**Fall 2016  
CSCE 666 Pattern Analysis**

**Homework #2**

**Dennis Rodrigo da Cunha Silva**

**UIN: 122001522**

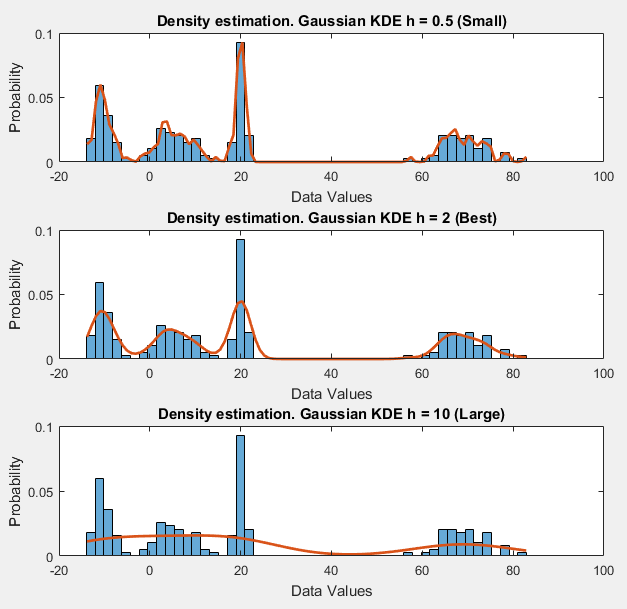
In recognition of the Texas A&M University policies of academic integrity, I certify that I have neither given nor received dishonest aid in this homework assignment.

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Problem 1

*Part 1*

The Gaussian density estimation plot along with the histogram is shown in Figure 1.1. As one can note, the larger the bandwidth (*h*) value, the smoother the Gaussian density estimation. For this particular dataset, when *h = 2* we have a smooth curve as the same time we maintain the characteristics of the original dataset. When *h = 0.5* the density curve is too spiky and for *h = 10*, although we also have a much smoother curve, we lose the characteristics of the data. For this particular part, I have generated 100 points linearly-spaced from the minimum and maximum data value to compute the density estimation.



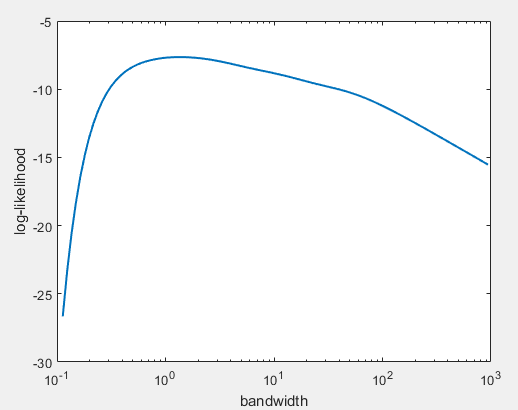
**Figure 1.1:** Histogram and Gaussian density estimation of the data from problem 1. Bandwidth values are 1, 2 and 10. Histogram plot is shown in blue bars and Gaussian density estimation is shown in orange line plot.

*Part 2* (a b c)

For this part I chose the best *h = 2* and performed the leave-one-out-method. The average log-likelihood obtained was equal to: -3.8574.

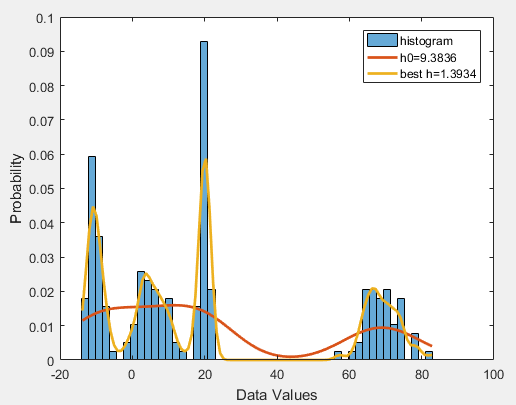
Part 2 (d e f)

