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,:]`.

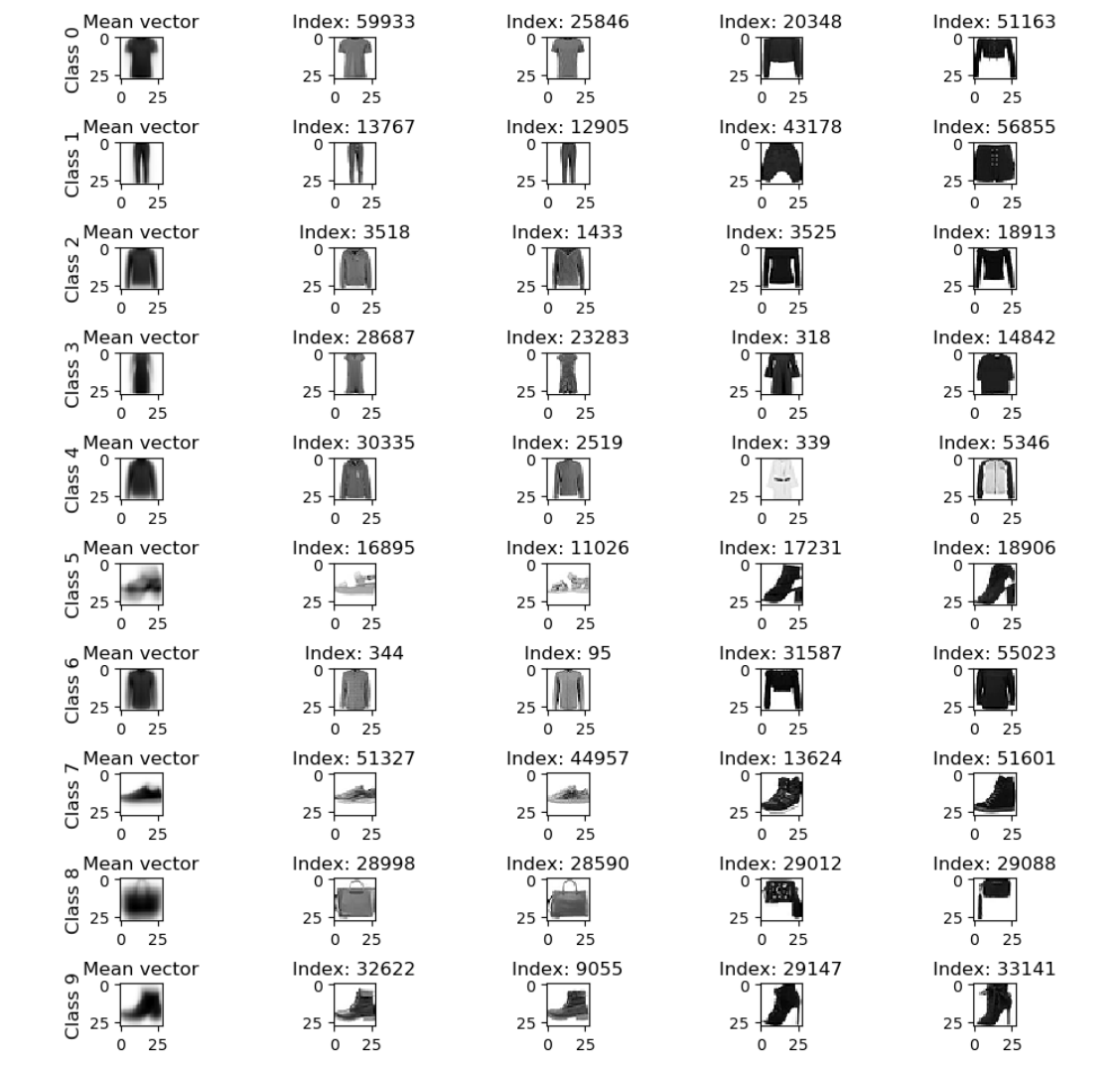
The values of the first 4 elements for the first training sample in Xtrn_nm are:

$$\mathbf{Xtrn\_nm}[\mathbf{0}, : \mathbf{4}] = \begin{bmatrix}\text{-3.00e-06} & \text{-2.30e-05} & \text{-1.18e-04} & \text{-4.07e-04}\end{bmatrix}$$

The values of the first 4 elements for the last training sample in Xtrn_nm are:

$$\mathbf{Xtrn\_nm}[-\mathbf{1}, : \mathbf{4}] = \begin{bmatrix}\text{-3.00e-06} & \text{-2.30e-05} & \text{-1.18e-04} & \text{-4.07e-04}\end{bmatrix}$$

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.



