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

In this question we employ PCA to analyse image data

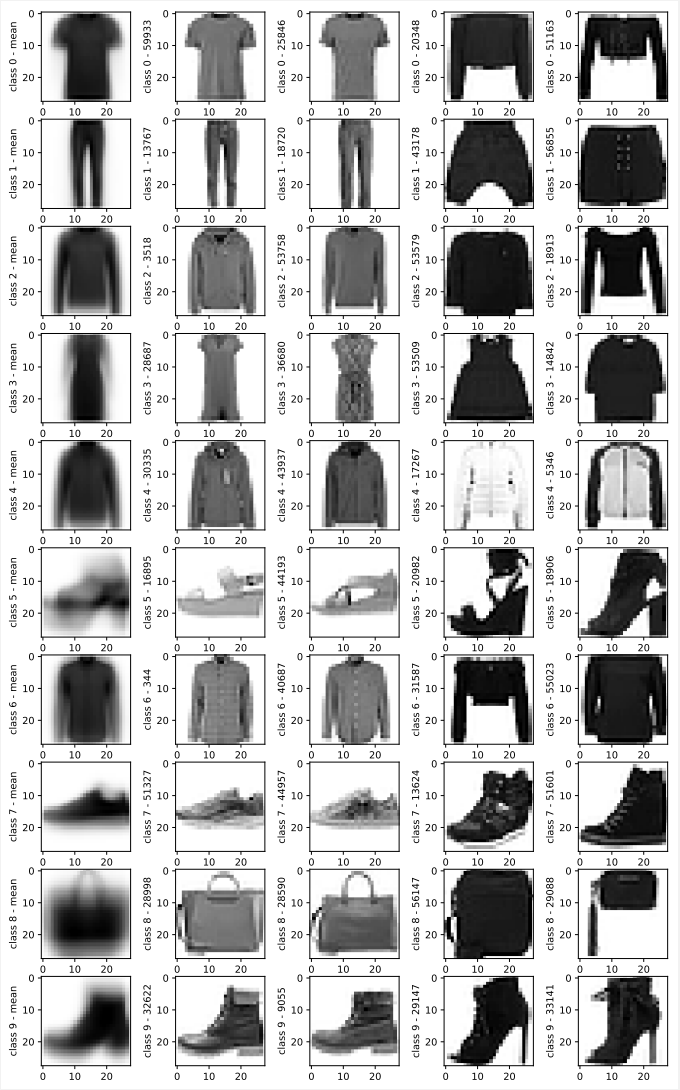
1.1 (3 points) Once you have applied the normalisation from Step 1 to Step 4 above, report the values of the first 4 elements for the first training sample in `Xtrn_nm`, i.e. `Xtrn_nm[0,:]` and the last training sample, i.e. `Xtrn_nm[-1,:]`.

The first 4 elements for the first training sample in Xtrn_nm shows below:
-3.13725490e-06, -2.26797386e-05, -1.17973856e-04, -4.07058824e-04
The first 4 elements for the last training sample in Xtrn_nm shows below:
-3.13725490e-06, -2.26797386e-05, -1.17973856e-04, -4.07058824e-04

Question 1 continued on next page...

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 image below shows the answer:



From the image above, the mean images in each class are blurred and represent main features of objects in each class. We can see that the closest 2 images has similar shape and size to mean image which means majority of samples in this class is similar to them. Mean image and last 2 images have very different shape and size due to different style of design and camera distance. This is why mean image in Class 5 has two different styles mix together and Class 8 has different size bags mix together. Other difference are orientation of objects in images like Class 7 sample 4 and color (darkness and lightness) of objects. It is clear to see 2 images closest to mean image normally has light color while last 2 images normally has dark color except Class 4 last 2 samples.

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`.

