Principal Component Analysis on Bags of Visual Words data

Group 5: Chirag Vashist, Prabhakar Prasad, Sujetth Rangannath

November 15, 2018

Contents

1 Aim

The aim is to classify images given in Dataset 2(b) of assignment 2 after first applying PCA for dimension reduction.

2 Theory

Principal Component Analysis (also called PCA) is a technique that is used to reduce the number of dimensions in the given data. Let us look at how it works:

1. In our case we have 32 dimensional BoVW data of images which was extracted for Assignment 2.

2. Now we aim to reduce the number of dimensions in the data from 32 to lets say L.

3. In order to do so first we build the co-variance matrix for the 32 dimensional data. After this we compute all of its eigenvalues and their corresponding eigenvectors.

4. Then we sort all the eigenvectors on the basis of their eigenvalues in descending order.

5. From this we take the top L eigenvectors. For each data point, we take its dot product with all these L eigenvectors. These L dot products constitute our new L dimensional data point.

6. With this we have converted out 32 dimensional data to L dimensional data.

The reason the name ”Principal Component Analysis” is justified is because in essence, the data we are losing is data that matters least and the data we are keeping is data which matters most. This point is obvious when one looks at the physical significance of eigenvalues and eigenvectors. Eigenvectors basically give the direction of spread of a matrix while eigenvalues give the intensity of spread in the direction of its corresponding eigenvector. So by selecting the top L eigenvectors with highest eigenvalues we are in essence selecting the part of the data that matters most (the principal components) and eliminating the rest. One obvious advantage of using PCA is that it helps reduce the computational time since we are now dealing with lesser number of dimensions. Another effect is the loss of some data. While this isn't a good thing it might not necessarily be a bad thing if the data that is being thrown away was actually unnecessary noise. So in the end it depends on the data and on the values of L. Reducing the dimension by a huge factor might be a bad idea since important information might be lost this way.

2.1 Case 2: 32-Dimensional Bag of Visual Words

2.1.1 Data Preparation

We prepared BoVW for each image from set of feature vectors extracted for each image in previous step. For that, we first combine all training data (i.e. all features vectors of all training images) and group them in 32 clusters by applying k-means clustering. After that, for each image in training and test images, we assign it's feature vectors to those 32 clusters and count the number of feature vectors assigned to each of the 32 clusters. This results in 32-dimensional BoVW representation of that image.

Image types and their corresponding class label is mentioned below.
Class 1 - Cottage_Garden
Class 2 - Firing_Range_Indoor
Class 3 - Pharmacy

2.1.2 Results using all 32 Dimensions

This result was obtained previously using all 32 dimension from the data and building a GMM-based Classifier using the data.

Confusion Matrix for k as 1= $\begin{bmatrix} 29 & 20 & 1 \\ 0 & 43 & 7 \\ 1 & 33 & 16 \end{bmatrix}$

Results for k=1				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.966	0.58	0.725	-
Firing_Range_Indoor	0.447	0.86	0.589	-
Pharmacy	0.666	0.32	0.432	-
Mean	0.693	0.586	0.582	0.558

Confusion Matrix for k as 2= $\begin{bmatrix} 31 & 2 & 17 \\ 3 & 15 & 32 \\ 2 & 9 & 39 \end{bmatrix}$

Results for k=2				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.861	0.62	0.720	-
Firing_Range_Indoor	0.576	0.3	0.394	-
Pharmacy	0.443	0.78	0.565	-
Mean	0.627	0.566	0.560	0.566

Confusion Matrix for k as 4= $\begin{bmatrix} 43 & 2 & 5 \\ 4 & 27 & 19 \\ 7 & 8 & 35 \end{bmatrix}$

Results for k=4				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.796	0.86	0.826	-
Firing_Range_Indoor	0.729	0.54	0.620	-
Pharmacy	0.593	0.7	0.642	-
Mean	0.706	0.7	0.696	0.7

$$\text{Confusion Matrix for k as 8=}\begin{bmatrix} 37 & 6 & 7 \\ 5 & 34 & 11 \\ 2 & 15 & 33 \end{bmatrix}$$

Results for k=8				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.840	0.74	0.787	-
Firing_Range_Indoor	0.618	0.68	0.647	-
Pharmacy	0.647	0.66	0.653	-
Mean	0.702	0.693	0.696	0.693