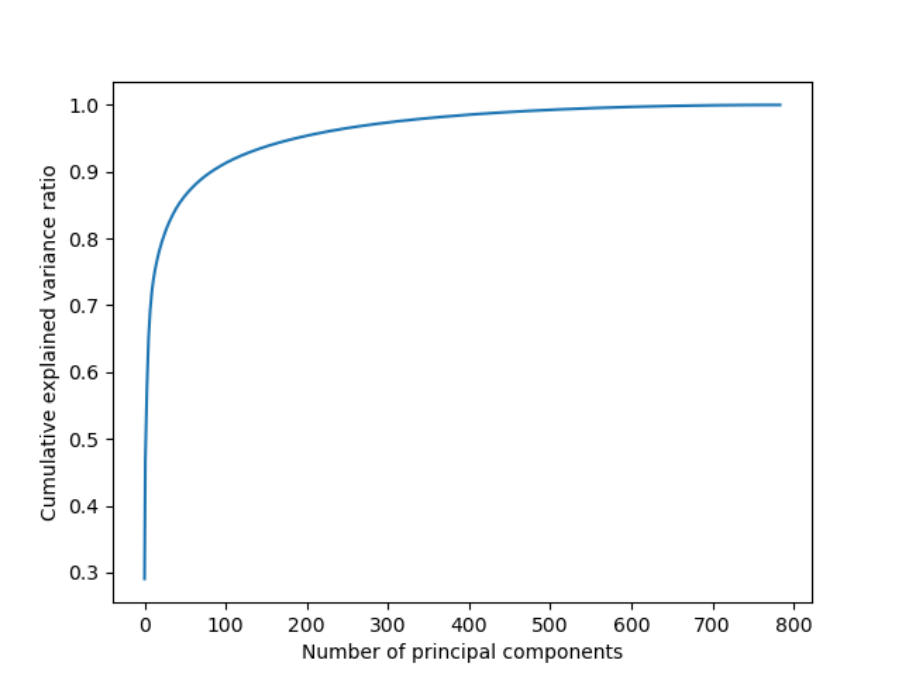
There can be observed that the 2 closest samples indeed look very similar to the sample projected by the mean vector, while the furthest samples are a completely different-looking element of the same class.

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`.

	Variance
PC 1	19.8098
PC 2	12.1122
PC 3	4.1062
PC 4	3.3818
PC 5	2.6248

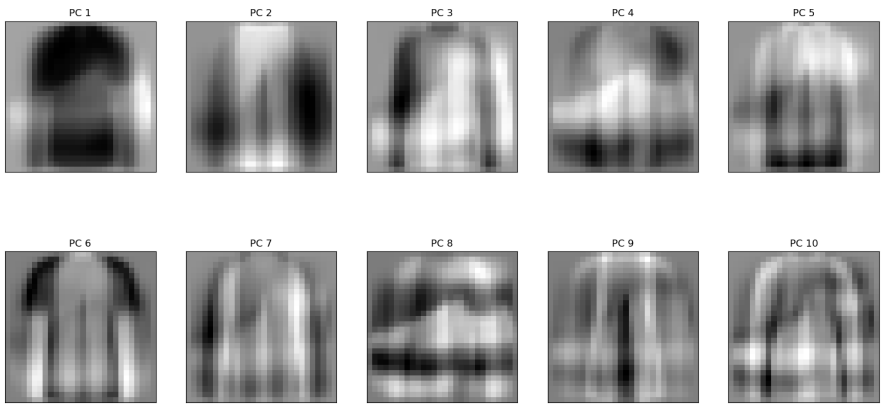
Question 1 continued on next page. .Alexandru-Ioan Chelba

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.