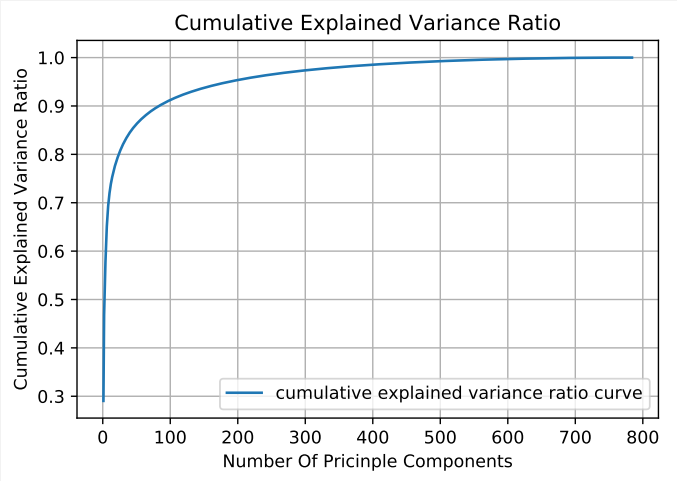
Results are presented in the table below. (variances rounded to 2 decimal places)

Principle Components	Explained Variance Of Projected Data
First	19.81
Second	12.11
Third	4.11
Fourth	3.38
Fifth	2.62

Question 1 continued on next page...

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.

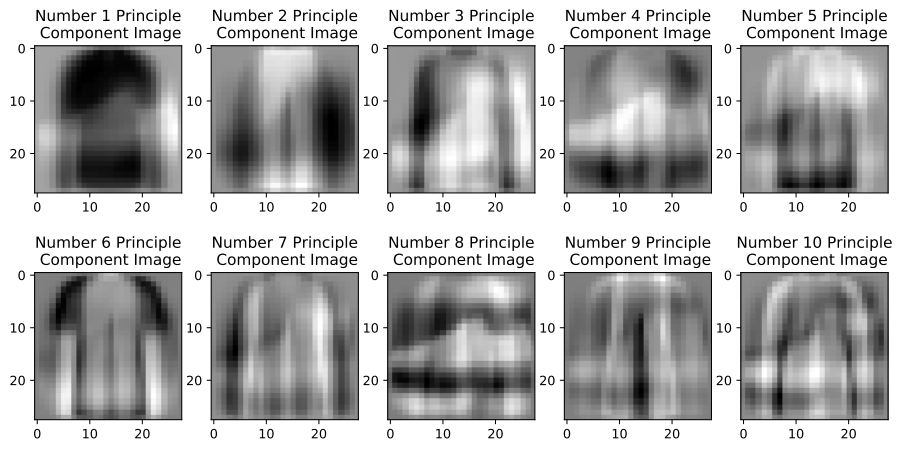
The image below shows the answer:



Cumulative explained variance ratio increases rapidly in the first few of principle components and then increases slowly in the rest of principle components. The ratio increases slower and slower as the number of principle components increase which means principle components become less and less important (component contains less and less variance). The cumulative explained variance ratio over 90%, 80% and 70% are when K=83, 23, 8 respectively. This means that the cumulative sum of variance in the first few principle components has taken up the majority of total variance.

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 image below shows the answer:



Each image represents some basis features of different class objects with dark or white color and background always be gray. Each image captures the structure that affects class separability. We can see that the first few principle components mainly distinguish different categories such as shoe, cloth or trousers and features are clear due to large variance on these components. They contain dark or white features that are similar to mean images of some classes. The last few images distinguish details of some classes and becomes blurred due to less variance on these components. White and black area are regions to distinguish something that contributes to a large amount of variance since their value are very positive or very negative which will significantly affect the variance of data which project on eigenimage. And each sample in data set can reform by linear combination of those eigenimages.

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.

