Homework 4 – Vector Quantization and Clustering

Heena Thakker

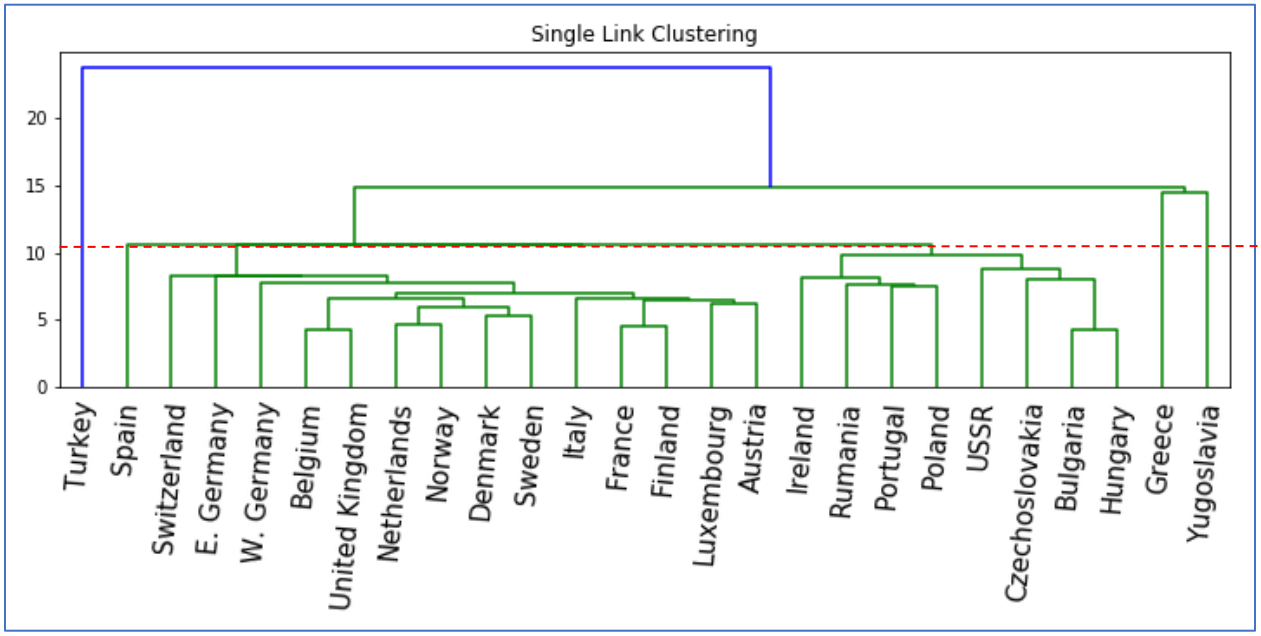
CS 498 Applied Machine Learning

PROBLEM 1

Part 1

This part was completed using python library scipy.cluster.hierarchy for Agglomerative clustering. The dendrograms obtained for single, complete and group average clustering are as follows -

