Question 1 : (41 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 table shows the result.

sample \ element	1st	2nd	3rd	4th
the first training sample	-3.137e-06	-2.268e-05	-1.180e-04	-4.071e-04
the last training sample	-3.137e-06	-2.268e-05	-1.180e-04	-4.071e-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 figure displays the result.

The samples in the same class are diverse. For each class, the two closest samples to the mean vector of the class are almost the same as the mean vector. The two furthest samples to the mean vector of the class are a little different from the mean vector. But it can be identified that they are of the same class. Some different classes have similarities.

Question 1 continued on next page...

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`.

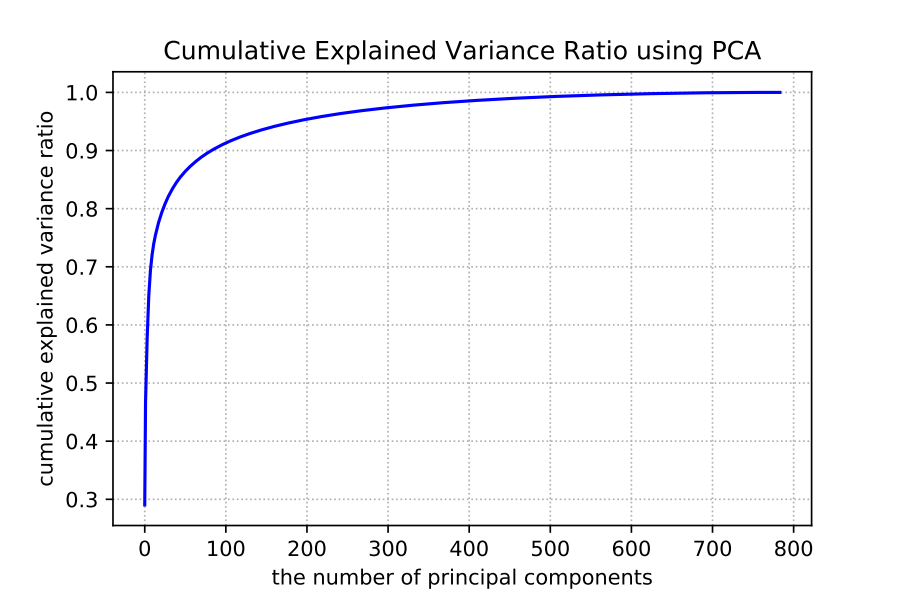
The table shows the result.

principal component	1st	2nd	3rd	4th	5th
variance	19.810	12.112	4.106	3.382	2.625

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 graph displays the result.

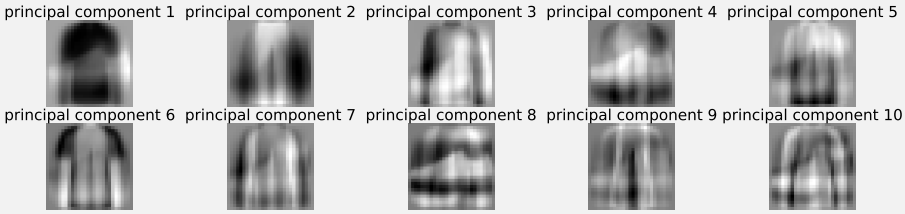


From the graph, we can find that the cumulative explained variance ratio of the first 200 principal components reached about 95%. The dimensionality reduction to this level can reflect most of the information. So the higher the explained variance ratio, the more important the principal component.

Question 1 continued on next page...

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 figure displays the first 10 principal components.



Each principal component contains different features. The more principal components, the greater the amount of information contained.

Question 1 continued on next page...

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.