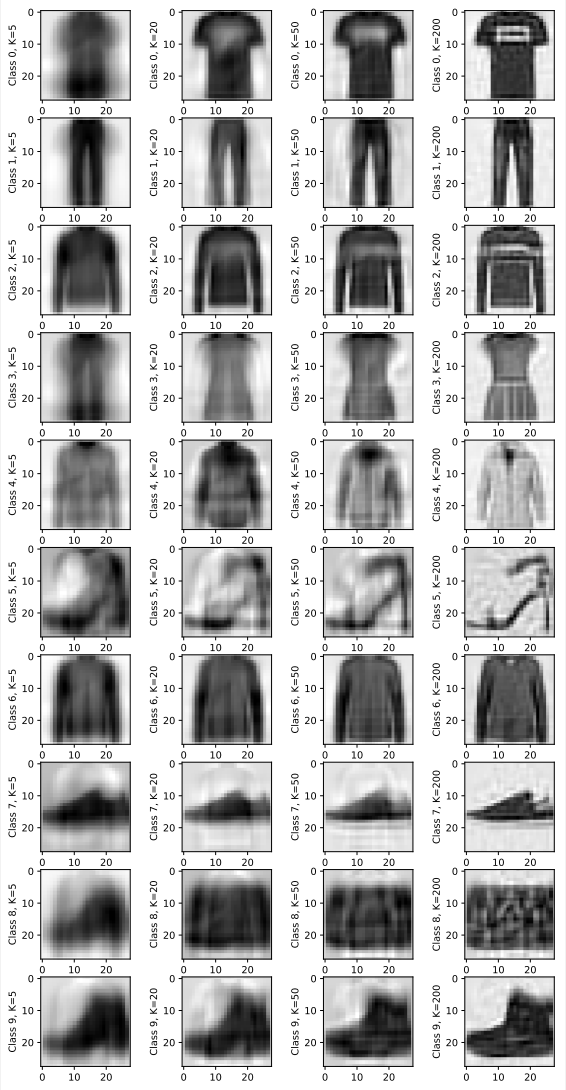
Results are presented in the table below. (RMSE rounded to 3 decimal places)

Class	RMSE(K=5)	RMSE(K=20)	RMSE(K=50)	RMSE(K=200)
0	0.256	0.150	0.127	0.061
1	0.198	0.140	0.095	0.038
2	0.199	0.146	0.124	0.080
3	0.146	0.107	0.083	0.056
4	0.118	0.103	0.088	0.047
5	0.181	0.159	0.143	0.089
6	0.129	0.096	0.072	0.046
7	0.166	0.128	0.107	0.064
8	0.223	0.145	0.124	0.091
9	0.184	0.151	0.122	0.072

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 $K = 5,\ 20,\ 50,\ 200$.

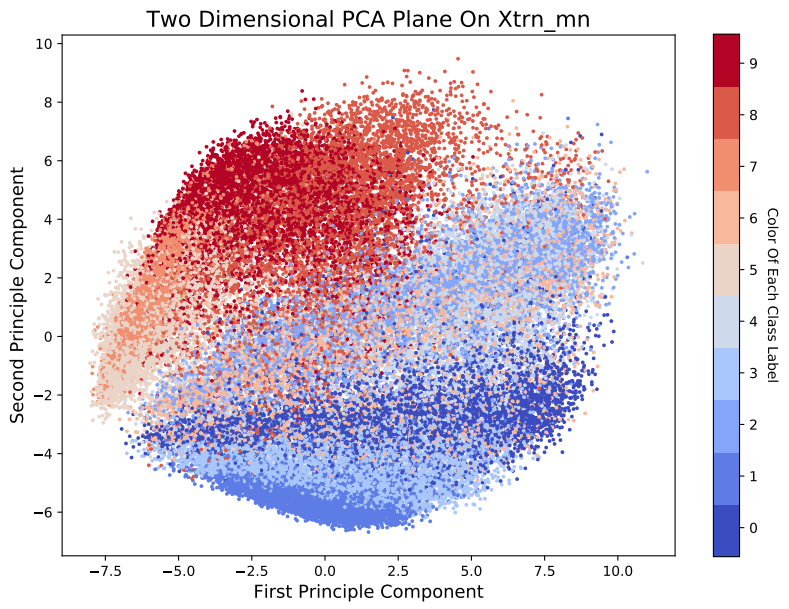
The image below shows the answer:



From image above, images in each class from left to right become more and more clear and we can see more and more details. The image with less number of principle components will be more blurred and easier to be affected by Xmean. We can see some of images in first column has clear Xmean background. They are also similar to mean images which only contains general features. As K becomes larger, more and more specific features appear and image becomes unique rather than only show general features. This is because each image above is made by linear combination of principle components plus a mean vector. The larger the number of principle components to made a image, the more variance will be in the image and image becomes more similar to the original one.