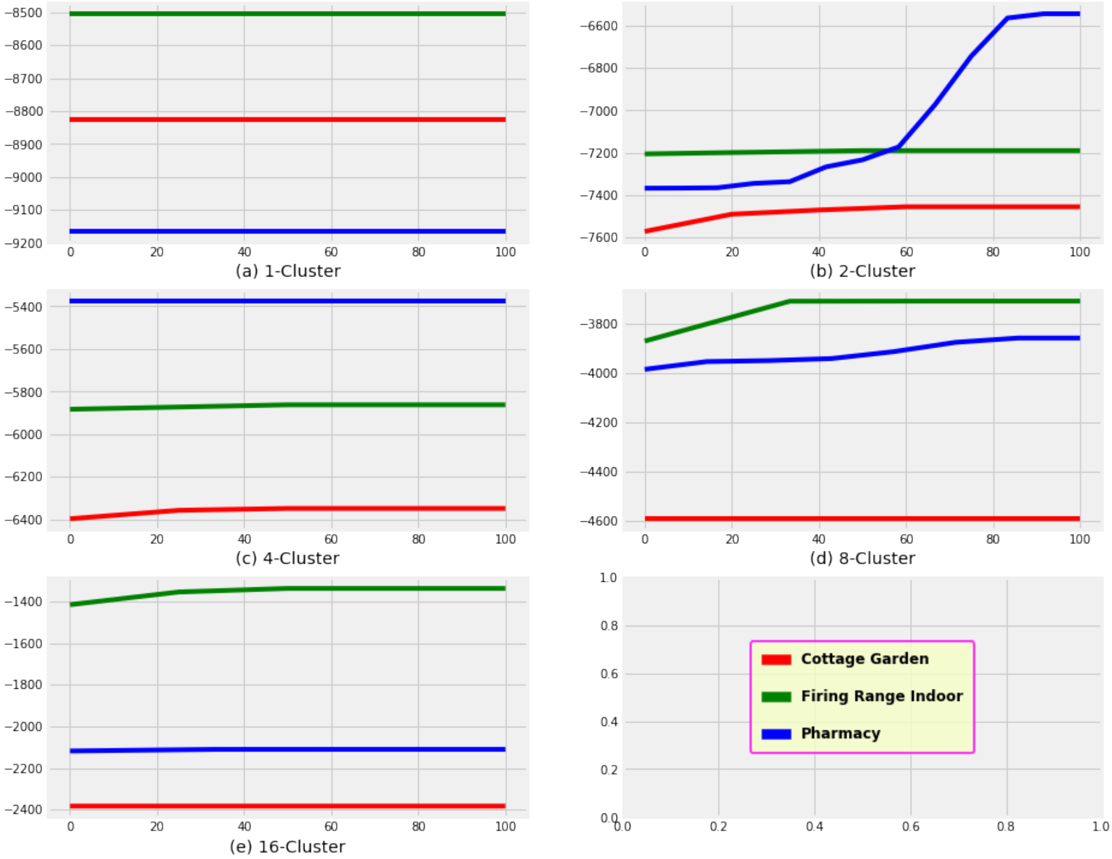


Figure 1: Log Likelihood for various clusters : Bag of Visual Words

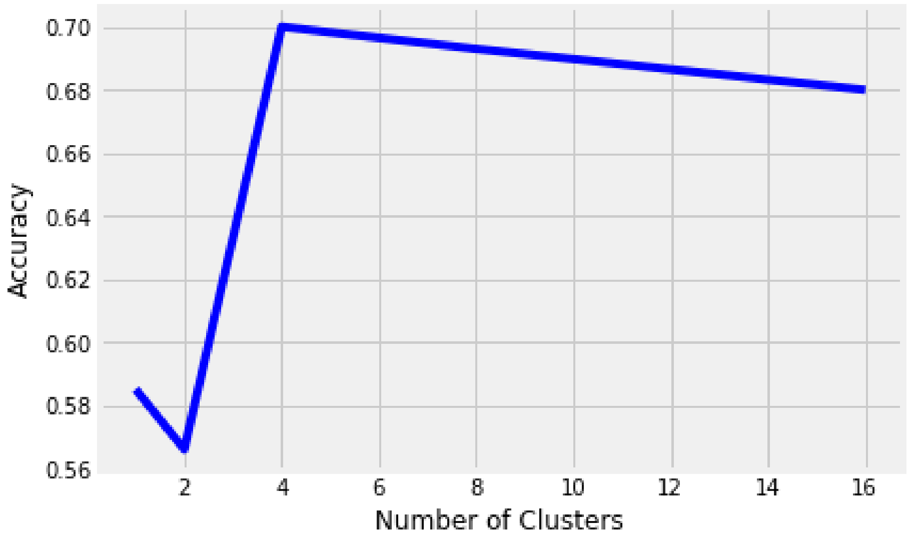


Figure 2: Accuracy vs Number of Cluster : Bag of Visual Words

2.1.3 Results using 15 Dimensions

Confusion Matrix for k as 1= $\begin{bmatrix} 26 & 24 & 0 \\ 2 & 41 & 7 \\ 4 & 37 & 9 \end{bmatrix}$