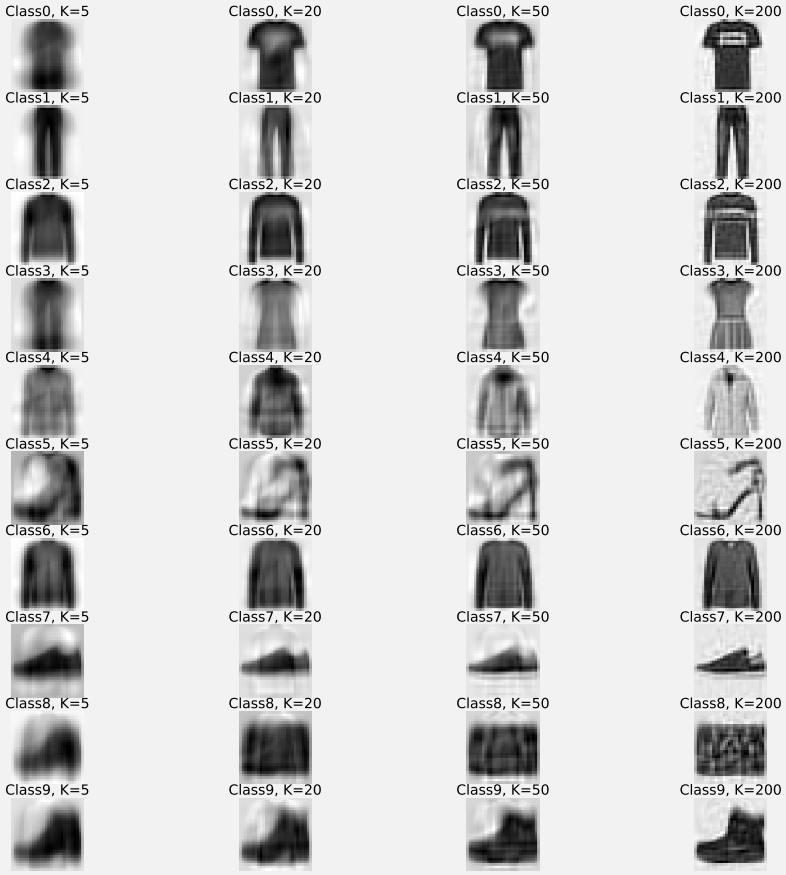
The table shows the result.

Class \ K	5	20	50	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 figure displays the result.

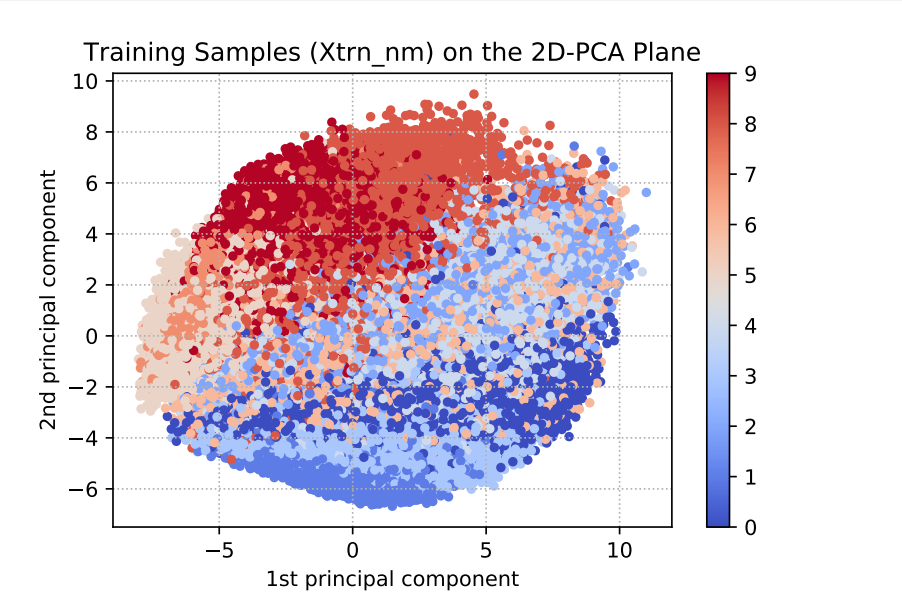


As the number of principal components increases, the information contained in the sample also increases, and it presents more obvious features. Each column of the image in the picture is clearer than the previous column.

Question 1 continued on next page...

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 figure displays the result.

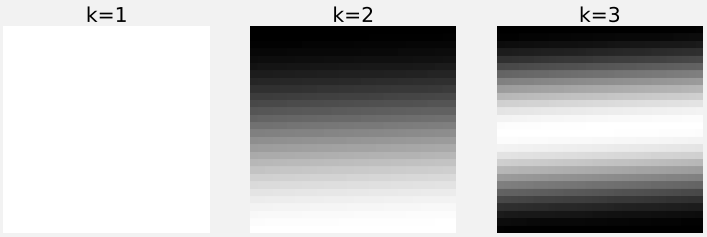


There are a lot of overlaps between classes. Due to the high dimensionality of the original data, representing them in two dimensions will lose many features. Therefore, the classification boundaries are not obvious.