It can be observed that the graph converges to one, and if we are to capture above 90% of information, we need less than 100 principal components.

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.



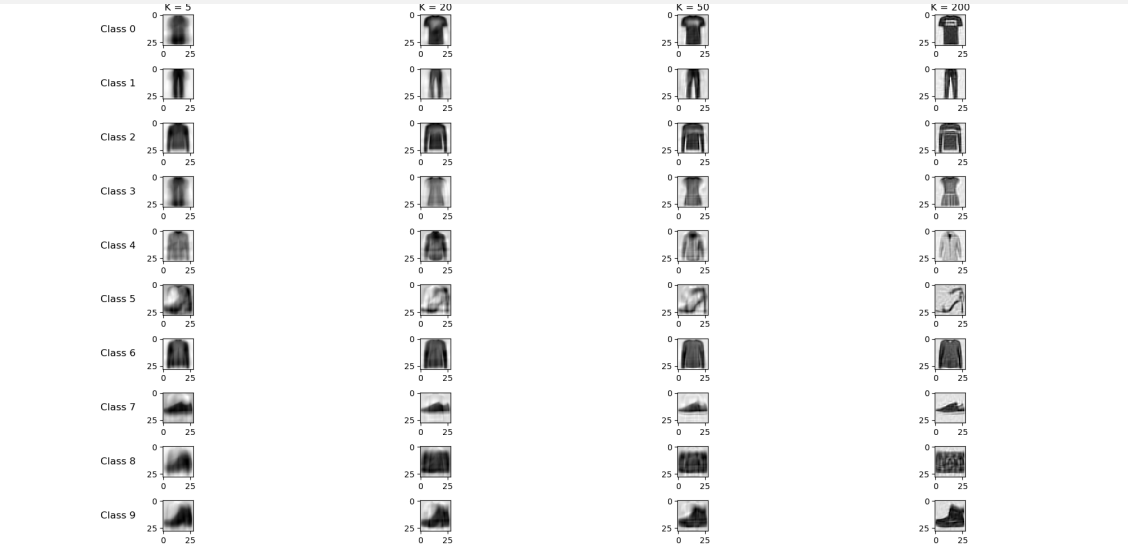
It can be observed that the first 10 principal components mainly deal with lighting aspects in different areas of the image. Interesting to note is the shoe form that appears in every image, which indicates the presence of shoes' class in the training set.

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.

	K = 5	K = 20	K = 50	K = 200
Class 0	0.2561	0.1501	0.1276	0.0612
Class 1	0.198	0.1405	0.0955	0.0346
Class 2	0.1987	0.1456	0.1218	0.0792
Class 3	0.1457	0.1072	0.0833	0.0564
Class 4	0.1182	0.1026	0.0878	0.0466
Class 5	0.1811	0.1587	0.143	0.0898
Class 6	0.1295	0.0952	0.0718	0.0454
Class 7	0.1656	0.128	0.1065	0.0643
Class 8	0.2234	0.1451	0.1236	0.0915
Class 9	0.1835	0.1512	0.1221	0.07

Question 1 continued on next page. .Alexandru-Ioan Chelba

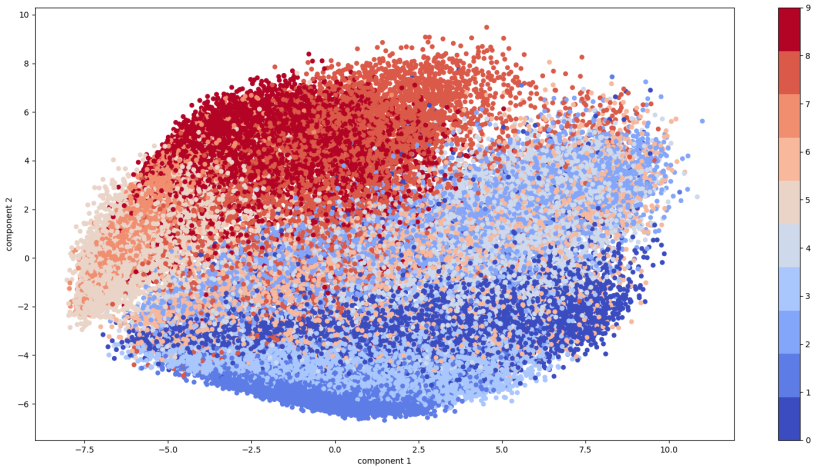
1.7 (4 points) Display the image for each of the reconstructed samples in a 10-by-4 grid, where each row corresponds to a class and each row column corresponds to a value of $K = 5,\ 20,\ 50,\ 200$.