Figure 1.2 shows the average log-likelihood computed for different values of *h* by using the leave-one-out-method. The bandwidths were logarithmically-spaced from h0/100 and 100h0. h0 computed for the dataset given is *h0 = 9.38*. As can be seen, the log-likelihood is smaller for small *h’s,* grows up to its maximum value (the bandwidth we are supposed to select) and starts decreasing again. For this case *hbest = 1.3934.*



**Figure 1.2:** Average log-likelihood obtained using the leave-one-out-method for different *h’s*.

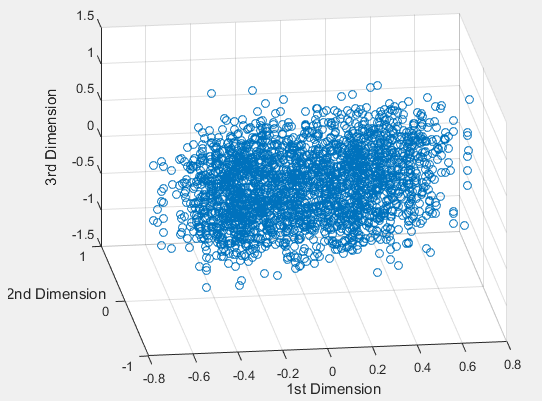
The density estimation plots for *h0* and *hbest*, along with the data histogram can be seen in Figure 1.3. As one can note, the density estimation by using *h0* is much smoother in comparison to the *hbest case.* However, when using *h0* the density estimation computed cannot capture the underlying data density. Therefore, a better density estimation is achieved when using *hbest*.



**Figure 1.3:** Histogram and Gaussian density estimation of the data from problem 1. The density estimations are shown for *h0* and *hbest*.

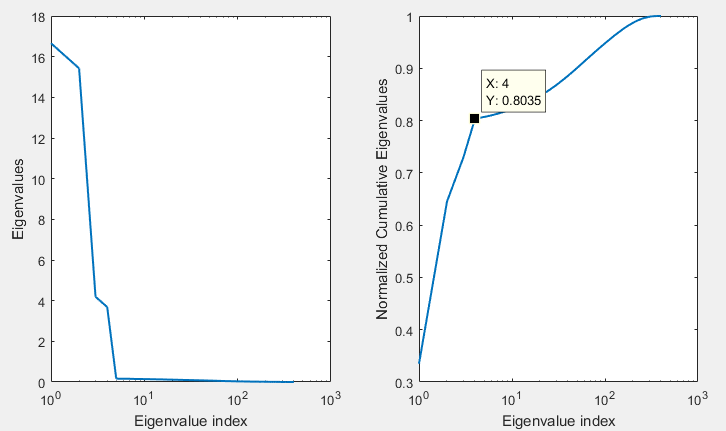
# Problem 2

Figure 2.1 shows the first three dimensions’ scatter plot for the data provided for problem 2. Although I rotated the axis looking for any sort of structure I was not able to find any.



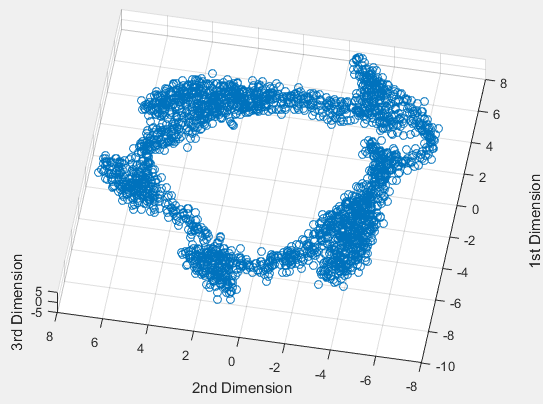
**Figure 2.1:** Scatter plot for the first three dimensions for the data given.

Figure 2.2 shows the eigenvalues plot calculated for the data provided. As one can see in the right side plot, the first four components are responsible for 80% of the data variance. From the 5th component and on, the eigenvalues are not as significant as the first four eigenvalues.