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.

The image below shows the answer:



The points of the same class are grouped together in 2D plane which means general feature in samples of each class are similar to each other. The classes with warm colors and classes with cool colors can be separated by a line from southwest to northeast. We can see that classes with warm colors normally have positive values in second principle component and classes with cool colors normally have negative values in second principle component. This is due to very white trousers and black shoe in eigenimage of second principle component which makes all types of shoes to positive and makes trousers or cloth (cloth is largely overlap with trousers in middle area) to negative. All classes are overlap with each other since 'true dimension' of data are higher than 2 dimensions.

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 for the test set: 0.840 rounded to 3 decimal places.
(0.8401 for original output)

	0	1	2	3	4	5	6	7	8	9
0	819	3	15	50	7	4	89	1	12	0
1	5	953	4	27	5	0	3	1	2	0
2	27	4	731	11	133	0	82	2	9	1
3	31	15	14	866	33	0	37	0	4	0
4	0	3	115	38	760	2	72	0	10	0
5	2	0	0	1	0	911	0	56	10	20
6	147	3	128	46	108	0	539	0	28	1
7	0	0	0	0	0	32	0	936	1	31
8	7	1	6	11	3	7	15	5	945	0
9	0	0	0	1	0	15	1	42	0	941

where left index represents the true labels of each class and upper columns names represent the predicted labels of each class.

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 for the test set: 0.846 rounded to 3 decimal places.
(0.8461 for original output)

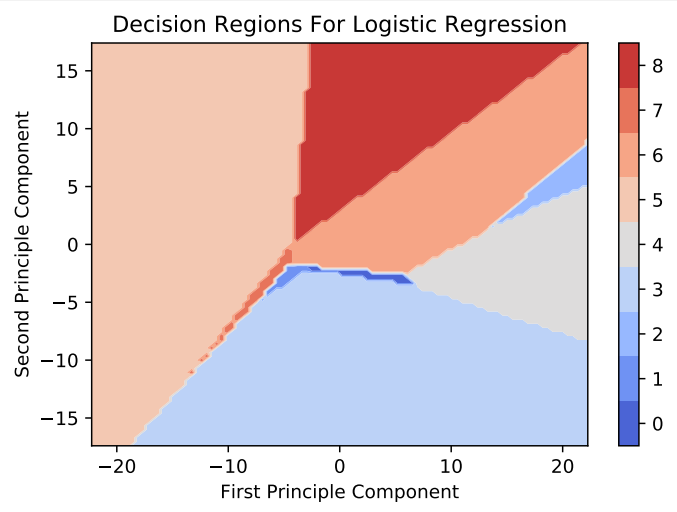
	0	1	2	3	4	5	6	7	8	9
0	845	2	8	51	4	4	72	0	14	0
1	4	951	7	31	5	0	1	0	1	0
2	15	2	748	11	137	0	79	0	8	0
3	32	6	12	881	26	0	40	0	3	0
4	1	0	98	36	775	0	86	0	4	0
5	0	0	0	1	0	914	0	57	2	26
6	185	1	122	39	95	0	533	0	25	0
7	0	0	0	0	0	34	0	925	0	41
8	3	1	8	5	2	4	13	4	959	1
9	0	0	0	0	0	22	0	47	1	930

where left index represents the true labels of each class and upper columns names represent the predicted labels of each class.

Question 2 continued on next page...