The images reveal that reconstruction of each class's element gets better and better as the number of principal components increases.

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 plot shows that classes are hardly distinguishable based only on 2 principal components, hence there is hardly any separation between them on the plot.

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 reported accuracy score is: 0.8401.

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

Question 2 continued on next page. .Alexandru-Ioan Chelba

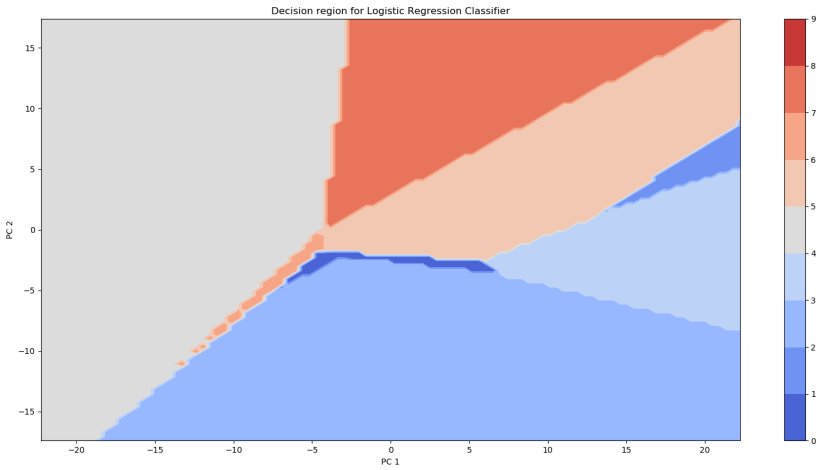
2.2 (3 points) Carry out a classification experiment with SVM classifiers, and report the mean accuracy and confusion matrix (in numbers) for the test set.

The reported accuracy score = 0.8461.

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

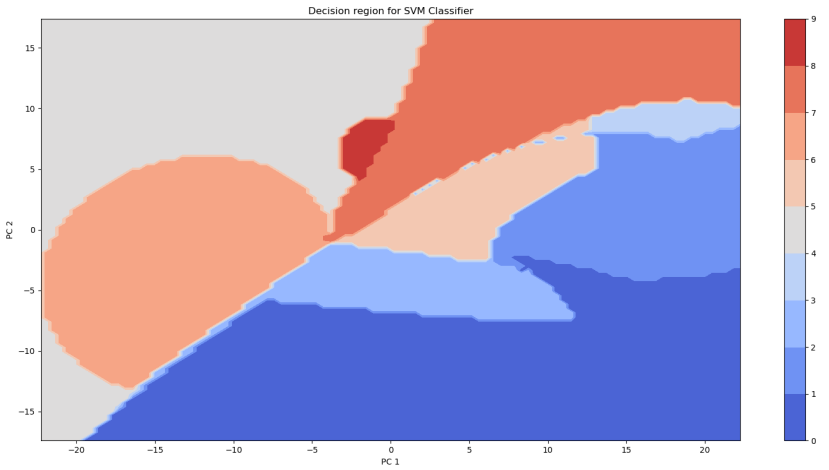
Question 2 continued on next page. .Alexandru-Ioan Chelba

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



It can be observed that class 9 is missed out completely, while classes 0,1,6 have small decision regions. There is a clear gear towards classes 2,3,4,5,7,8.

