Question 1 : (30 total points) Image data analysis with PCA

In this question we employ PCA to analyse image data

1.1 (3 points) Once you have applied the normalisation from Step 1 to Step 4 above, report the values of the first 4 elements for the first training sample in `Xtrn_nm`, i.e. `Xtrn_nm[0,:]` and the last training sample, i.e. `Xtrn_nm[-1,:]`.

First 4 elements for the first training sample in `Xtrn_nm` [-3.14e-06, -2.268e-05, -1.1797e-04, 4.0706e-04]
First 4 elements for the last training sample in `Xtrn_nm` = [-3.14e-06, -2.268e-05, -1.1797e-04, 4.0706e-04]

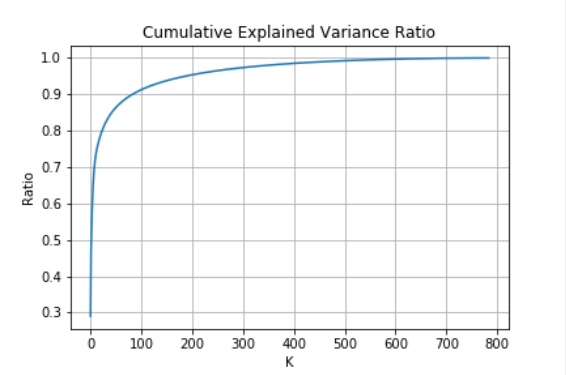
1.2 (4 points) Using `Xtrn` and Euclidean distance measure, for each class, find the two closest samples and two furthest samples of that class to the mean vector of the class.

The first column shows how each class would look like in general (mean) and we can easily determine the object. The second and the third columns show the samples that close to average one (first column). The last two columns show the samples that do not look similar compared to the general one in each of the category.

1.3 (3 points) Apply Principal Component Analysis (PCA) to the data of `Xtrn_nm` using sklearn.decomposition.PCA, and report the variances of projected data for the first five principal components in a table. Note that you should use `Xtrn_nm` instead of `Xtrn`.

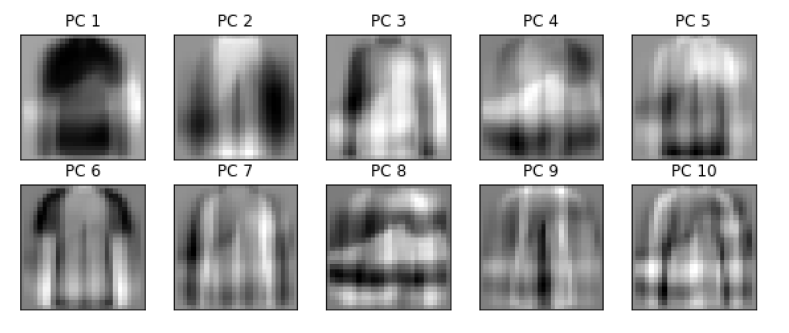
Principal Components	Variance
PC 1	19.810
PC 2	12.112
PC 3	4.106
PC 4	3.382
PC 5	2.625

1.4 (3 points) Plot a graph of the cumulative explained variance ratio as a function of the number of principal components, K, where $1 \leq K \leq 784$. Discuss the result briefly.



We are trying to find the K value which can project the original data to a lower-dimensional while preserving as much as of the original data's variation. It is clear that by setting K = 100 would preserve at least 90% variance of the data. We can also see that when K = 90, the cumulative variance is around 90% which is already quite good for PCA. It seems that making K larger does not increase significant after K = 400 (almost 100%). Therefore, using K = 400 has most the same result for the model compared to using the original data.

1.5 (4 points) Display the images of the first 10 principal components in a 2-by-5 grid, putting the image of 1st principal component on the top left corner, followed by the one of 2nd component to the right. Discuss your findings briefly.