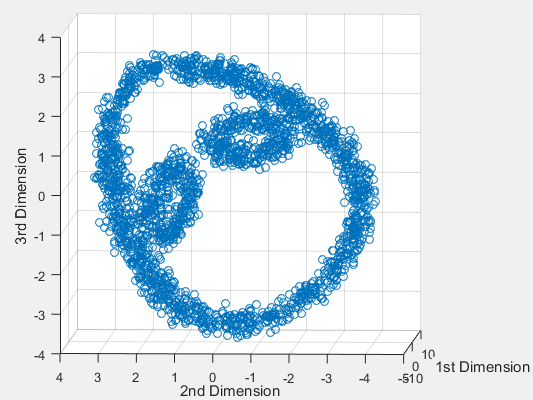
**Figure 2.2:** Scree plot for the data given. In the left, the eigenvalues are shown. In the right, the normalized cumulative eigenvalues are shown.

In Figure 2.3, the scatter plot for the first three PCA projections can be seen. By rotating the graph, a structure similar to a penguin can be seen.



**Figure 2.3:** Data projected into the first three components.

Also, when projecting the data into components 2, 3 and 4, an alien head structure can be seen.

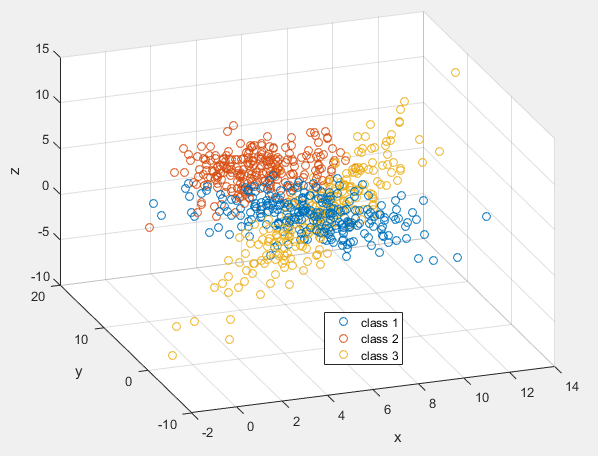


**Figure 2.4:** Data projected into the PCA components 2, 3 and 4.

No other structures were found by combining other PCA components projections when analyzing components 1-4.

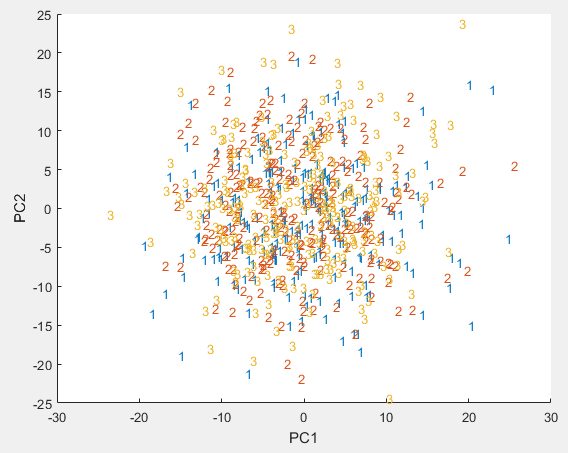
# Problem 3

Figure 3.1 shows a graph for the samples generated for the given distributions for a particular code run.



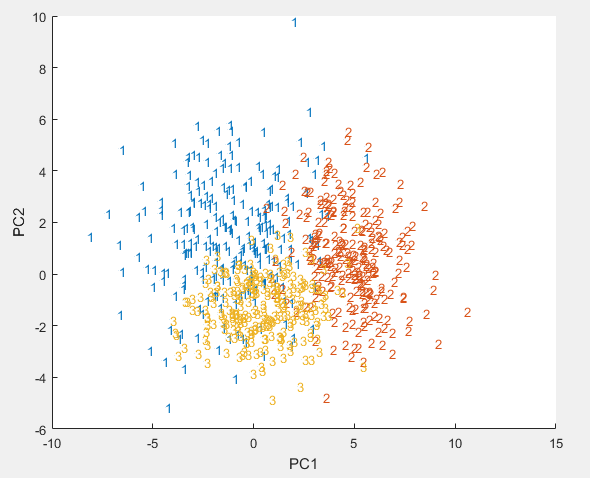
**Figure 3.1:** Samples generated for the given means and covariance matrices.