Single Link Clustering

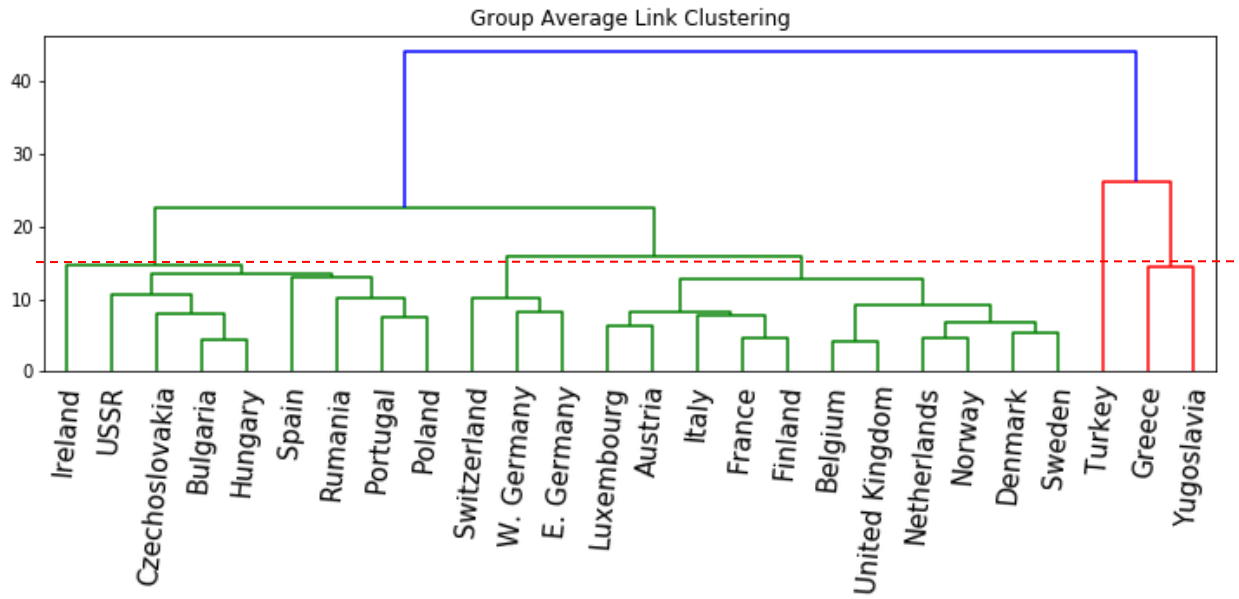


Slicing the diagram at the level indicated by the red dotted line yields the following clusters. Colors are used for the larger clusters to easily identify sub-clusters.

1. Turkey
2. Spain
3. Greece + Yogoslavia
4. Switzerland + E Germany + W Germany + Belgium + UK + Netherlands + Norway + Denmark + Sweden + Italy + France + Finland + Luxembourg + Austria