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

The image below shows the answer:

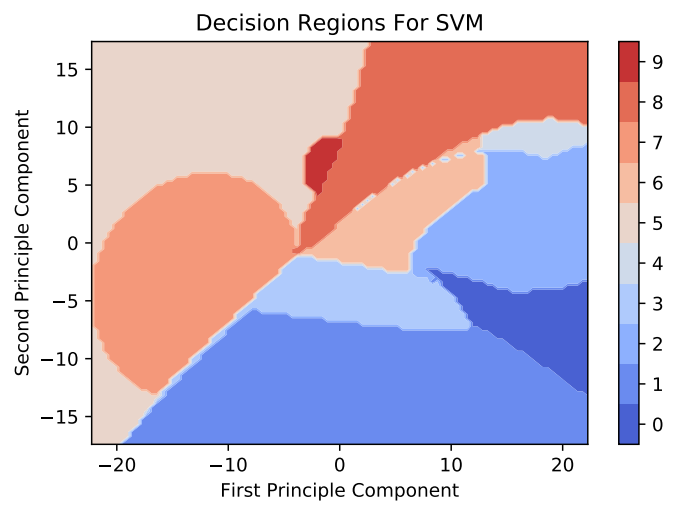


There are just 9 classes occur. The decision regions have straight line decision boundaries since logistic is a linear classifier. Class 1 and 2 and 7 almost disappear and Class 9 is missing since the data in these classes are not linearly separable on original dimensions. From 1.8, we know that those classes in 2D plane overlap with others. Inverse transformation of data that classes are not linear separable in 2D plane to original dimensions still be a 2D plane with non-linear separable data. Therefore, logistic regression which is a linear classifier can not classify these classes well in original dimensions. In general, inverse transformation from 2D data to original dimensions will lost lots of variance (only first two components information remain) so it is difficult for classifier to identify different classes.

Question 2 continued on next page...

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.

The image below shows the answer:

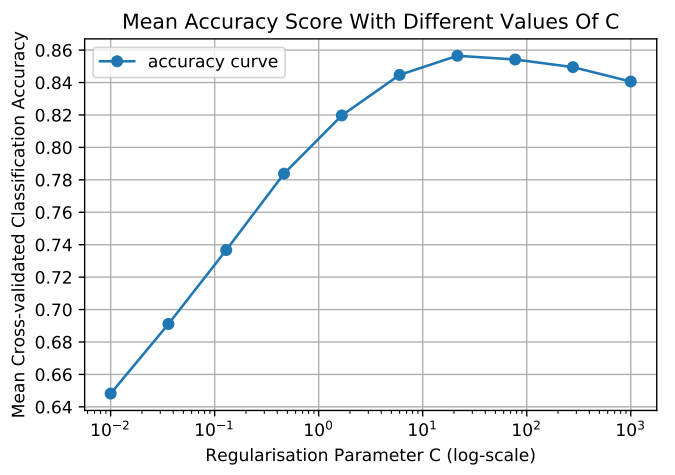


The decision boundaries of SVM classifier are not linear since SVM with Gaussian kernel is non linear classifier. The shape of decision regions in SVM is very different compared to Logistic. Class 9 occur in this region but not occur in Logistic. Class 1,2,7 which almost disappear in Logistic have become larger. This is because data is not linear separable on original dimensions (inverse transformation of non linearly separable data in 2D plane to original dimensions are still be non linearly separable 2D plane in high dimensions) and SVM classify data by using Gaussian kernel to find a hyperplane in another vector space to make non linearly separable data separable. That is why SVM has better decision regions than Logistic Regression.

Question 2 continued on next page...

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`.

The image below shows the answer:



The highest obtained mean accuracy score is 0.857 rounded to 3 decimal places.
The value of C corresponds to this accuracy is 21.54 rounded to 2 decimal places.

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.

Results are presented in the table below. (rounded to 3 decimal places)

Training Accuracy	Test Accuracy
0.908	0.877

Original result on training set by using optimal value of C is 0.9084.
Original result on test set by using optimal value of C is 0.8765.

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.