Results for k=1				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.81	0.52	0.634	-
Firing_Range_Indoor	0.401	0.82	0.539	-
Pharmacy	0.562	0.18	0.272	-
Mean	0.506	0.506	0.482	0.506

Confusion Matrix for k as 2= $\begin{bmatrix} 27 & 7 & 16 \\ 7 & 22 & 21 \\ 11 & 8 & 31 \end{bmatrix}$

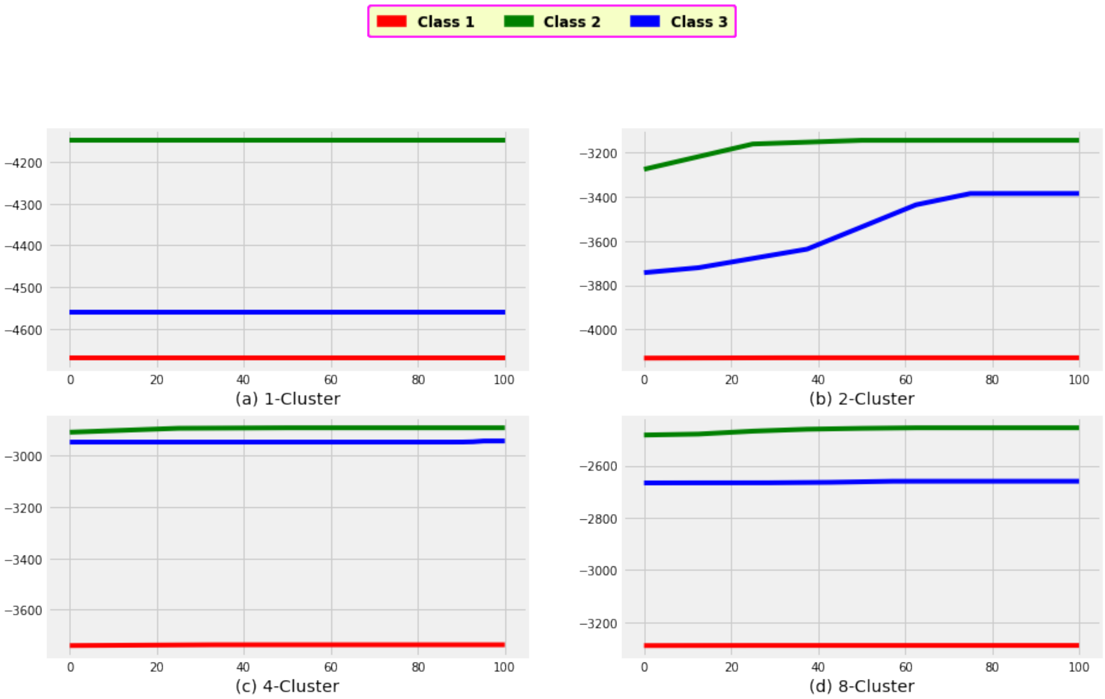
Results for k=2				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.6	0.54	0.568	-
Firing_Range_Indoor	0.594	0.44	0.505	-
Pharmacy	0.455	0.62	0.525	-
Mean	0.550	0.533	0.533	0.533

Confusion Matrix for k as 4= $\begin{bmatrix} 29 & 5 & 16 \\ 1 & 27 & 22 \\ 3 & 8 & 39 \end{bmatrix}$

Results for k=4				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.878	0.58	0.698	-
Firing_Range_Indoor	0.675	0.54	0.600	-
Pharmacy	0.506	0.78	0.614	-
Mean	0.686	0.633	0.637	0.633

$$\text{Confusion Matrix for k as 8=}\begin{bmatrix} 36 & 8 & 6 \\ 4 & 31 & 15 \\ 7 & 8 & 35 \end{bmatrix}$$

