean distance** and **complete linkage** to split the 40 samples into *two* clusters. How many tissue samples from among samples 21–40 are in the second cluster?

Possible Code

```
dist.genes = dist(genesNew, method="euclidean")

hc.complete = hclust(dist.genes,method="complete")

memb = cutree(hc.complete,k=2); memb
```

Question

At time of diagnosis, the actual state for tissue samples 1-20 was "healthy", and tissue samples 21–40 were "diseased". What do the results of the clustering from the previous question tell us about the ability of the gene expression measurements to identify diseased tissue?

Possible Answer: *For this data set, the "answer" tells us that the gene expression measurements can perfectly distinguish diseased tissue from healthy tissue. Note that it would be useful to identify which gene expression measurements are the most useful in doing so.*

Questions

Use `prcomp` to compute the principal components for all 40 samples. How many *meaningful* (that is, explaining a non-zero proportion of the variability) principle components are able to be computed?

What is the cumulative PVE explained by the first two principal components?

Produce a biplot of the first two principal components and upload it on the Canvas quiz,

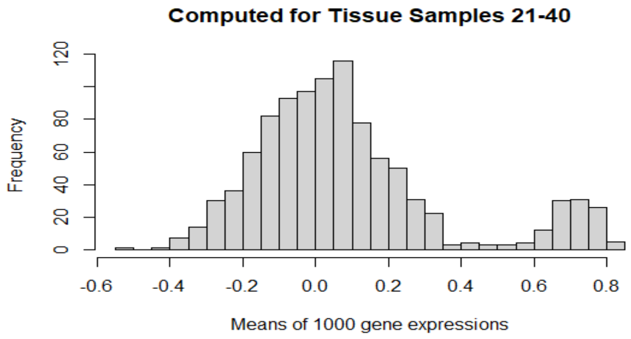
Possible Code

```
genes.pc = prcomp(genesNew)

summary(genes.pc)

CumulativePVE <- summary(genes.pc)$importance[3,]; CumulativePVE

biplot(genes.pc,choices=1:2,scale=0)
```

Variable Importance

Question

A histogram of **means2** (the 1000 means computed for the second half of samples for each of the 1000 gene expressions) is displayed below.



- **Describe** the distribution visualized in the histogram.
- **Discuss** how this pattern could occur, even when the gene-expressions (across the full data set) having been standardized.

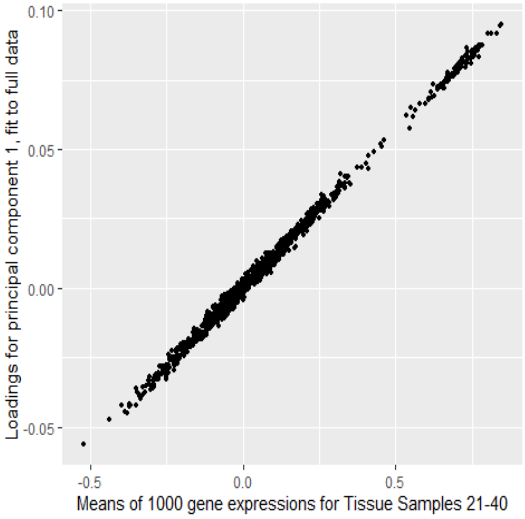
Possible Answer: *The distribution shows most of the gene-expression means centered around 0 (as expected, after centering the data), and the distribution of that bigger subset is approximately symmetric. But there is also a small peak of higher gene-expression means, which suggests that a subset of gene-expressions have higher means for the last 20 tissue samples as compared to the first 20 tissue samples.*

Question

Below we see a plot of the loadings for principal component 1 against **means2** (the 1000 means computed for the second half of samples for each of the 1000 gene expressions). Code to obtain the loadings for the first principal component is:

```
pc.loadings1 = genes.pc$rotation[,1]
```

Use values from the prior code definition of **means2**, along with this plot, to select two variables (from the list below) that are most important in the first principal component. You may also find the biplot to be helpful.



```
pc.loadings1[c(95,564,568,703,907)]
```

```
means2[c(95,564,568,703,907)]
```