Question 1 continued on next page...

1.9 (4 points) We here compare PCA with Discrete Cosine Transform (DCT), which is another technique for dimensionality reduction with cosine functions at different frequencies.

The figure displays the result.

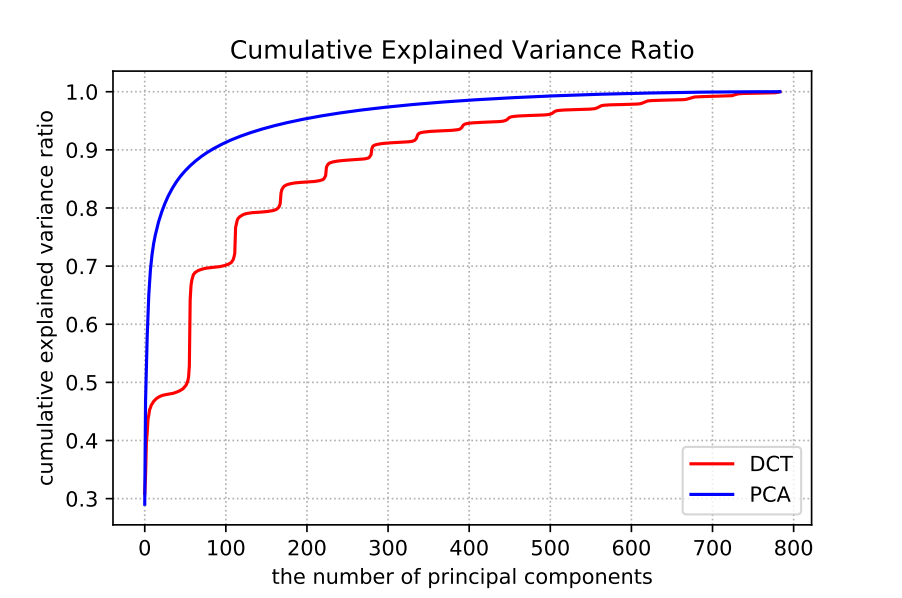


They are very different from the principal components we get in Question 1.5. As the value of k increases, the elements of the image become richer. It means that the vector c_k contains more information.

Question 1 continued on next page...

1.10 (4 points) Apply DCT to `Xtrn_nm` and calculate the variance, $\text{Var}(z_k^{(DCT)})$, for $k = 1, \ldots, D$.

The figure displays the cumulative explained variance ratio for both DCT and PCA.



Question 1 continued on next page...

1.11 (3 points) Based on the comparison between DCT and PCA above, discuss their advantages and disadvantages in terms of data compression.

DCT-advantage: Low computational complexity.
DCT-disadvantage: It loses a lot of features. The principal components are not in descending order.

PCA-advantage: It can represent more information with fewer dimensions. The most important component can be found quickly.
PCA-disadvantage: A large amount of calculation and the data needs to be preprocessed.

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.

The classification accuracy: 0.840
The confusion matrix is shown in the table below.

True \ Predict	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

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.

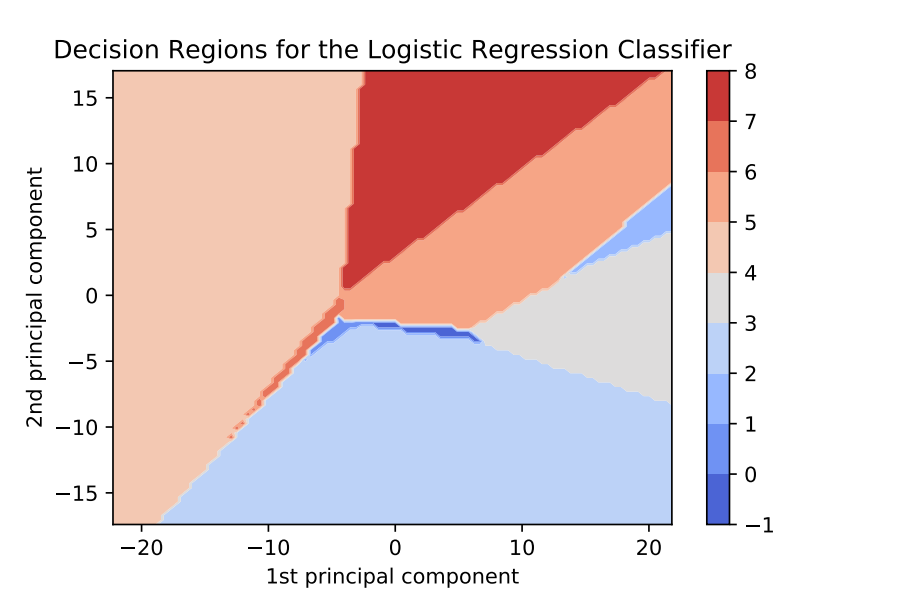
The mean accuracy: 0.846
The confusion matrix is shown in the table below.