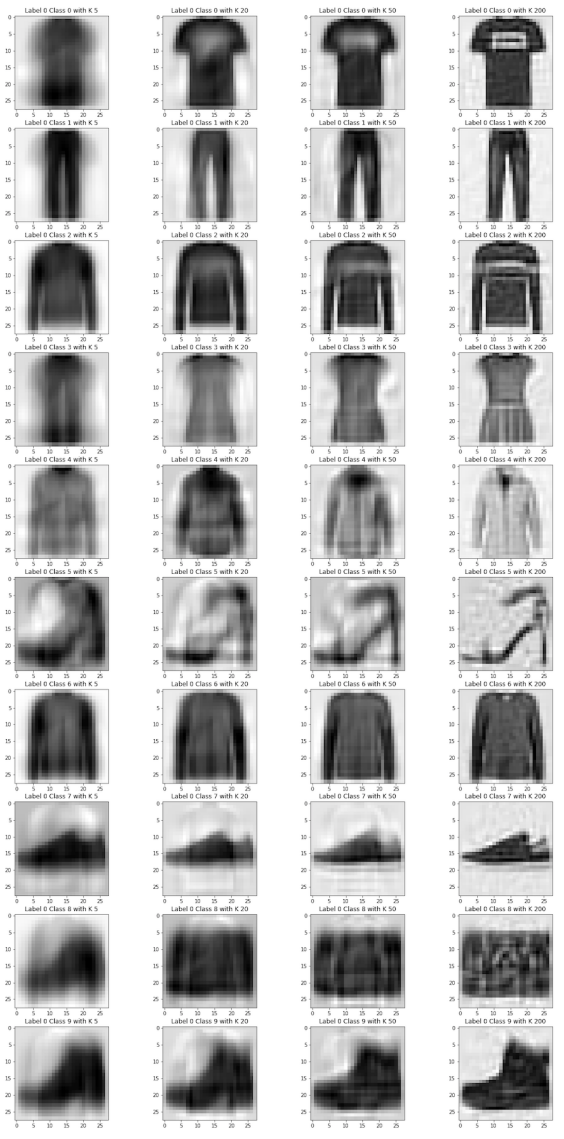
The principal components capture the what are the importance from the data so that it can separate the difference between different classes. As we can see from the principal components, they can find how the shirt and jacket look like. When we use more principal components to reconstruct the sample, we can get a better and clear image at the end.

1.6 (5 points) Using `Xtrn_nm`, for each class and for each number of principal components $K = 5, 20, 50, 200$, apply dimensionality reduction with PCA to the first sample in the class, reconstruct the sample from the dimensionality-reduced sample, and report the Root Mean Square Error (RMSE) between the original sample in `Xtrn_nm` and reconstructed one.

	PC = 5	PC = 20	PC = 50	PC = 200
	RMSE	RMSE	RMSE	RMSE
Class 0	25.615%	14.982%	12.763%	6.108%
Class 1	19.802%	14.045%	9.586%	3.708%
Class 2	19.870%	14.559%	12.513	%8.169%
Class 3	14.566%	10.724%	8.310%	5.613%
Class 4	11.821%	10.267%	8.818%	4.595%
Class 5	18.113%	15.870%	14.277%	9.039%
Class 6	12.948%	9.595%	7.196%	4.624%
Class 7	16.563%	12.767%	10.694%	6.191%
Class 8	22.340%	14.489%	12.353%	9.186%
Class 9	18.351%	15.113%	12.202%	7.317%

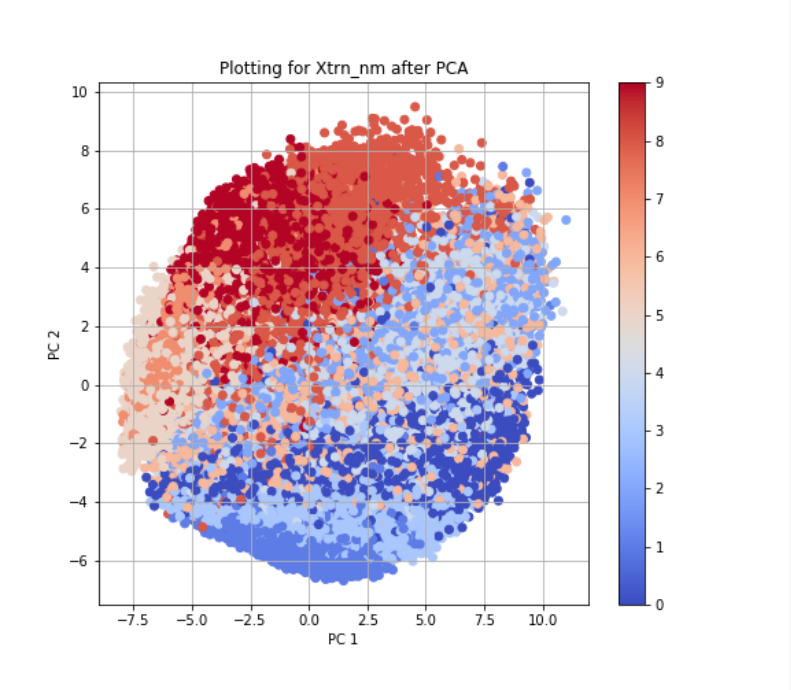
Question 1 continued on next page...