5. Ireland + Rumania + Portugal + Poland + USSR + Czechoslovakia + Bulgaria + Hungary

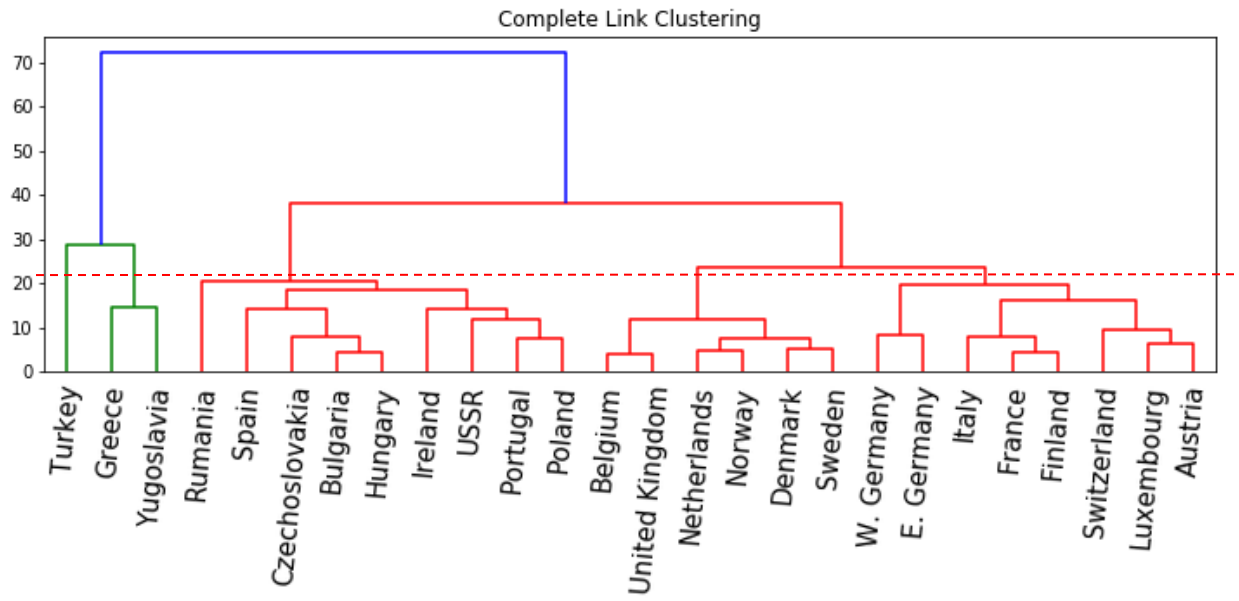
Group Average Clustering



Slicing the diagram at the level indicated by the red dotted line yields the following clusters. Colors are used for the larger clusters to easily identify sub-clusters.

1. Turkey
2. Greece + Yugoslavia
3. Switzerland + E Germany + W Germany
4. Luxembourg + Austria + Italy + France + Finland + Belgium + UK + Netherlands + Norway + Denmark + Sweden
5. Ireland + USSR + Czechoslovakia + Bulgaria + Hungary + Spain + Rumania + Portugal + Poland

Complete Link Clustering



Slicing the diagram at the level indicated by the red dotted line yields the following clusters. Colors are used for the larger clusters to easily identify sub-clusters.

1. Turkey
2. Greece + Yugoslavia
3. Belgium + UK + Netherlands + Norway + Denmark + Sweden
4. E Germany + W Germany + Luxembourg + Austria + Italy + France + Finland + Switzerland
5. Ireland + USSR + Portugal + Poland + Czechoslovakia + Bulgaria + Hungary + Spain + Rumania

All 3 clustering techniques broadly divide the countries by their proximity. In single link and group average clustering, groups 4 and 5 seem to represent western and eastern Europe respectively and to a certain extent the Warsaw Pact and NATO countries.

Complete Link clustering seems to differ in this that western Europe countries are further split into north western (group 3) and mid+south western (group 4) countries.

Turkey (almost Asia) and Greece & Yugoslavia consistently form a separate cluster in all 3 clustering techniques.

One of the noticeable differences is Spain – it is clustered with the eastern (Warsaw Pact) countries in both Group Average and Complete Link Clustering. Looking at the employment data it can be observed that this is because it has a high percentage for both manufacturing and agriculture like other eastern countries.

Similarly, Ireland, a neutral country has always been clustered with the Warsaw Pact (Eastern Countries) because of its diverse employment areas. These exceptions make better sense when looking at the table below which highlights the top most jobs for each country.

Country	Agr	Min	Man	PS	Con	SI	Fin	SPS	TC
Belgium	3.3	0.9	27.6	0.9	8.2	19.1	6.2	26.6	7.2
Denmark	9.2	0.1	21.8	0.6	8.3	14.6	6.5	32.2	7.1
France	10.8	0.8	27.5	0.9	8.9	16.8	6	22.6	5.7
W. Germany	6.7	1.3	35.8	0.9	7.3	14.4	5	22.3	6.1
Italy	15.9	0.6	27.6	0.5	10	18.1	1.6	20.1	5.7
Luxembourg	7.7	3.1	30.8	0.8	9.2	18.5	4.6	19.2	6.2
Netherlands	6.3	0.1	22.5	1	9.9	18	6.8	28.5	6.8
United Kingdom	2.7	1.4	30.2	1.4	6.9	16.9	5.7	28.3	6.4
Austria	12.7	1.1	30.2	1.4	9	16.8	4.9	16.8	7
Finland	13	0.4	25.9	1.3	7.4	14.7	5.5	24.3	7.6
Norway	9	0.5	22.4	0.8	8.6	16.9	4.7	27.6	9.4
Spain	22.9	0.8	28.5	0.7	11.5	9.7	8.5	11.8	5.5
Sweden	6.1	0.4	25.9	0.8	7.2	14.4	6	32.4	6.8
Switzerland	7.7	0.2	37.8	0.8	9.5	17.5	5.3	15.4	5.7
E. Germany	4.2	2.9	41.2	1.3	7.6	11.2	1.2	22.1	8.4
Ireland	23.2	1	20.7	1.3	7.5	16.8	2.8	20.8	6.1
Portugal	27.8	0.3	24.5	0.6	8.4	13.3	2.7	16.7	5.7
Bulgaria	23.6	1.9	**32.3**	0.6	7.9	8	0.7	18.2	6.7
Czechoslovakia	16.5	2.9	**35.5**	1.2	8.7	9.2	0.9	17.9	7
Hungary	21.7	3.1	**29.6**	1.9	8.2	9.4	0.9	17.2	8
Poland	31.1	2.5	25.7	0.9	8.4	7.5	0.9	16.1	6.9
Rumania	34.7	2.1	30.1	0.6	8.7	5.9	1.3	11.7	5
USSR	23.7	1.4	25.8	0.6	9.2	6.1	0.5	23.6	9.3
Yugoslavia	48.7	1.5	16.8	1.1	4.9	6.4	11.3	5.3	4
Greece	41.4	0.6	17.6	0.6	8.1	11.5	2.4	11	6.7
Turkey	66.8	0.7	7.9	0.1	2.8	5.2	1.1	11.9	3.2