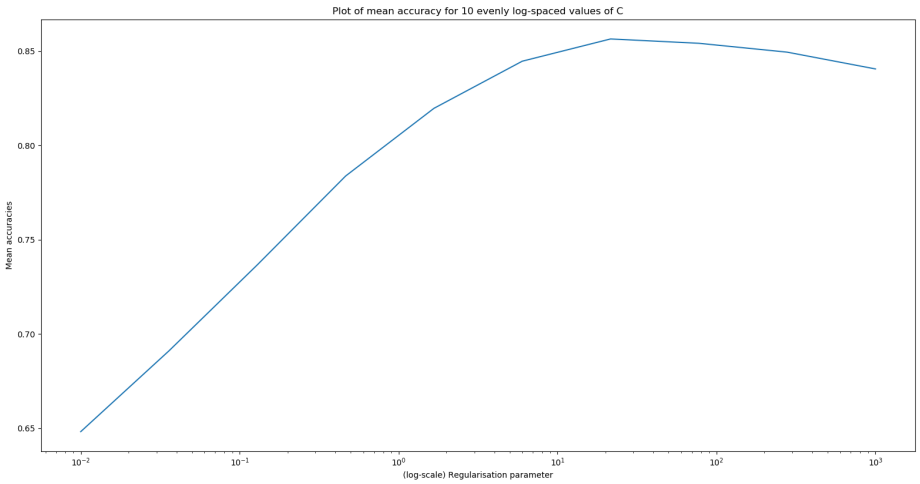
2.4 (4 points) Using the same method as the one above, plot the decision regions for the SVM classifier you trained in Question 2.2. Comparing the result with that you obtained in Question 2.3, discuss your findings briefly.



The biggest difference between the 2 models is that the SVM classifier recognises a region for class 9, whereas Logistic Regression misses it completely.
There are also visible differences in where a decision region, for the same class, lies on both plots: for example, class 0 has a much larger decision region on the plot for the SVM classifier, than it does on the plot for the Logistic Regression classifier.

Question 2 continued on next page. .Alexandru-Ioan Chelba Page 12 of 19

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 highest obtained mean accuracy score is: 0.8565, and the value of C which yielded it, is: 21.5443.

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 training set classification accuracy is: 0.9084.
The test set classification accuracy 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.

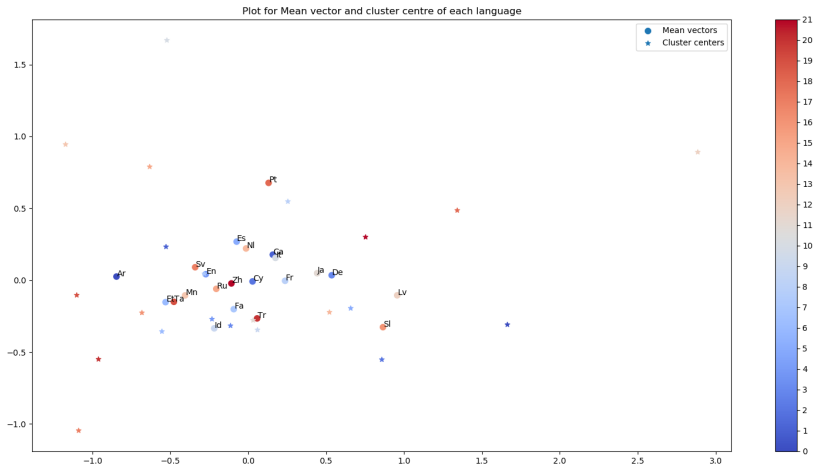
3.1 (3 points) Apply k-means clustering on `Xtrn` for $k = 22$, where we use sklearn.cluster.KMeans with the parameters `n_clusters=22` and `random_state=1`. Report the sum of squared distances of samples to their closest cluster centre, and the number of samples for each cluster.