1.7 (4 points) Display the image for each of the reconstructed samples in a 10-by-4 grid, where each row corresponds to a class and each row column corresponds to a value of $K = 5,\ 20,\ 50,\ 200$.



We are taking the first instance of each class and try to lower the original dimension and see how well the lower dimension one can preserve the pattern of the original instance. With different K values to reconstruct the sample plus the mean of each class, we can compare the clearness of the first sample. From the figure 1.4, K captures 90% of the variance after K $\approx$ 100. Therefore, it is reasonable to see the samples are not so clear with K = 5 and K = 20. It is also worth mention that the lower the K value, the more ambiguous of the sample since it does not capture more variance of the data and it is like a mixture of different classes. Adding the mean can helps the sample to look more similar to the original class. We can tell that higher K values and adding the mean to the reconstructed sample looks more close to original sample as the k values captures more variance.

1.8 (4 points) Plot all the training samples (`Xtrn_nm`) on the two-dimensional PCA plane you obtained in Question 1.3, where each sample is represented as a small point with a colour specific to the class of the sample. Use the 'coolwarm' colormap for plotting.



We can see that there is a clear separation of classes from 0 to 9. Class 7, 8 and 9 are located on the top left corner and class 0 to 3 are on the middle to the bottom. Therefore, the 2 dimension plotting gives a solid classification on the visualisation that the clothes are on the top and top left corner while the shoes and the bags and shoes are on the bottom.

Question 2 : (25 total points) Logistic regression and SVM

In this question we will explore classification of image data with logistic regression and support vector machines (SVM) and visualisation of decision regions.

2.1 (3 points) Carry out a classification experiment with multinomial logistic regression, and report the classification accuracy and confusion matrix (in numbers rather than in graphical representation such as heatmap) for the test set.

Classification Accuracy: 84.01%

Confusion Matrix:

$$\begin{bmatrix} 819 & 3 & 15 & 50 & 7 & 4 & 89 & 1 & 12 & 0 \\ 5 & 953 & 4 & 27 & 5 & 0 & 3 & 1 & 2 & 0 \\ 27 & 4 & 731 & 11 & 133 & 0 & 82 & 2 & 9 & 1 \\ 31 & 15 & 14 & 866 & 33 & 0 & 37 & 0 & 4 & 0 \\ 0 & 3 & 115 & 38 & 760 & 2 & 72 & 0 & 10 & 0 \\ 2 & 0 & 0 & 1 & 0 & 911 & 0 & 56 & 10 & 20 \\ 147 & 3 & 128 & 46 & 108 & 0 & 539 & 0 & 28 & 1 \\ 0 & 0 & 0 & 0 & 0 & 32 & 0 & 936 & 1 & 31 \\ 7 & 1 & 6 & 11 & 3 & 7 & 15 & 5 & 945 & 0 \\ 0 & 0 & 0 & 1 & 0 & 15 & 1 & 42 & 0 & 941 \end{bmatrix}$$

Question 2 continued on next page...