True \ Predict	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

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 figure displays the result.

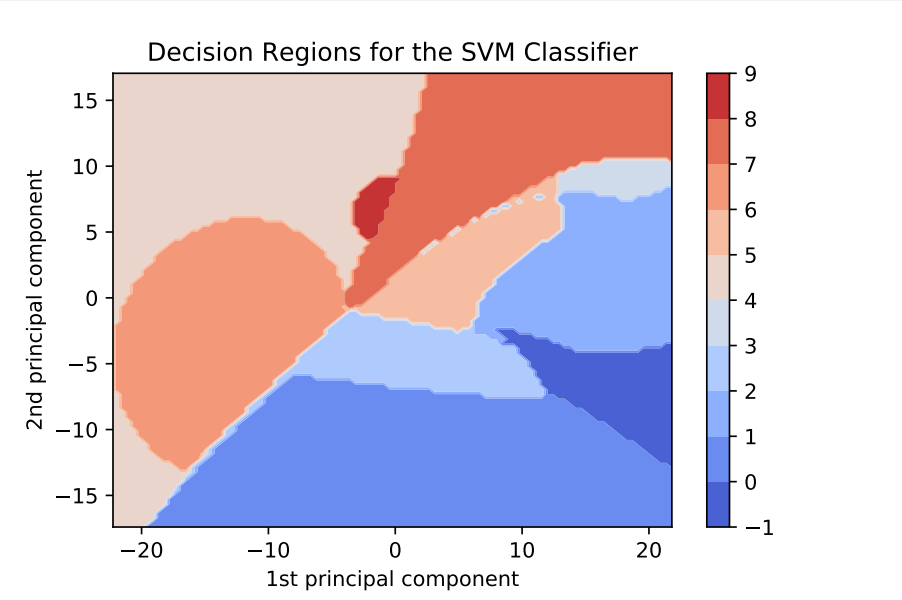


In the two-dimensional plane, the decision regions for the logistic regression classifier is linear. There are nine classes in this area. Different classes have different proportions. There are four classes with very small proportions.

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 figure displays the result.

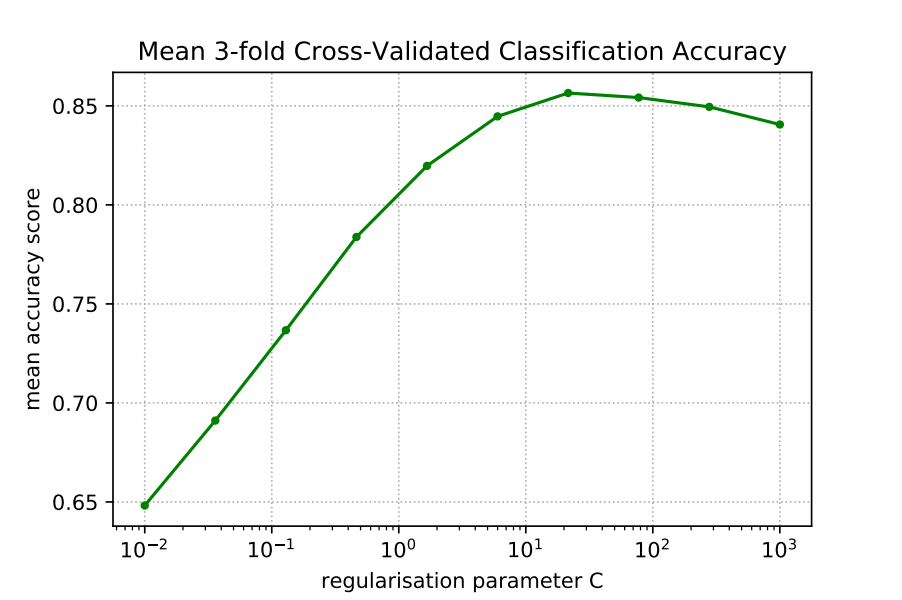


Compare with the result in Question 2.3, the decision regions for the SVM classifier is non-linear. All the classes appear in this area, ten in all. The proportion of each class is more balanced.