Results for k=8				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.765	0.72	0.742	-
Firing_Range_Indoor	0.659	0.62	0.639	-
Pharmacy	0.647	0.67	0.66	-
Mean	0.683	0.68	0.68	0.68



(a) 1-Cluster

(b) 2-Cluster

(c) 4-Cluster

(d) 8-Cluster

Figure 3: Log Likelihood for various clusters : 15 Principal Components

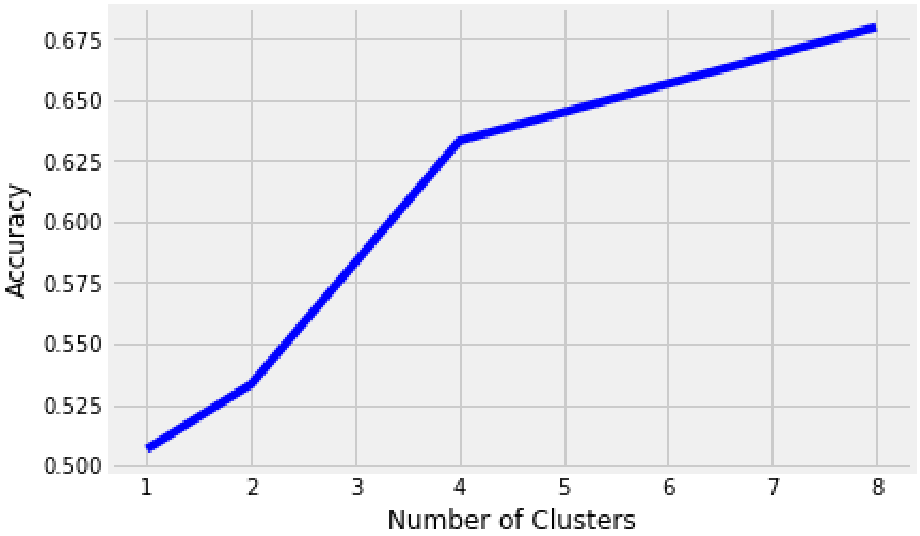


Figure 4: Accuracy vs Number of Cluster : 15 Principal Components

2.1.4 Results using 10 Dimensions

Confusion Matrix for k as 1= $\begin{bmatrix} 23 & 25 & 2 \\ 3 & 41 & 6 \\ 4 & 38 & 8 \end{bmatrix}$

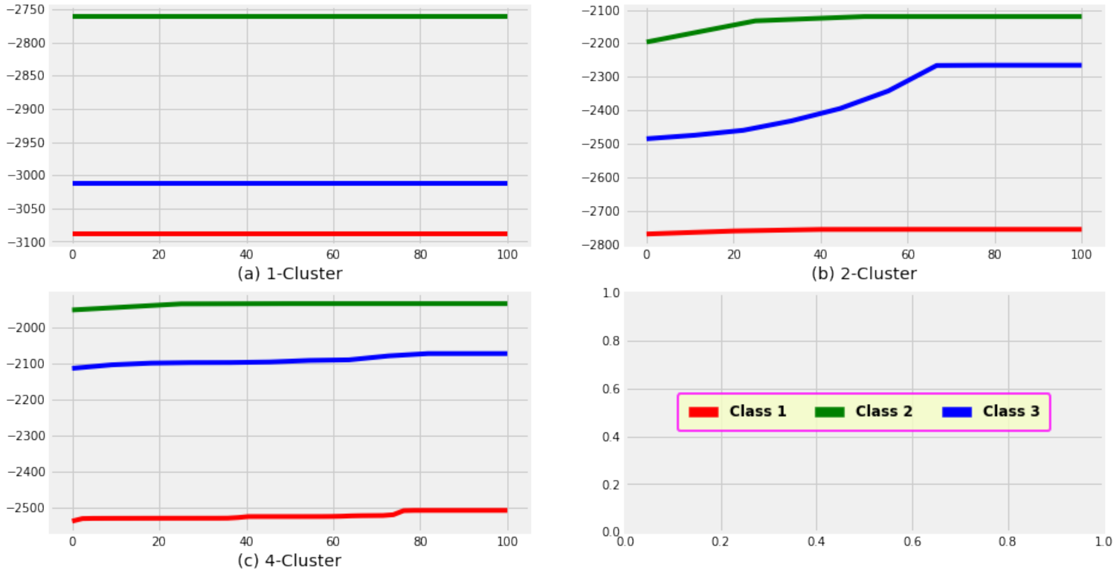
Results for k=1				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.766	0.46	0.575	-
Firing_Range_Indoor	0.394	0.82	0.532	-
Pharmacy	0.5	0.16	0.242	-
Mean	0.553	0.48	0.449	0.48