The sum of squared distances of samples to their closest cluster centre is: 38185.817.
The number of samples per cluster is:

Cluster index	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

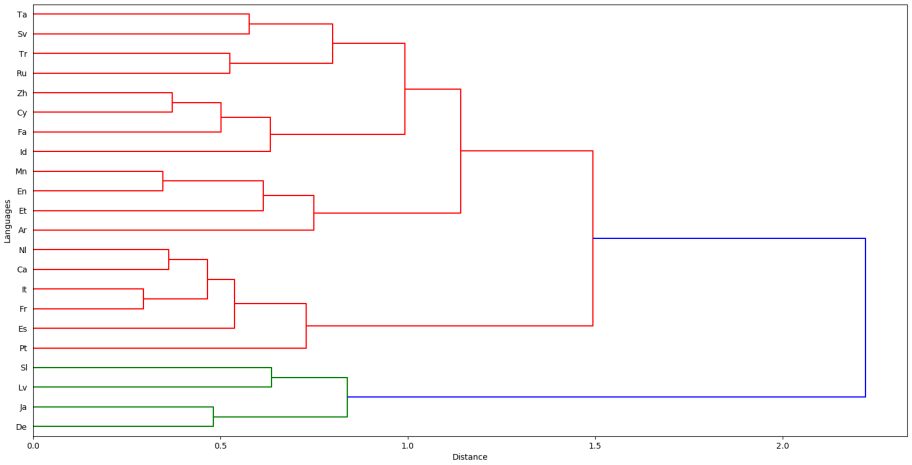
Question 3 continued on next page. .Alexandru-Ioan Chelba

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.



It can be observed that the mean vectors and the cluster centres are quite different. This could be due to a number of reasons, one of which could be outliers affecting the mean vectors. It is worth noting that we do not know what class each cluster centre is for, since it is an unsupervised method. They are colored on the map because I could not find a way to show the colorbar otherwise.

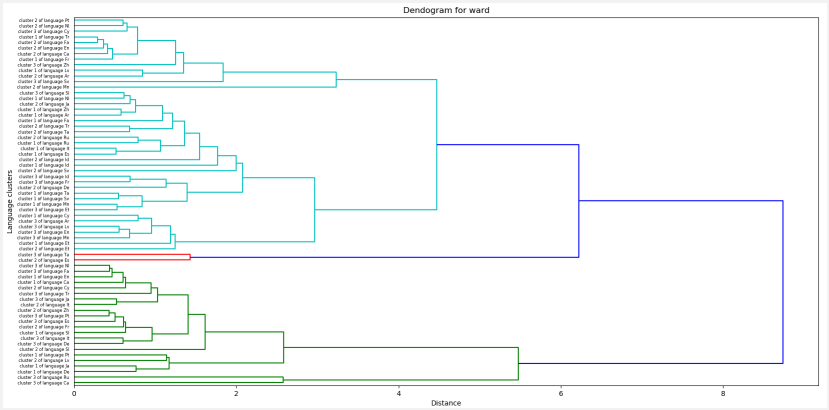
3.3 (3 points) We now apply hierarchical clustering on the training data set to see if there are any structures in the spoken languages.