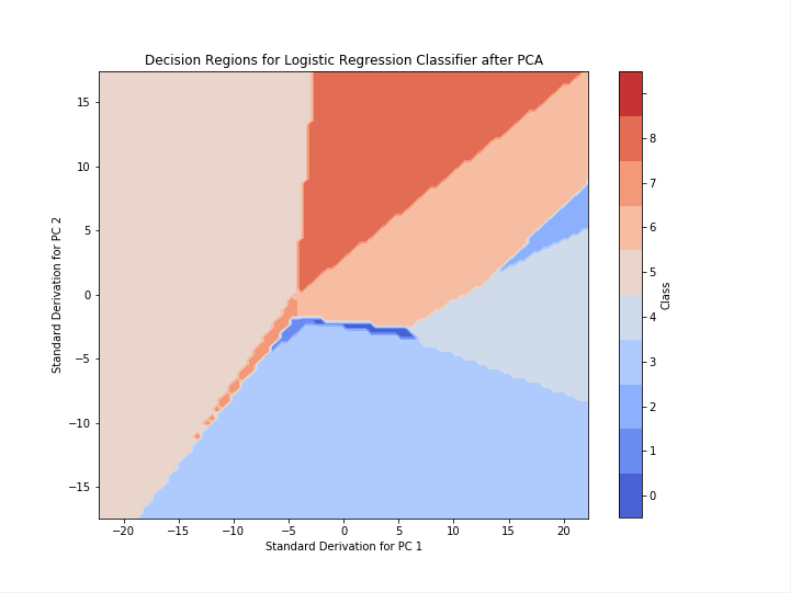
2.2 (3 points) Carry out a classification experiment with SVM classifiers, and report the mean accuracy and confusion matrix (in numbers) for the test set.

Classification Accuracy: 84.61%

Confusion Matrix:

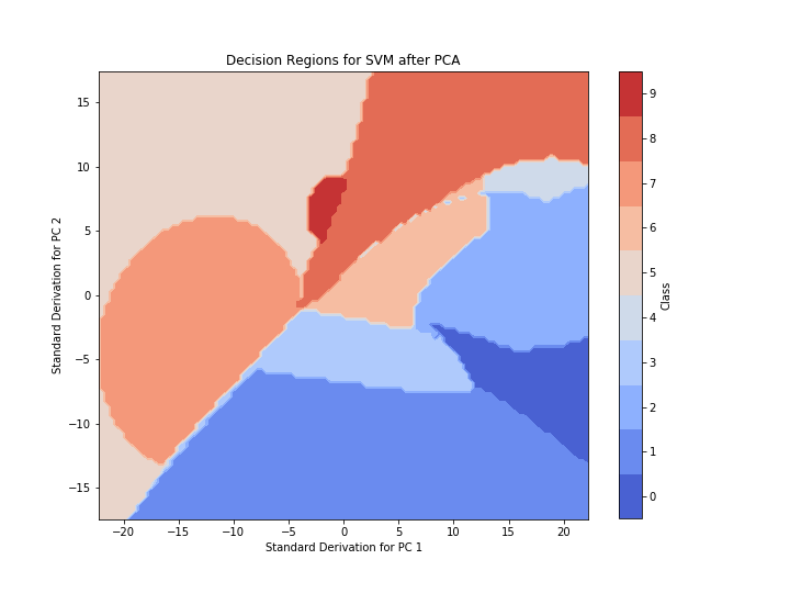
$$\begin{bmatrix} 845 & 2 & 8 & 51 & 4 & 4 & 72 & 0 & 14 & 0 \\ 4 & 951 & 7 & 31 & 5 & 0 & 1 & 0 & 1 & 0 \\ 15 & 2 & 748 & 11 & 137 & 0 & 79 & 0 & 8 & 0 \\ 32 & 6 & 12 & 881 & 26 & 0 & 40 & 0 & 3 & 0 \\ 1 & 0 & 98 & 36 & 775 & 0 & 86 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 914 & 0 & 57 & 2 & 26 \\ 185 & 1 & 122 & 39 & 95 & 0 & 533 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 34 & 0 & 925 & 0 & 41 \\ 3 & 1 & 8 & 5 & 2 & 4 & 13 & 4 & 959 & 1 \\ 0 & 0 & 0 & 0 & 0 & 22 & 0 & 47 & 1 & 930 \end{bmatrix}$$

2.3 (6 points) We now want to visualise the decision regions for the logistic regression classifier we trained in Question 2.1.