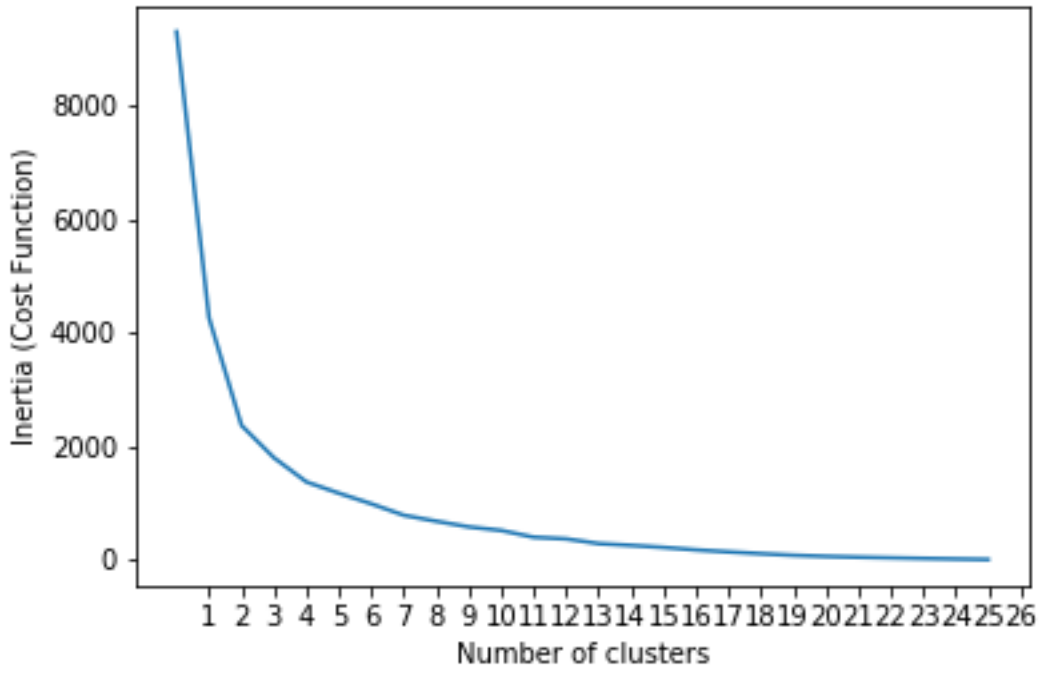
Part 2

Clustering the data using k-means (library sklearn.cluster.Kmeans) and k = 3 produces the following 3 clusters where 0, 1 and 2 are the class labels.

