Prelims Data Analysis TT 2019
Sheet 8

At the end of this exercise sheet there are Optional Practical Exercises in `R` and `Matlab`. It is **strongly recommended** that students do these exercises, but students should ask their college tutor whether to use `R` or `Matlab`. The course website has an Introduction to `R`, which students should work through before starting the `R` exercises.

1. Let $\mathbf{X}$ be an $n \times p$ data matrix, with x_i denoting the ith row of $\mathbf{X}$. Let $C_1, C_2, \ldots, C_K$ be a set of K clusters of observations. Let $\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$ be the mean of the observations in cluster k.

   (a) Prove the following identity for the within-cluster sum of squares for a given cluster C_k

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

   (b) Let v_k be a p-vector. Show that

   $$\sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - v_{kj})^2 = \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \mu_{kj})^2 + |C_k| \sum_{j=1}^{p} (\mu_{kj} - v_{kj})^2$$

2. Consider the following small dataset with $n = 6$ observations on $p = 2$ variables

   $$\mathbf{X} = \begin{bmatrix} -0.220 & -0.708 \\ 0.463 & -1.040 \\ 1.698 & 0.835 \\ -2.175 & 0.565 \\ 0.700 & 0.010 \\ 1.414 & 0.656 \end{bmatrix}.$$

   with associated Euclidean distance matrix $\mathbf{D}^{(6)}$. The lower diagonal of the matrix $\mathbf{D}^{(6)}$ is given by

   $\mathbf{D}^{(6)} =$

	1	2	3	4	5
2	0.76				
3	2.46	2.25			
4	2.33	3.09	3.88		
5	1.17	1.08	1.29	2.93	
6	2.13	1.95	0.34	3.59	0.96

   Apply agglomerative clustering using (a) Single Linkage and (b) Complete Linkage to this dataset. Use the notation $\mathbf{D}^{(k)}$ to denote the dissimilarities of k clusters of observations. You should write down the sequence of dissimilarity matrices at each stage of the algorithm, and draw an approximate dendrogram that summarizes the clustering process for each method.

Optional Practical Exercises using R

Students should carry out these practical exercises and produce a report summarizing the results of their analysis i.e. produce a document that contains the plots produced and hand this in to your tutor.

NOTE To run these exercises in R you will need to install a few packages called MASS, rgl, stats and car. To do this in RStudio click

Tools $->$ Install Packages

and then type in the names of the packages and install them. Make sure to click the box that says "install dependencies"

1. Download the Single Cell Genomics dataset from

   www.stats.ox.ac.uk/~sejdinov/teaching/data/single_cell.data

   Load the dataset into R using

   ```
   load("single_cell.data")
   ```

   This creates a data matrix object called X.

   Note you will need to change the command so that file includes the path to it's location on your computer.

   Load stats library using

   ```
   library(stats)
   ```

   Run the PCA using

   ```
   f3=prcomp(X, scale = TRUE, retx = TRUE)
   ```

   Run the the k-means algorithm with $K = 11$ on the first 11 PCs using

   ```
   fk = kmeans(f3$x[, 1:11], centers = 11)
   ```

   Create an interactive 3D plot of the data projected onto the first 3 PCs with the points labelled according to the results of the k-means clustering

   ```
   cols = c("black", "red", "green", "orange", "purple", "blue", "cyan", "yellow", "brown", "grey",
   "pink")

   scatter3d(x = f3$x[,1], y = f3$x[,2], z = f3$x[,3], point.col = cols[fk$cluster], pch = 16, surface
   = FALSE, xlab = "PC1", ylab = "PC2", zlab = "PC3")
   ```

   In fact the 300 cells that are of 11 distinct cell types and the cell labels are contained in the R object called Z. We can compare the true cell labels with the labels assigned by the k-means algorithm by cross-tabulating the labels (Note : we would not expect the actual numeric labels to agree since the numbering of clusters produced by k-means is arbitrary. A good agreement would be indicated by lots of 0s in the cross-table of labels.)

   ```
   table(Z, fk$cluster)
   ```

   If you run the k-means clustering with $K = 11$ but only using the top 3 PCs do you get as good a clustering of the dataset when compared to the true cell labels?