Confusion Matrix for k as 2= $\begin{bmatrix} 28 & 6 & 16 \\ 6 & 21 & 23 \\ 10 & 10 & 30 \end{bmatrix}$

Results for k=2				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.636	0.56	0.595	-
Firing_Range_Indoor	0.567	0.42	0.482	-
Pharmacy	0.434	0.6	0.504	-
Mean	0.546	0.526	0.527	0.526

Confusion Matrix for k as 4= $\begin{bmatrix} 38 & 3 & 9 \\ 4 & 30 & 16 \\ 9 & 11 & 30 \end{bmatrix}$

Results for k=4				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.745	0.76	0.752	-
Firing_Range_Indoor	0.681	0.6	0.632	-
Pharmacy	0.681	0.6	0.671	-
Mean	0.653	0.653	0.654	0.653



(a) 1-Cluster

(b) 2-Cluster

(c) 4-Cluster

Figure 5: Log Likelihood for various clusters : 10 Principal Components

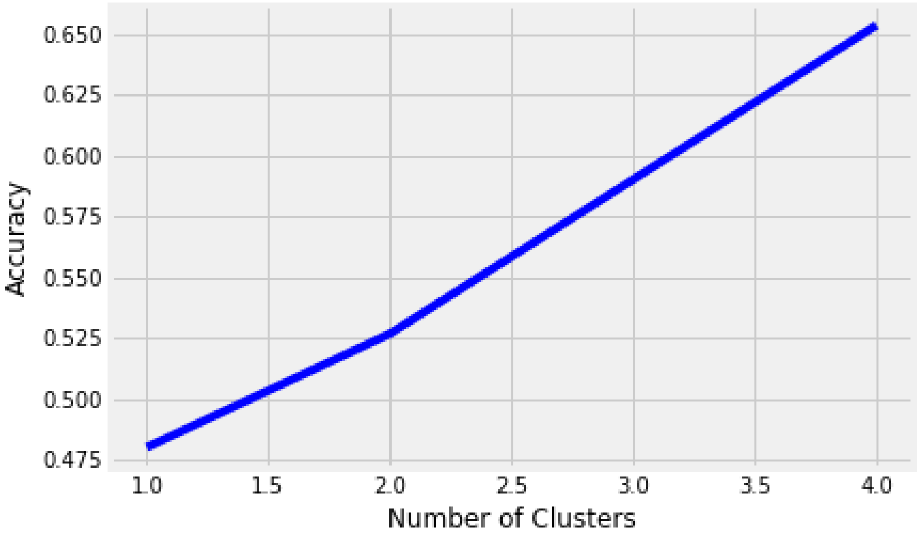


Figure 6: Accuracy vs Number of Cluster : 10 Principal Components

2.1.5 Results using 5 Dimensions

Confusion Matrix for k as 1= $\begin{bmatrix} 22 & 26 & 2 \\ 0 & 42 & 8 \\ 3 & 39 & 8 \end{bmatrix}$