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 figure displays the mean cross-validated classification accuracy against the regularisation parameter C by using a log-scale for the x-axis.



When the value of C is $10^{1.333}(21.544)$, we obtained the highest mean accuracy score of 0.857.

Question 2 continued on next page...

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.

The table shows the result.

	training set	test set
classification accuracy	0.908	0.877

Question 3 : (32 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.

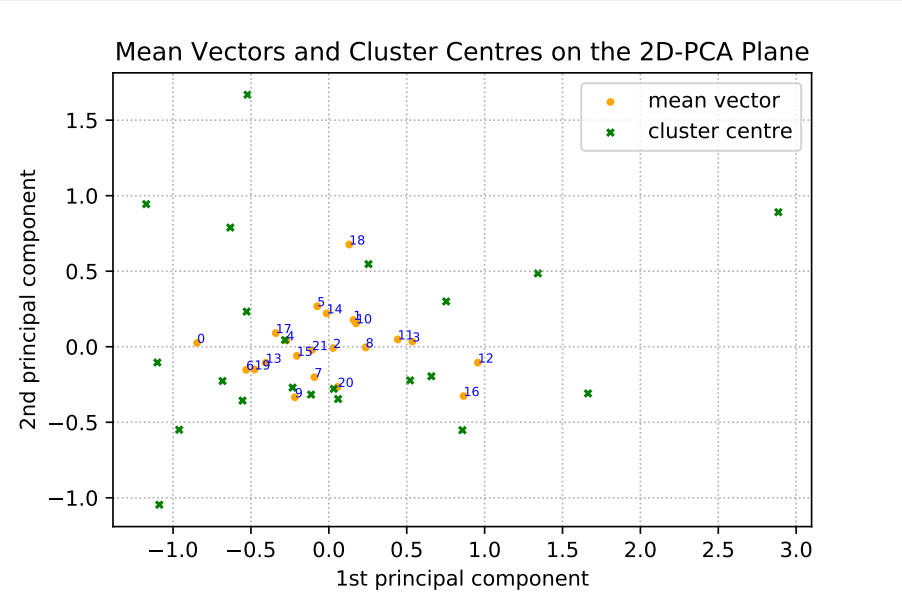
The sum of squared distances of samples to their closest cluster centre: 38185.817
The table shows the number of samples for each cluster.

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

Question 3 continued on next page...

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 figure displays the result.

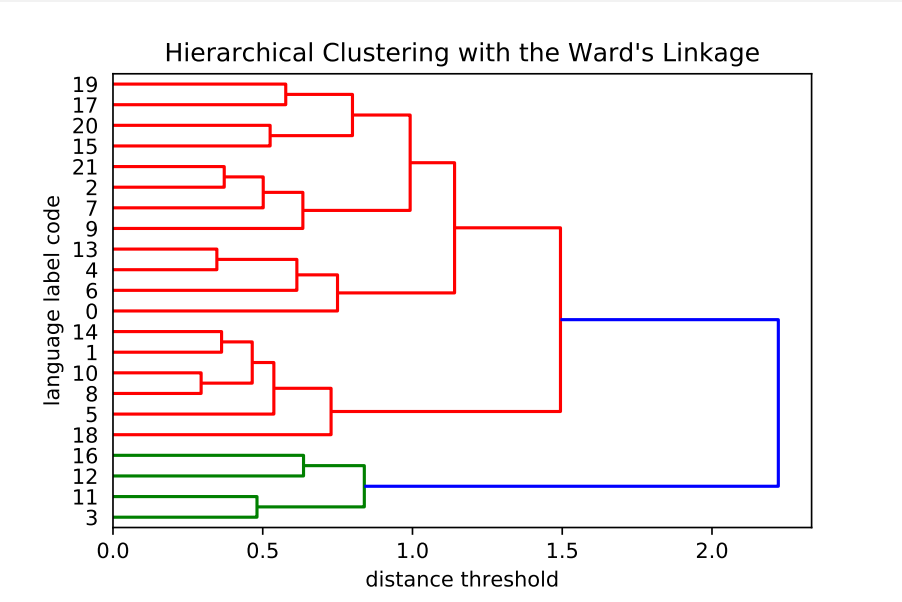


The mean vectors and cluster centres are not similar. The distribution of the mean vectors on the 2D-PCA plane is relatively concentrated. The divisions between classes are not obvious. The result of the k-means clustering is scattered, with large differences between classes.