2. Download the EU indicators dataset from

   www.stats.ox.ac.uk/~sejdinov/teaching/data/eu.csv

   Load stats library using

   ```
   library(stats)
   ```

   Load the dataset into R using

   ```
   eu = read.csv("eu.csv",sep="", hea = T, row.names = 1)
   ```

   Note you will need to change the command so that file includes the path to it's location on your computer.

   Run hierarchical clustering with Single Linkage on the dataset as follows

   ```
   hc1 = hclust(dist(scale(eu[,-1])), method = "single")
   ```

   Plot the dendrogram using

```
plot(hc1)
```

Now change the clustering to use Complete Linkage (set `method = "complete"`) and Average Linkage (set `method = "average"`). How does this change the results?

Optional Practical Exercises using Matlab

Students should carry out these practical exercises and produce a report summarizing the results of their analysis i.e. produce a document that contains the plots produced and hand this in to your tutor.

NOTE If you get the error: Undefined function or variable 'princomp'." when you try to run the PCA commands, then you might not have the "Statistics and Machine Learning Toolbox" installed with your copy of matlab. If you rerun the matlab installer that you used for your previous matlab course, you should be able to add the "Statistics and Machine Learning Toolbox" to your installation by following the instructions here:
http://uk.mathworks.com/help/install/ug/install-mathworks-software.html
You might have to download the matlab installer again in case you've deleted it since your previous matlab course. You can download it again using this link: https://www.maths.ox.ac.uk/members/it/software-personal-machines/matlab

1. Download both files for the Single Cell Genomics dataset:

   www.stats.ox.ac.uk/~sejdinov/teaching/data/single_cell.csv
   www.stats.ox.ac.uk/~sejdinov/teaching/data/single_cell.Z.csv

   Load the dataset into Matlab using

   ```
   cells = readtable('single_cell.csv', 'Delimiter', 'space');
   Z = load('single_cell.Z.csv');
   ```

   Note you will need to change the command so that file includes the path to it's location on your computer.

   Run the PCA using

   ```
   X1 = table2array(cells(:, 2:end));
   for d = 1:size(X1, 2)
   X1(:, d) = X1(:, d) - mean(X1(:, d));
   X1(:, d) = X1(:, d)/std(X1(:, d), 1);
   end
   [coeff1, score1, latent1] = pca(X1);
   ```

   Run the the k-means algorithm with $K = 11$ on the first 11 PCs using

   ```
   fk = kmeans(score1(:, 1:11), 11);
   ```

   Create an interactive 3D plot of the data projected onto the first 3 PCs with the points labelled according to the results of the k-means clustering

   ```
   scatter3(score1(:, 1), score1(:, 2), score1(:, 3), [], fk);
   xlabel('PC1');
   xlabel('PC2');
   xlabel('PC3');
   ```

   In fact the 300 cells that are of 11 distinct cell types and the cell labels are contained in the vector Z. We can compare the true cell labels with the labels assigned by the k-means algorithm by cross-tabulating the labels (Note : we would not expect the actual numeric labels to agree since the numbering of clusters produced by k-means is arbitrary. A good agreement would be indicated by lots of 0s in the cross-table of labels.)

   ```
   crosstab(Z, fk)
   ```

   If you run the k-means clustering with $K = 11$ but only using the top 3 PCs do you get as good a clustering of the dataset when compared to the true cell labels?

2. Download the EU indicators dataset from

   www.stats.ox.ac.uk/~sejdinov/teaching/data/eu.csv

   Load the dataset into Matlab using

   ```
   eu = readtable('eu.csv', 'Delimiter', 'space');
   ```

   Note you will need to change the command so that file includes the path to it's location on your computer.

   Run hierarchical clustering with Single Linkage on a scaled version of the dataset as follows

   ```
   X2 = table2array(eu(:, 3:end));
   for d = 1:size(X2, 2)
   ```

```
X2(:, d) = X2(:, d) - mean(X2(:, d));
X2(:, d) = X2(:, d)/std(X2(:, d), 1);
end
hc1 = linkage(X2, 'single');
```

Plot the dendrogram using

```
dendrogram(hc1, 'Labels', table2cell(eu(:, 1)));
set(gca, 'XTickLabelRotation', 90);
```

Now change the clustering to use Complete Linkage (set the second argument of the `linkage` command to `'complete'`) and Average Linkage (set the second argument `'average'`). How does this change the results?