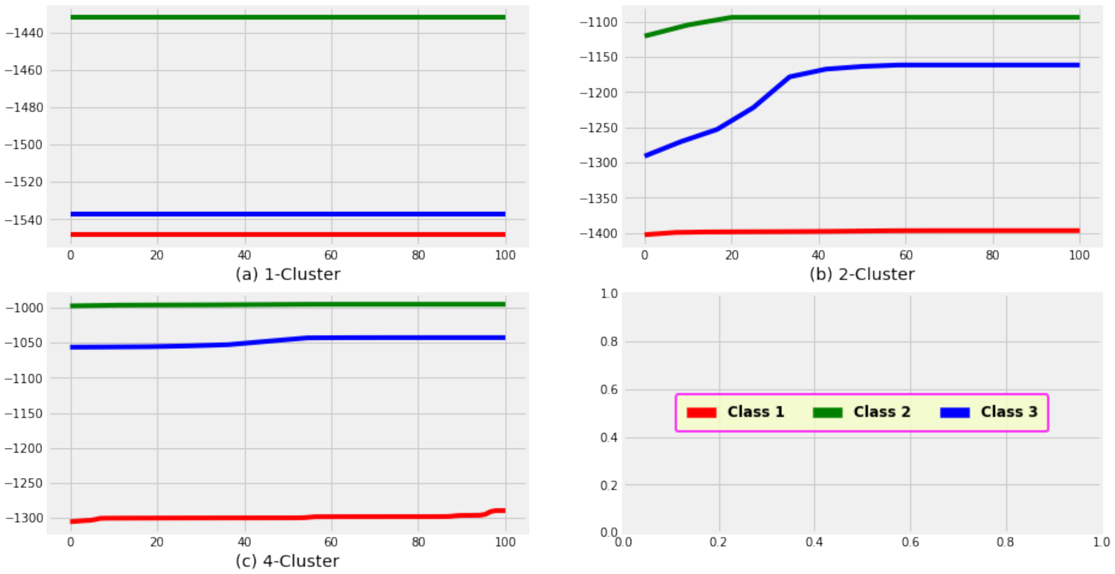
Results for k=1				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.88	0.44	0.586	-
Firing_Range_Indoor	0.392	0.84	0.535	-
Pharmacy	0.444	0.16	0.235	-
Mean	0.572	0.48	0.452	0.48

Confusion Matrix for k as 2= $\begin{bmatrix} 27 & 4 & 19 \\ 5 & 15 & 30 \\ 10 & 5 & 35 \end{bmatrix}$

Results for k=2				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.642	0.54	0.589	-
Firing_Range_Indoor	0.625	0.3	0.405	-
Pharmacy	0.416	0.7	0.522	-
Mean	0.561	0.513	0.504	0.513

Confusion Matrix for k as 4= $\begin{bmatrix} 32 & 4 & 14 \\ 3 & 28 & 19 \\ 8 & 11 & 31 \end{bmatrix}$

Results for k=4				
	Precision	Recall	F-Measure	Accuracy
Cottage_Garden	0.744	0.64	0.688	-
Firing_Range_Indoor	0.65	0.56	0.602	-
Pharmacy	0.484	0.62	0.543	-
Mean	0.626	0.606	0.611	0.606



(a) 1-Cluster

(b) 2-Cluster

(c) 4-Cluster

Figure 7: Log Likelihood for various clusters : 5 Principal Components

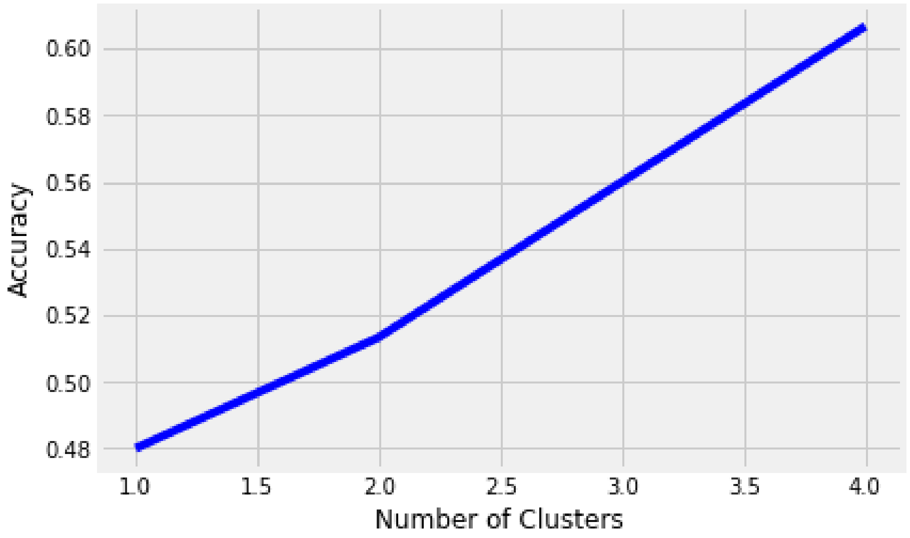


Figure 8: Accuracy vs Number of Cluster : 5 Principal Components

3 Conclusion

In this section, we will look at some important inferences, possible computational problem and their solutions.

3.1 Performance of PCA

The space-complexity of Gaussian Mixture model is directly dependent on square of number of dimensions.
From the above tables, we observe that using PCA, the dimension of the data can be reduced multi-fold without affecting the accuracy significantly. This is evident because 10 dimensional data-set has an accuracy of 65.3% for 8-cluster GMM Classifier, while the original 32-dimensional has an accuracy of 69.3% for the same classifier.
This means that we can safely reduce the dimensions of the data using PCA without the accuracy of the Classifier being affected very much.