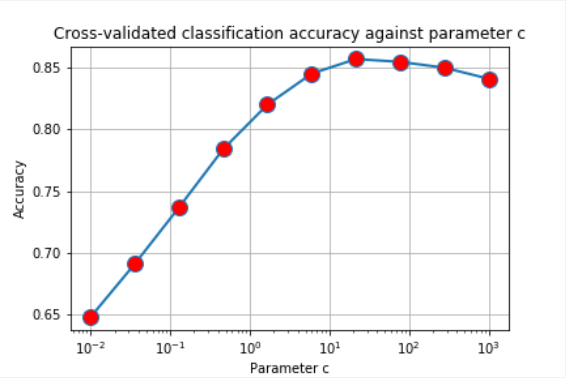
We can see that the Logistic Regression Classifier does not performing the way we are expecting. It only classifies the data into 8 classes after dimension reduction because Logistic Regression has lienar boundaries. It fails to capture the class 9 even though the algorithm performs well in high dimension and only produce a simple regions to determine the classes.

2.4 (4 points) Using the same method as the one above, plot the decision regions for the SVM classifier you trained in Question 2.2. Comparing the result with that you obtained in Question 2.3, discuss your findings briefly.



It is clear that the SVM classification can successfully separate the data into 9 classes. SVM tried to find the separating hyperplane that maximizes the distance of the closest points to the margin. Therefore, SVM is able to classify the classes with more details and better accurate.

2.5 (6 points) We used default parameters for the SVM in Question 2.2. We now want to tune the parameters by using cross-validation. To reduce the time for experiments, you pick up the first 1000 training samples from each class to create `Xsmall`, so that `Xsmall` contains 10,000 samples in total. Accordingly, you create labels, `Ysmall`.



As we can see from the graph, the highest obtained mean accuracy score is 85.65% when c = 21.544.

2.6 (3 points) Train the SVM classifier on the whole training set by using the optimal value of C you found in Question 2.5.

Data	Accuracy
Training Data	90.842%
Test Data	87.65%

Question 3 : (20 total points) Clustering and Gaussian Mixture Models

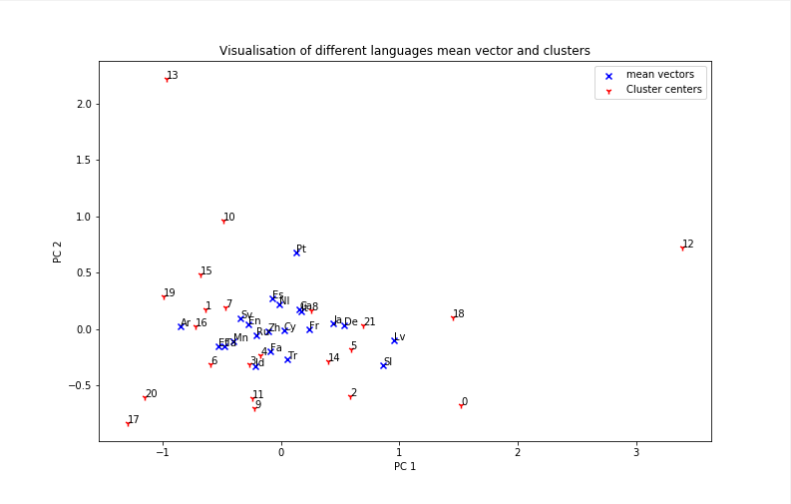
In this question we will explore K-means clustering, hierarchical clustering, and GMMs.

3.1 (3 points) Apply k-means clustering on `Xtrn` for $k = 22$, where we use sklearn.cluster.KMeans with the parameters `n_clusters=22` and `random_state=1`. Report the sum of squared distances of samples to their closest cluster centre, and the number of samples for each cluster.