After adding 48 dimensions with noise the dataset becomes much more complex. Figure 3.2 shows the first two PCA projections. As one should expect, due to the larger variance in the new dataset, PCA projections do not help at all. In other words, by using PCA on the dataset we would not be able to build a model able to classify the data correctly.



**Figure 3.2:** First two PCA projections. The points represented as 1, 2 and 3, represent the data points from class 1, 2 and 3, respectively.

However, computing the first two LDA projections we end up with a better structure. Therefore, by using LDA, it is expected that a model can be built and achieve better accuracy levels when compared to PCA.



**Figure 3.2:** First two LDA projections. The points represented as 1, 2 and 3, represent the data points from class 1, 2 and 3, respectively.

# Problem 4

For this question, due to the covariance matrices characteristics, the quadratic classifier implemented has to be the general one, given by the equation below:

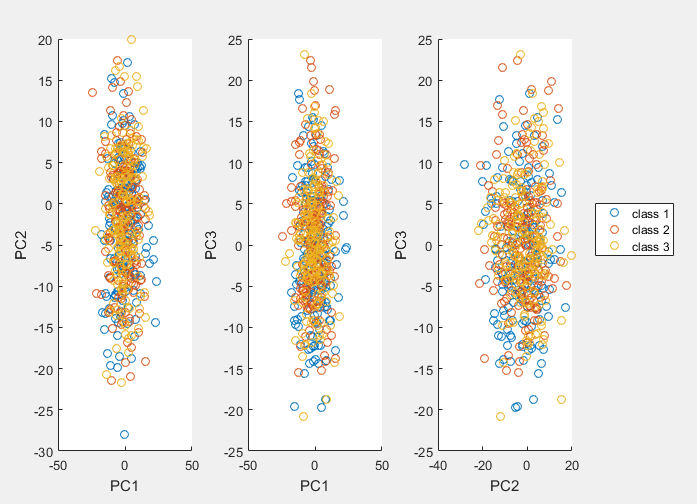
Table 4.1 shows the accuracy obtained by different data configurations. As one can see, the quadratic classifier performed better for the high-dimensional data, than for the lower-dimensional PCA and LDA projected data.

**Table 4.1:** Average accuracy obtained by applying the general quadratic classifier into different data configurations. Number of executions = 30.

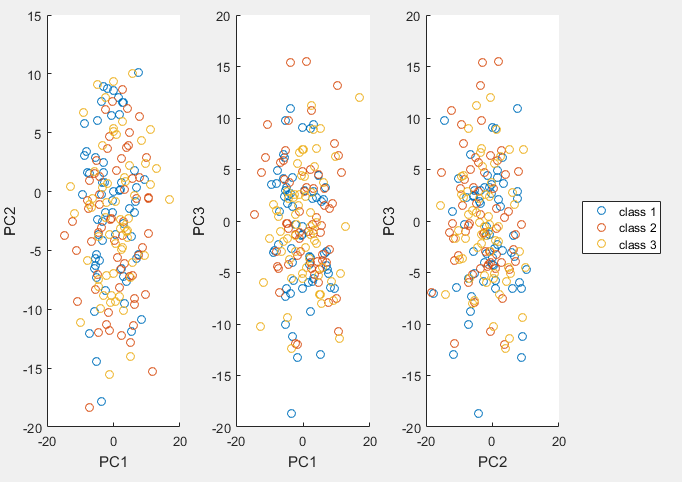
|  |  |
| --- | --- |
| **Data Configuration** | **Accuracy** |
| High-Dimensional data | 80.72% |
| PCA projection | 34.71% |
| LDA projection | 73.35% |

For parts e-f we have to project the for the first three PCA eigenvectors and select a 2-dimensional PCA subspace. In order to do that, I have generated the scatter plot for each pair of components, in order to analyze which of those would make us achieve higher accuracy. The scatter plots for the training and test data can be seen in Figures 4.1 and 4.2, respectively.

As one can note, from analyzing these pairwise scatter plots, no 2-dimensional PCA subspace will make a model achieve high accuracy. Therefore, components 1 and 2 have been chosen.