3.2 Singular Covariance Matrix

When one of the components of the mixture model, has its mean exactly equal to one of the data points, we will get a very spiky Gaussian that ”collapses” to that point. This causes the coavariance matrix to be singular, leading to the probability being infinte. This causes computational problems, leading to NaN (Not a Number Error) in python. This problem can be solved by two ways :

1. Adding a very little value to the diagonal terms of the covariance matrix. In out case, we added 1.0e-6. This method is used by the Sklearn library.

2. Observing when the covariance matrix becomes singular and setting its mean and/or covariance matrix to a new, arbitrarily high value(s).

3.3 Optimum number of Principal Components

As we can see, the number of Principal Components directly influence the accuracy of the GMM model built upon it. Thus, it is necessary to find a way to calculate an optimal value for number of Principal Components. This is done using the Scree Test.
We plot the component number on the horizontal access, and the eigenvalues on the vertical axis. The rule is to simply pick the number of components when your slope starts leveling off. From the below graph, we can see that for this data set, any number of components between 5 and 10 would be suitable.

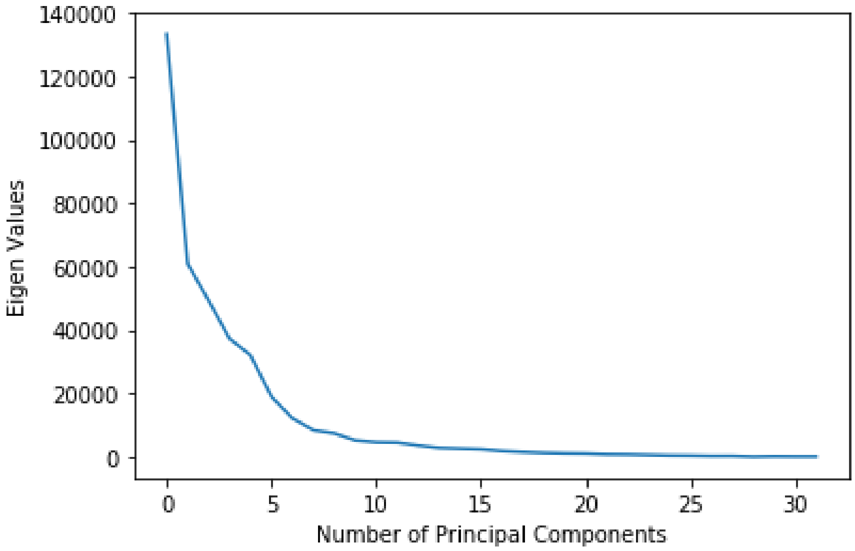


Figure 9: Scree Test for finding optimal number of Principal Components

Alternatively, we can plot Cumulative Variance graph. First, we find the variance explained by each component by dividing each component's eigenvalue by the sum of all eigenvalues. To find the cumulative variance explained by a component C3, you should add the variance explained by components (C1, C2, C3).
Ground rule is to choose number of components that amounts to 70-80% of the cumulative variance.