Belgium	0
Denmark	0
France	0
W. Germany	0
Italy	0
Luxembourg	0
Netherlands	0
United Kingdom	0
Austria	0

Finland	0
Norway	0
Sweden	0
Switzerland	0
E. Germany	0
Greece	1
Turkey	1
Yugoslavia	1
Ireland	2
Portugal	2
Spain	2
Bulgaria	2
Czechoslovakia	2
Hungary	2
Poland	2
Rumania	2
USSR	2

I think k = 3 to 5 is a reasonable number of clusters for this dataset because it produces meaningful clusters. Plotting the cluster size (1 to 27) against the cost function – inertia (Sum of squared distances of samples to their closest cluster center) produces the following plot –



A substantial decrease in cost is obtained when the number of clusters changes from 1 to 3. Post k = 5 there isn't a significant reduction in cost. Therefore depending upon the level of detail required in the clusters, a choice of 3 / 4 / 5 is reasonable.

I personally prefer k = 5 because it allows me to analyze the clusters in detail and then merge them as indicated by the dendrograms produced above.

PROBLEM 2

To vector quantize the accelerometer data I used k-means clustering to create the dictionary using 80% training data.

Random Decision Forest classifier was used to train the classifier.

Trying with segment size = 16/25/32/50 and k = 100/200/400/600/800/1000, one of the best error rates I obtained was with segment size = 16 and k = 600. The metrics for this run were –

Error Rate = 0.202312138728

Confusion Matrix =

```
[[ 2  0  0  0  0  0  0  0  0  0  1  0  0  0]
 [ 0 19  0  0  0  0  0  0  0  0  1  0  0  1]
 [ 0  0  6  0  0  0  0  0  0  1  0  0  0  0]
 [ 0  1  0  8  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 18  0  0  0  0  1  1  0  0  0]
 [ 0  0  0  0  0  1  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  1  0  0  0  0]
 [ 0  0  0  0  0  0  0 15  0  2  0  4  0  0]
 [ 0  0  0  0  0  0  0  1  0  3  2  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 20  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  1 16  2  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  5 16  0  0]
 [ 0  0  0  0  2  0  0  0  0  1  0  0  0  0]
 [ 0  1  0  0  0  0  0  0  0  0  2  0  0 17]]
```

However it is to be noted that this was a 1 off case. On multiple runs the average error rate obtained varied from 0.20 to 0.30.

Hence I decided to shortlist to the model with segment size = 32 and number of clusters = 480. This model produced metrics as noted below -

Error Rate = 0.254335260116

Accuracy = 0.745664739834

Confusion Matrix =

```
[[ 2  0  0  0  0  0  0  0  0  0  0  1  0  0]
 [ 0 18  0  0  0  0  0  0  0  0  1  0  0  2]
 [ 0  0  6  0  0  0  0  0  0  1  0  0  0  0]
```

```
[ 0  1  0  8  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0 19  0  0  0  0  0  1  0  0  0]
[ 0  0  0  0  0  1  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  1  0  0  0  0]
[ 0  0  0  0  1  0  0 11  0  3  1  5  0  0]
[ 0  0  0  0  0  0  0  2  0  2  2  0  0  0]
[ 0  0  0  0  0  0  0  0  0 20  0  0  0  0]
[ 0  0  0  0  1  0  0  2  0  1 14  2  0  0]
[ 0  0  0  0  0  0  0  0  0  0  6 15  0  0]
[ 0  0  0  0  2  0  0  0  0  1  0  0  0  0]
[ 0  4  0  0  0  0  0  0  0  0  1  0  0 15]]
```

References –

1. Data sources -
   http://lib.stat.cmu.edu/DASL/Datafiles/EuropeanJobs.html
   http://lib.stat.cmu.edu/DASL/Stories/EuropeanJobs.html

2. Code references for agglomerative clustering and dendrograms
   https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html

   https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.cluster.hierarchy.dendrogram.html

3. Code references for kmeans clustering
   http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

4. Python vs R
   https://www.dataquest.io/blog/python-vs-r/

5. Adjust margin in plot
   https://stackoverflow.com/questions/18619880/matplotlib-adjust-figure-margin

6. Piazza post for histogram
   https://piazza.com/class/jchzguhsowz6n9?cid=781

7. Code references for Random Forest Classifier
   http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html