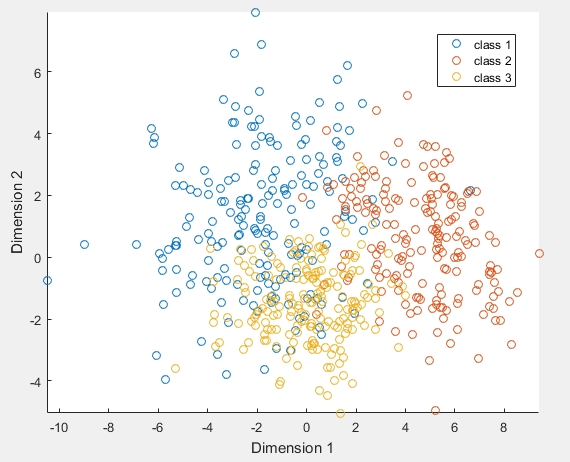
**Figure 4.1:** Pairwise scatter plot for the first three training set PCA projections.



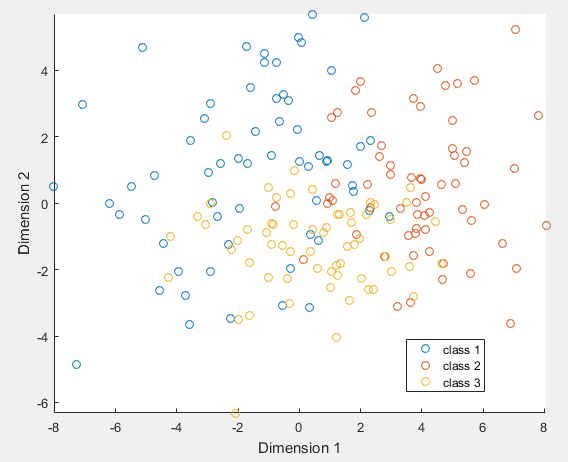
**Figure 4.2:** Pairwise scatter plot for the first three test set PCA projections.

By applying the general quadratic classifier on the 2-dimensional PCA subspace, the accuracy obtained was around 34%. Since the priors for each class in this problem are equal to 33%, the model built do as good as a random classifier. This result is expected, since the noise added to the former dataset has a relative high variance.

Figures 4.3 and 4.4 respectively shows the LDA data projected for the training and test set. As one can note, by using LDA, the dataset derived has samples for each class more concentrated in different regions. It is expected that the classifier implemented performs better in the LCA projected data than in the PCA projected data. According to Table 4.1, the quadratic classifier achieved accuracy around 73%, a much better result when compared to the PCA projected data (34%).



**Figure 4.3:** Pairwise scatter plot for the first two training set LDA projections.

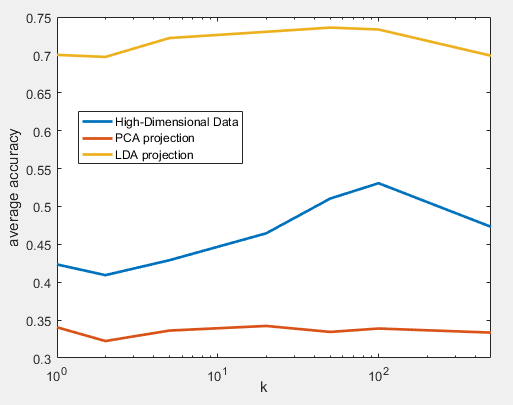


**Figure 4.4:** Pairwise scatter plot for the first two test set LDA projections.

# Problem 5

Figure 5.1 shows the average accuracy for different values of *k*. For this problem the *k’s* considered were 1, 2, 5, 20, 50, 100 and 500. As one can note, by using the LDA projected data, KNN achieved the best results, regardless of the *k* value. The results achieved for this classifier were similar to those found in the previous question for the PCA and LDA projected data. KNN achieved similar accuracies, for the PCA and LDA projected data, for different values of *k*.

However, KNN did not performed as good as the quadratic classifier for the original high-dimensional data. This phenomenon happened because the Euclidian Distance used as measure of similarity is affected by the noise in the original dataset. Although higher values of *k* make the classifier perform slightly better for this case, the results achieved were still not as good as those achieved by the quadratic classifier.



**Figure 5.1:** Average accuracy obtained for different *k* values with different datasets. Number of iterations = 30.

# Problem 6