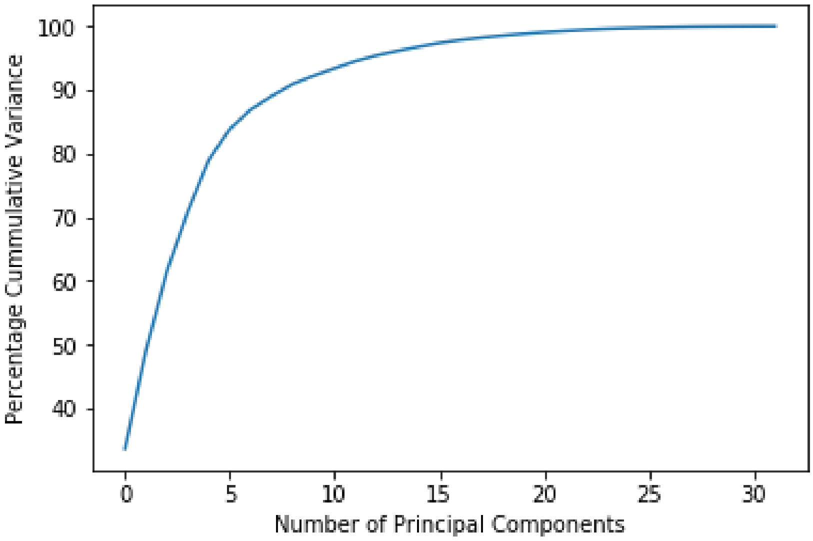


Figure 10: Scree Test for finding optimal number of Principal Components

3.4 Source of Wrong Classification

By observing the confusion matrix, we can observe that the major source of wrong classification are images of Class ”Firing Range Indoor” being classified as Class ”Pharmacy”.
On carefully observing which images are being classified incorrectly, we compared them with images of Class ”Pharmacy” and found that they are quite similar in terms of colour composition and structure.

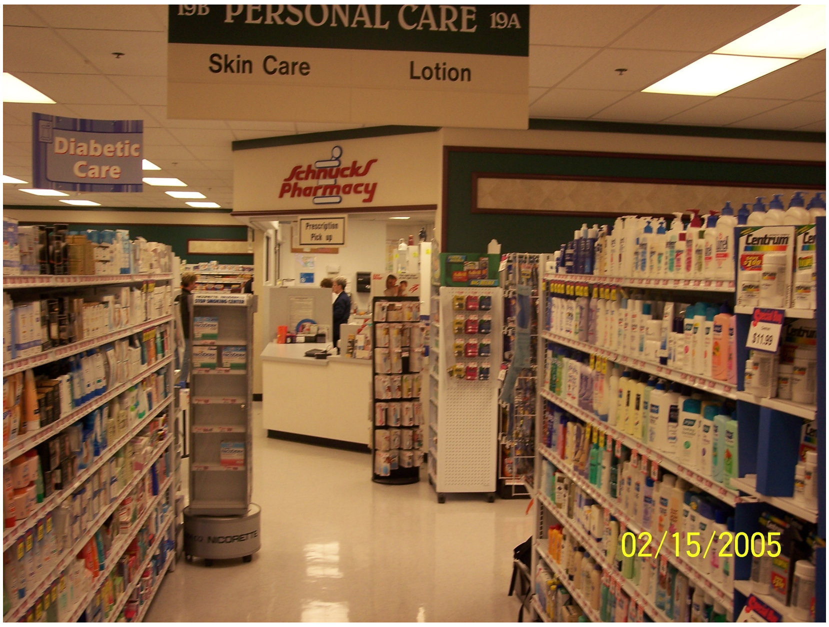


Figure 11: Typical image of Class ”Pharmacy”

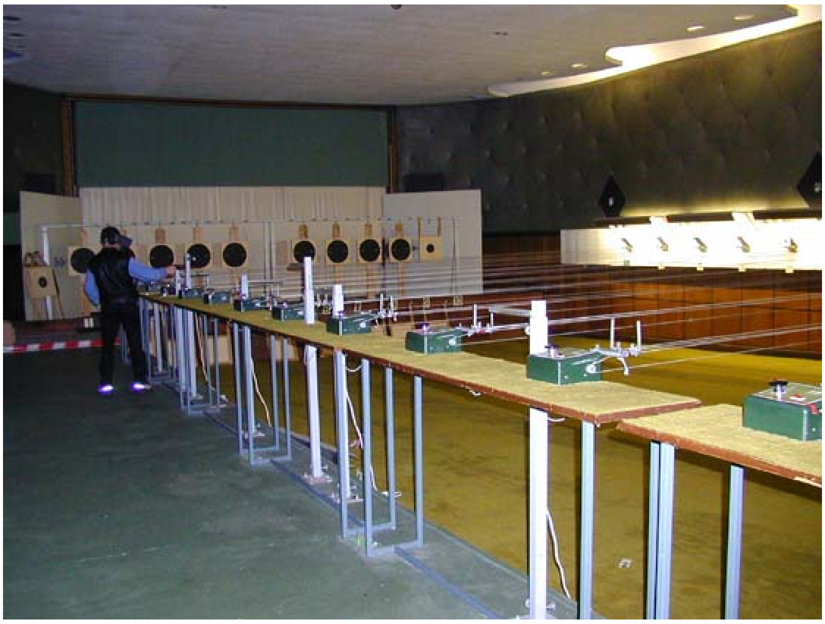


Figure 12: Image of class ”Indoor Firing Range” being wrongly classified as of class ”Pharmacy”

References

[1] $https://plot.ly/ipython-notebooks/principal-component-analysis/$

[2] $https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c$