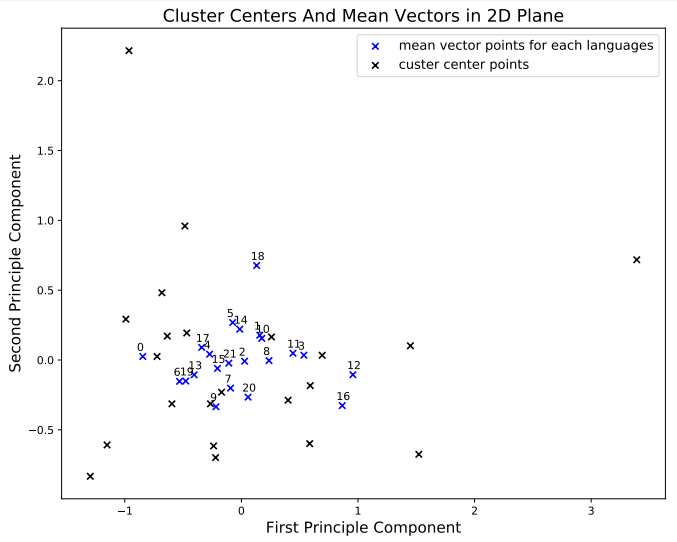
Results are presented in the table below.

Cluster Number	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 of samples to their closest cluster centre is 38185.82 rounded to 2 decimal places.

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.

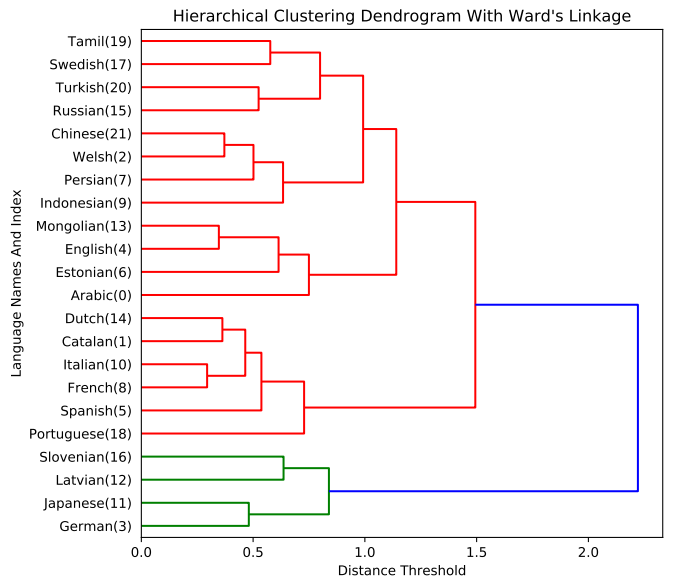
The image below shows the answer:



From image above, we can see that mean vector of each language projected into 2D plane close to each other. This means that the general feature of speech information in different languages are similar. But mean vectors and cluster centers are not similar. Cluster centers widely spread while mean vectors group into a small region. Some cluster centers are far away from other points. This is because some clusters has been dragged by outliers. Different initial position of centers will results in different position of cluster centers because Kmeans is sensitive to outliers. So it cannot guarantee the positions of 22 clusters will get best match with 22 languages. Kmeans is unsupervised learning so we cannot ensure which cluster belongs to which language.

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.

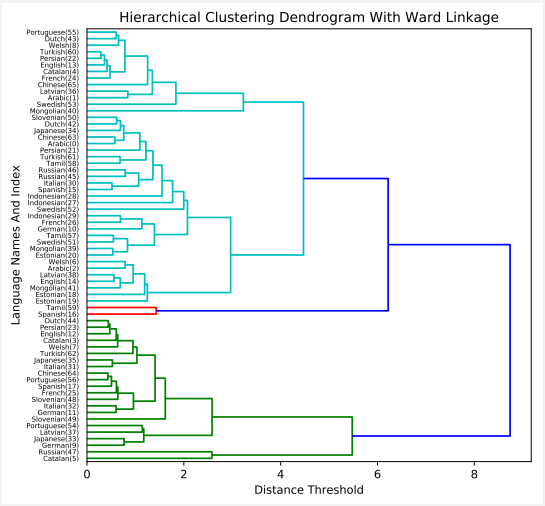
The image below shows the answer:

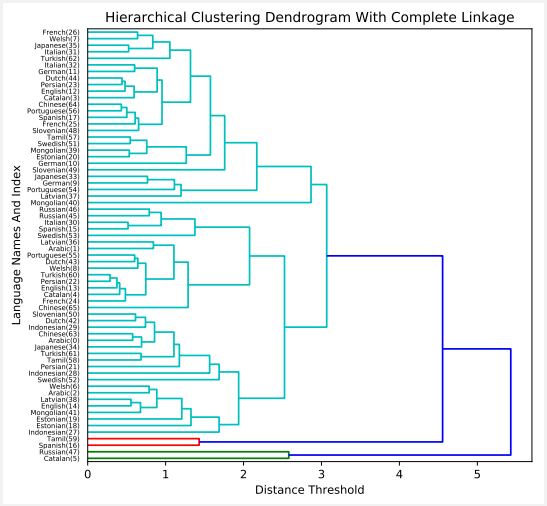


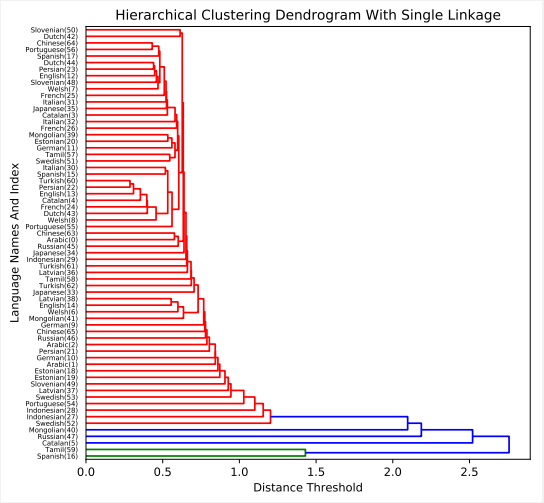
Ward Linkage measures distance of clusters by minimizing the change of within-cluster variances before and after merge. So it forms small groups first and combines them later. Ward can group the languages in the same geographical region into the same cluster such as European language cluster between Dutch and Portuguese. Those languages may have the same ancestor language, so they have similar speech information. Some of languages are in different regions but cluster together such as Chinese and Welsh, Japanese and German. This may because they have similar tone of voice.

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

The images below show the answer:



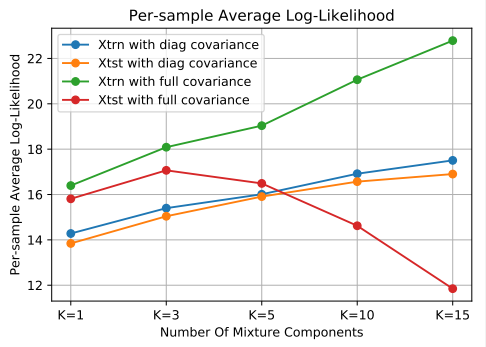




Ward and Complete has similar cluster pattern that forces spherical clusters but Single forces a long chains. This difference is caused by different similarity measure between clusters. Ward Linkage measures similarity by within-cluster variance change before and after merge. Complete Linkage measures similarity by distance between furthest elements in clusters. Single Linkage measures similarity by distance between closest elements in clusters. We can see that there are two outlier clusters (47,5 and 59,16). This is caused by outliers in their language samples drag the centers in KMeans. The threshold distance are different. maximum distance in Single is the smallest, in Ward is the largest and in Complete is middle.

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.,

The images below show the answer:



	K=1	K=1	K=3	K=3	K=5	K=5	K=10	K=10	K=15	K=15
	full	diag	full	diag	full	diag	full	diag	full	diag
Xtrn	16.39	14.28	18.09	15.40	19.04	16.01	21.06	16.92	22.79	17.50
Xtst	15.81	13.84	17.07	15.04	16.49	15.91	14.62	16.57	11.85	16.90

For GMM with full covariance matrix, the likelihood of training data increases when K increases but the likelihood of testing data decreases after K=3 because more components added shrink size of Gaussians so the model will be more likely to be overfitting. Both likelihood of training and testing data in GMM with diagonal covariance matrix increase when K increases and likelihood values are similar. This is because diagonal will lose information provided by correlation of dimensions of data and make dimensions of data indepedent to each other. Therefore the results will be less affected by increase in components.