Cluster	Number of samples
0	1018
1	1125
2	1191
3	890
4	1162
5	1332
6	839
7	623
8	1400
9	838
10	659
11	1276
12	121
13	152
14	950
15	1971
16	1251
17	845
18	896
19	930
20	1065
21	1466

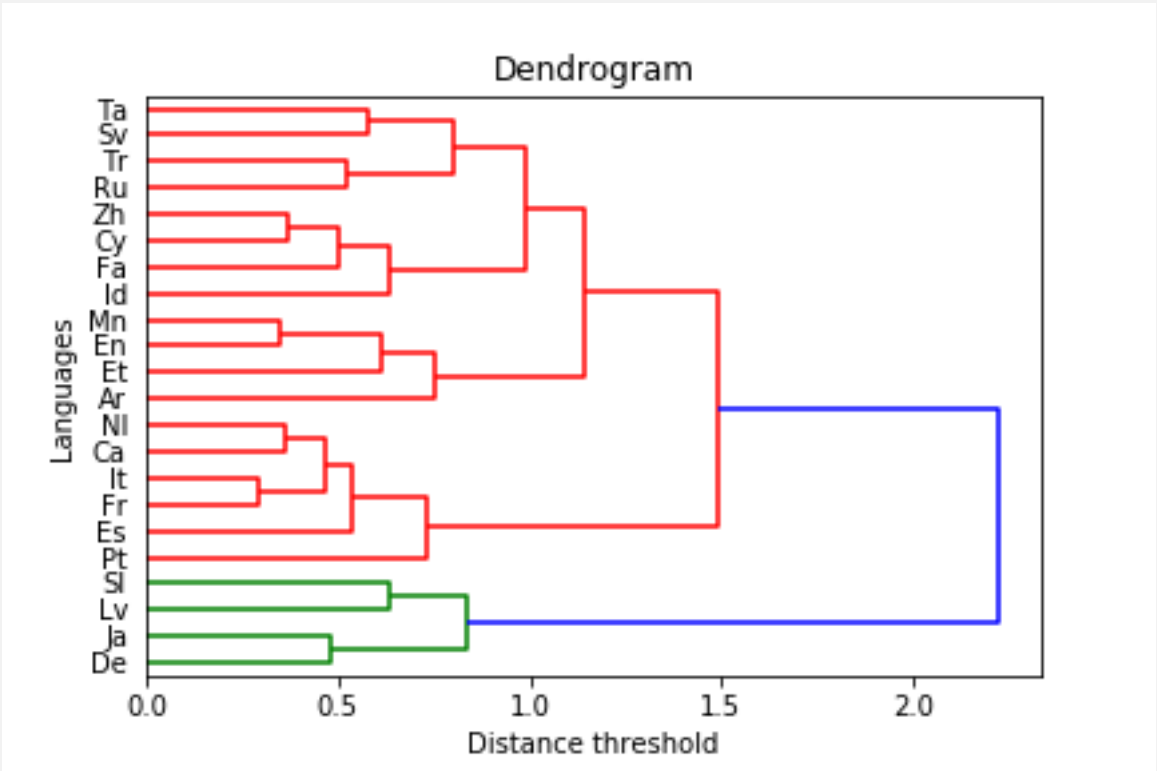
Sum of squared distances: 38185.817

3.2 (3 points) Using the training set only, calculate the mean vector for each language, and plot the mean vectors of all the 22 languages on a 2D-PCA plane, where you apply PCA on the set of 22 mean vectors without applying standardisation. On the same figure, plot the cluster centres obtained in Question 3.1.



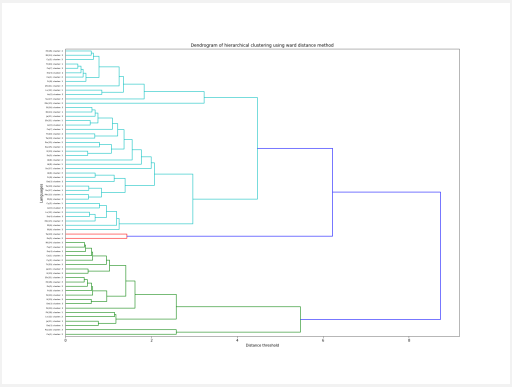
At first glance, most of the mean vectors are concentrated around (0,0). There are some languages have mean vector far from the group such as Portuguese(Pt), Latvian(Lv) and Slovenian(Sl). Moreover, the clusters have center quite separate from the mean vectors and some are terrible far from the mean vectors and other clusters.We can conclude that the K means clustering is not doing good to separate each class after the PCA. To improve the performance we can increase the k in PCA to capture more pattern from the languages.