Question 3 continued on next page...

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.

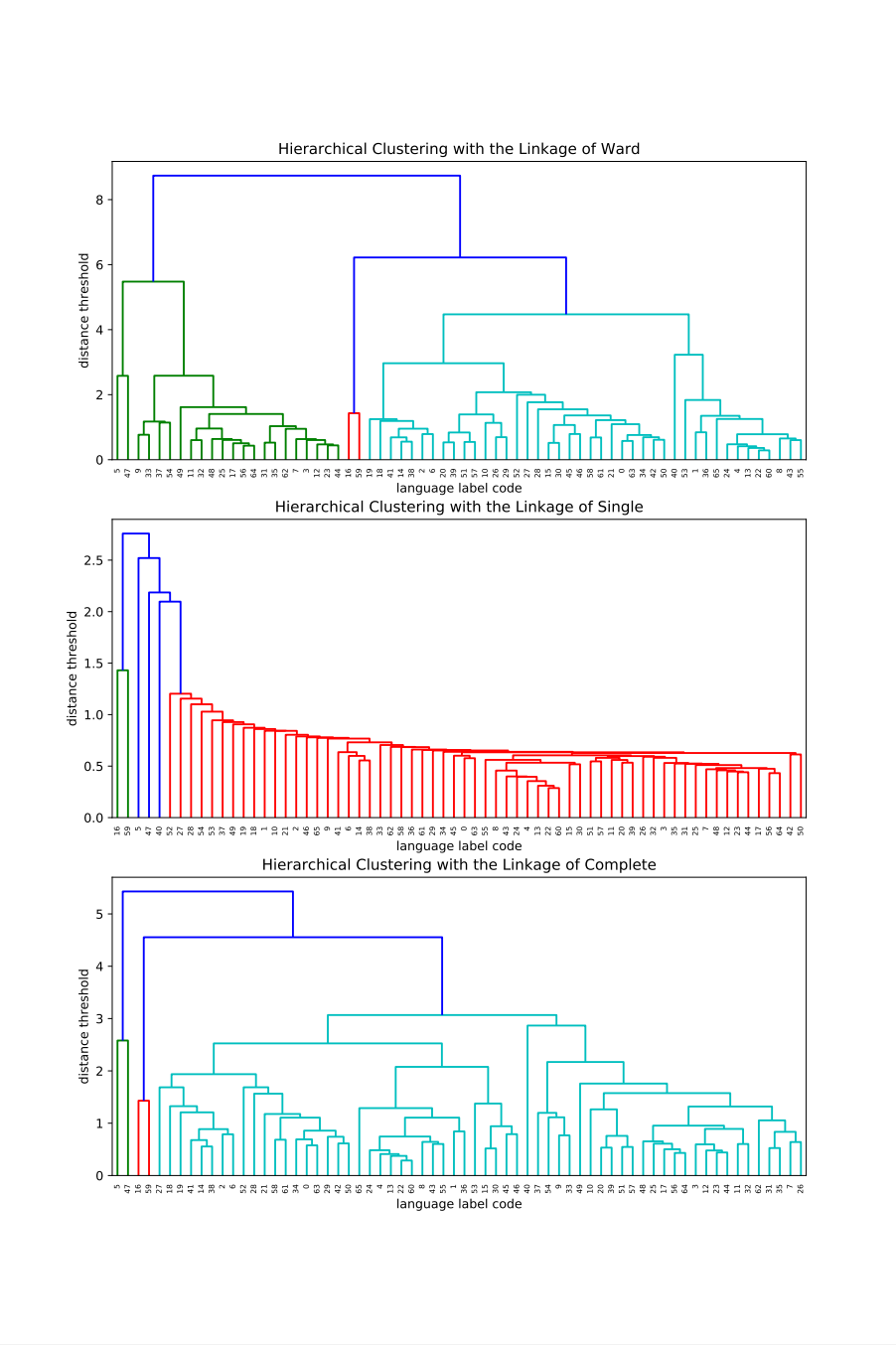
The figure displays the result.