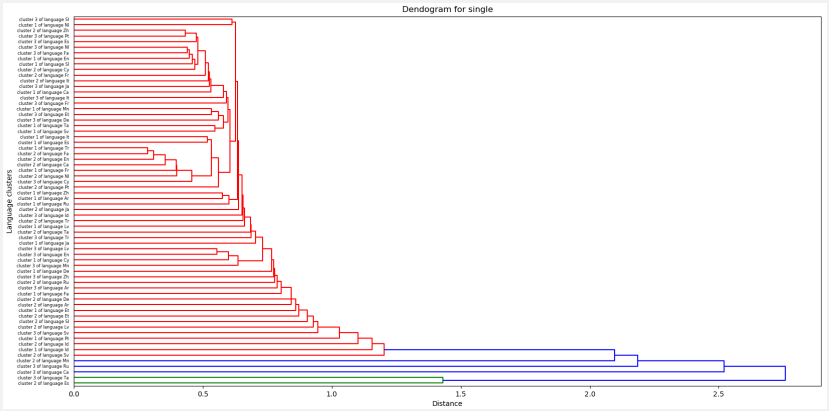


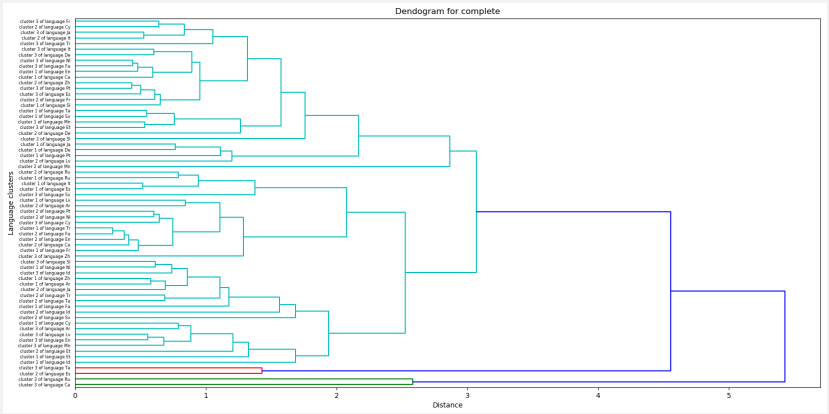
The dendogram suggests that, if we take only the vector means into account, languages are pretty similar between them, with maximum dissimilarity at just over 2. It can be seen that German (De), Slovenian (Sl), Japanese (Ja) and Latvian (Lv) join the rest of the group at latest, suggesting these 4 languages are more different from the others.

Question 3 continued on next page. .Alexandru-Ioan Chelba

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



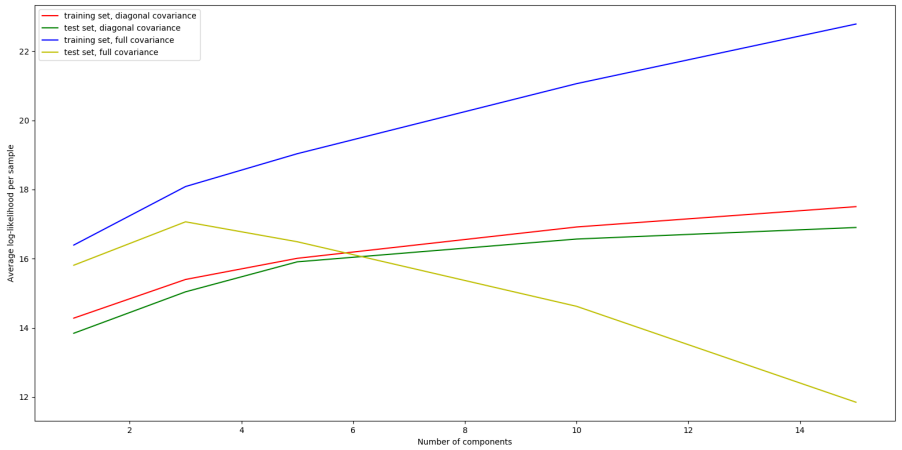




It can be observed that Ward linkage gives the highest dissimilarity score for all languages, followed by complete linkage and then the single linkage. It highlights how each method works: single linkage creates a long chain, while complete linkage and Ward linkage get a considerable share of objects clustered even on early steps and then proceed to merge those.

Question 3 continued on next page. .Alexandru-Ioan Chelba Page 18 of 19

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.,



	1 components	3 components	5 components	10 components	15 components
Diagonal covariance	(train, test) = (14.28,13.843)	(train, test) = (15.398,15.041)	(train, test) = (16.01,15.909)	(train, test) = (16.917,16.568)	(train, test) = (17.505,16.902)
Full covariance	(train, test) = (16.394,15.811)	(train, test) = (18.086,17.066)	(train, test) = (19.036,16.489)	(train, test) = (21.062,14.622)	(train, test) = (22.786,11.848)

It can be seen from the graph, that as the number of components increases, the GMM tends to overfit the training data. This is especially obvious for full covariance matrix, where the average log-likelihood per test sample goes down steeply as the number of components increase, while the average log-likelihood per training sample grows with the number of components. A slight such tendency can be observed for diagonal covariance can be observed as well, although it is more subtle.