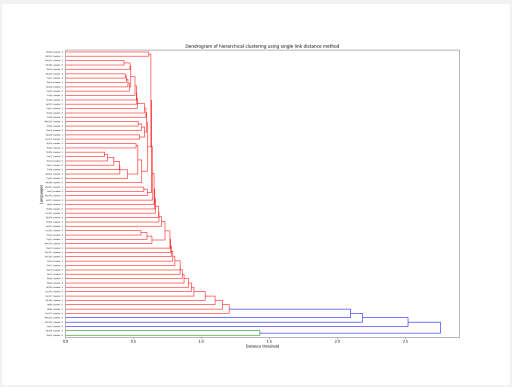
3.3 (3 points) We now apply hierarchical clustering on the training data set to see if there are any structures in the spoken languages.

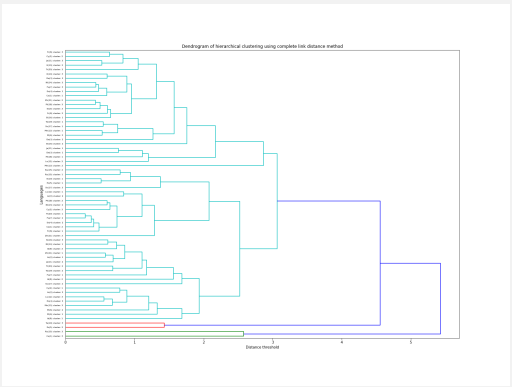


We are trying to cluster different languages given their mean vectors. The higher the distance threshold, the fewer the cluster class will be. Overall, we can see that there are two main clusters before merging at the distance threshold above 2. The biggest cluster contains 18 languages which the algorithm believes they are similar using ward distance method. Meanwhile, there are only four languages in the small cluster which the algorithm believes they are similar and quite different from the rest.

3.4 (5 points) We here extend the hierarchical clustering done in Question 3.3 by using multiple samples from each language.



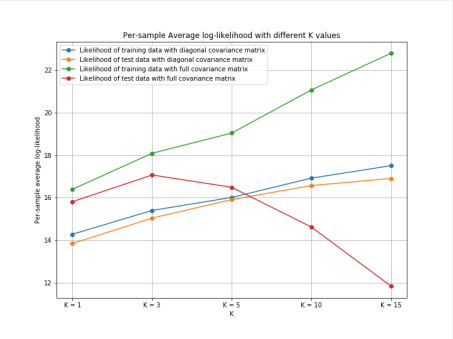




For using Single-link method, we can see there is a long chain to merge two similar clusters. Using Complete-link methods also has a large group clusters on one side but inside it is trying to find the minimum distance between farthest elements in the clusters which unlike the long chain from Single-link. Three methods put TA(19) cluster 3 and ES(5) cluster 2 together alone and not merging with other clusters until the end.

3.5 (6 points) We now consider Gaussian mixture model (GMM), whose probability distribution function (pdf) is given as a linear combination of Gaussian or normal distributions, i.e.,

	Average log-likelihood for two types of covariance matirx			
	Diagonal Covariance		Full Convariance	
K	Training Data	Test Data	Training Data	Test Data
K = 1	14.280	13.843	16.394	15.811
K = 3	15.398	15.041	18.086	17.066
K = 5	16.010	15.909	19.036	16.489
K = 10	16.917	16.568	21.062	14.622
K = 15	17.505	16.902	22.786	11.848



For diagonal covariance matrix, the likelihood of training and test data performs quite similar because the diagonal covariance matrix only look at the variance of each Gaussian sources to compute the likelihood. For full convariance matrix, the likelihood in training data increased around 6 when we increase the K from 1 to 15. On the other hand, the likelihood in test data drops significantly when K = 15. It could be applying the K too high and overfitted in the training data.