In the initial division into two classes, language 3, 11, 12, and 16 are similar. From right to left, selecting different distance thresholds in the figure will divide the data set into different numbers of classes. And the classes increase sequentially. Finally, the entire data set is divided into 22 classes.

Question 3 continued on next page...

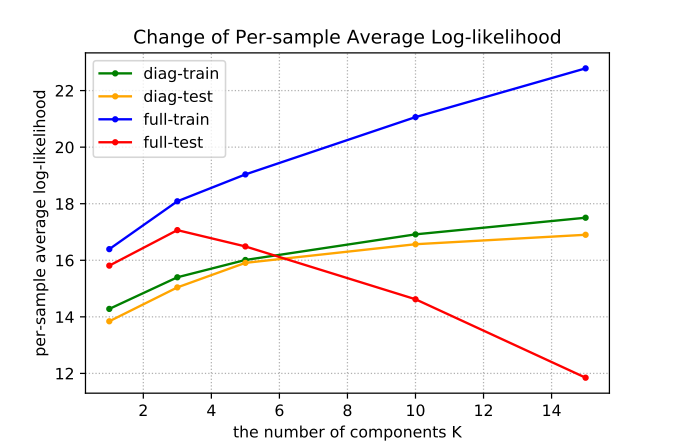
3.4 (5 points) We here extend the hierarchical clustering done in Question 3.3 by using multiple samples from each language.



Different linkage methods are based on different distance algorithms to measure the similarity of two subcategories. Therefore, their distance threshold ranges are different. The final clustering results are also completely different.

Question 3 continued on next page...

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.,



K	diag-train	diag-test	full-train	full-test
1	14.280	13.843	16.394	15.811
3	15.398	15.041	18.086	17.066
5	16.010	15.909	19.036	16.489
10	16.917	16.568	21.062	14.622
15	17.505	16.902	22.786	11.848

When using 'full covariance', with the K increases, the per-sample average log-likelihood on training data increases while it decreases on test data. Models may be overfitting. When using 'diag covariance', the values on training data and test data are almost the same. The generalization performances of models are better.

Question 3 continued on next page...

3.6 (5 points) The training of GMM employs the Expectation-Maximisation (EM) algorithm, which incrementally optimises model parameters - mean vectors, covariance matrices, and component weights,

```
def SimpleGMM_train(X, n_components):
    kmeans = KMeans(n_clusters=n_components, random_state=1).fit(X)
    Ms = kmeans.cluster_centers_

    Dcovs = []
    Pk = []

    for label in range(n_components):
        Dcovs.append(np.diagonal(np.cov(X[kmeans.labels_==label].T)))
        Pk.append(np.sum(kmeans.labels_==label))

    return Ms, np.array(Dcovs), np.array(Pk/np.sum(Pk))


def SimpleGMM_eval(X, Ms, Dcovs, Pk):
    likelihood = []

    for x in X:
        res = 0
        for i in range(len(Ms)):
            mn = multivariate_normal.pdf(x, mean=Ms[i], cov=Dcovs[i])
            res += np.dot(mn, Pk[i])
        likelihood.append(np.log(res))

    return np.array(likelihood)
```

Question 3 continued on next page...

3.7 (3 points) Now train the SimpleGMM on the training set for Language 0, and report the weights of mixture components for each $K = 1, 5, 10$.

The table shows the result.

K	the weights of mixture components (from 1 to K)									
1	1.000									
5	0.151	0.287	0.249	0.181	0.133					
10	0.090	0.113	0.065	0.100	0.143	0.071	0.102	0.071	0.114	0.132

Question 3 continued on next page...

3.8 (4 points) Using the data for Language 0 and using the SimpleGMM you obtained in Question 3.7, report, in a table, per-sample average log-likelihood on the training data and test data for $K = 1, 3, 5, 10, 15$.

The table shows the result.

K	training data	test data
1	14.280	13.843
3	15.223	14.967
5	15.798	15.602
10	16.619	16.438
15	17.128	16.551

When K=1, per-sample average log-likelihood on the training data and test data of SimpleGMM are the same as the results obtained in Question 3.5. When K=3,5,10,15, these values decrease slightly on both training data and test data. Using SimpleGMM with diagonal covariance can obtain almost the same per-sample average log-likelihoods as GMM.