Austin Bianchini (20506675)
Abilas Sathiyanesan (20567746)
Rashmi Umashankar (20562741)
Nakash Ali Babwany (20586425)

Lab 2 Report

1 Introduction

This lab examines the areas of statistical model estimation and classifier aggregation. In cases where the density functions are not known a priori, it may be possible to use labeled samples to obtain estimates. Model estimation will be performed by implementing parametric and non-parametric estimators. Aggregation is introduced by combining several simple linear discriminants into one more powerful classifier.

2 Model Estimation 1-D Case

In this part of the lab, model estimation is done for 1-D data sets which are as follows:

- variable a - a bunch of Gaussian samples with $\mu = 5$, $\sigma = 1$.
- variable b - a bunch of Exponential samples with $\lambda = 1$

Estimation is done using both parametric and non-parametric approaches. In parametric estimation, the class conditional probability is known except for some set of defining parameters. There are two methods of parametric estimation namely, *Maximum Likelihood* and *Maximum a Posteriori Estimation*. For this lab, we use the ML approach for parametric estimation where the parameters are treated as fixed but unknown quantities. The problem of estimation then reduces to finding estimate values which maximize the probability that the given samples came from the resulting PDF. We also examine non-parametric estimation in which the functional form of the class PDFs is not known. Instead, we try to directly estimate the class distribution or density $p(\underline{x})$ from the samples $\underline{x}_i$.

2.1 Parametric Estimation - Gaussian

The unknown density is assumed to be a univariate Gaussian. Since both the mean and the variance are unknown, the parameter θ is then a vector:

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \mu \\ \sigma^2 \end{bmatrix}$$

The class PDF is then of the form:

$$p(\underline{x}_i|\underline{\theta}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\theta_2}} \exp\Big[-\frac{1}{2}\frac{(x_i - \theta_1)^2}{\theta_2} \Big]$$

Expressing the above equation in its log-likelihood form and taking the derivative with respect to each parameter lets us maximize the likelihood function. It follows that the ML estimate for the mean (μ) and the variance (σ^2) are the sample mean and the sample covariance respectively.

$$\mu_{est} = \hat{\theta}_{1,ML} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma^2_{est} = \hat{\theta}_{2,ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\theta}_1)^2$$

The estimated values of mean and covariance for classes a and b were calculated based on the above equations using the *lab2_1.mat* data set provided and were found to be as follows:

Gaussian samples:

$$\mu_{est,a} = 5.076 \;\; \sigma^2_{est,a} = 1.062$$

Exponential samples:

$$\mu_{est,b} = 0.963 \;\; \sigma^2_{est,b} = 0.929$$

The estimated distribution, $\hat{p}(x)$ was obtained using the above estimated parameters above. Figure 1 shows the estimated $\hat{p}(x)$ superimposed with the actual $p(x)$.

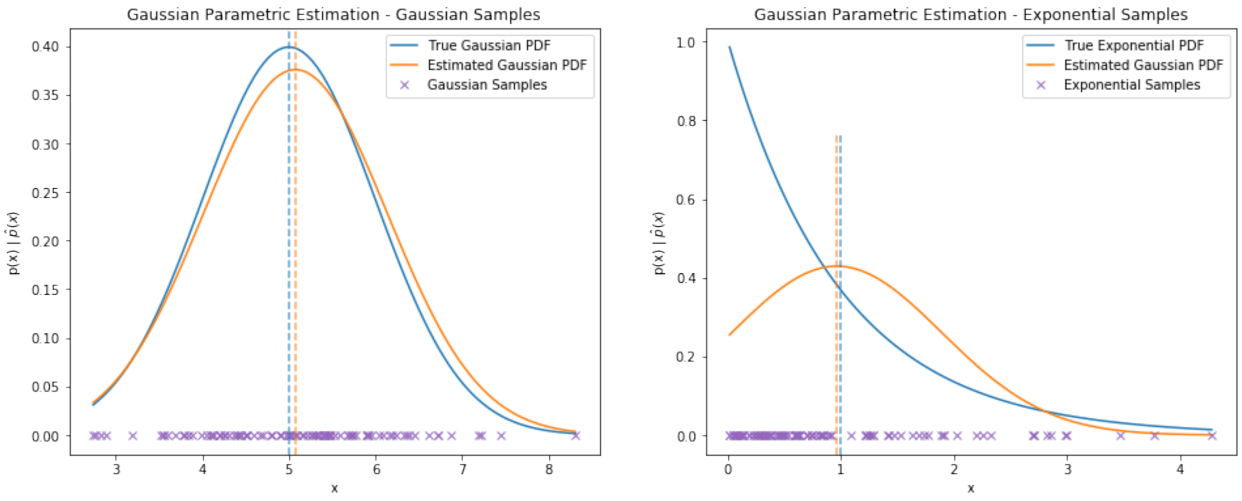

Figure 1: Gaussian Parametric Estimation - Class A & B

As expected, assuming a Gaussian form for a distribution that is exponential in nature wouldn't produce the best estimate and this is reflected in estimation of class B samples. For class A, since the true form of the PDF is also Gaussian, the estimate resembles the actual PDF very well.

2.2 Parametric Estimation - Exponential

In this case, the general form of the PDF is assumed to be exponential in nature and the parameter to be estimates is λ.

Therefore:

$$p(\underline{x}_i|\hat{\lambda}) = \prod_{i=1}^{N} \hat{\lambda} e^{-\hat{\lambda} x_i} = \hat{\lambda}^N e^{-N\hat{\lambda} x_i}$$

Nx_i can be written as $\sum_{i=1}^{N} x_i$ and hence, the above equation reduces to

$$p(\underline{x}_i|\hat{\lambda}) = \hat{\lambda}^N e^{-\hat{\lambda}\sum_{i=1}^{N} x_i}$$

Taking the log,

$$l(\hat{\lambda}) = Nln(\hat{\lambda}) - \hat{\lambda}\sum_{i=1}^{N} x_i$$

Taking the derivative of $l(\hat{\lambda})$ with respect to $\hat{\lambda}$ and setting it zero, we can find the estimated value of λ which as follows:

$$\hat{\lambda} = \frac{N}{\sum_{i=1}^{N} x_i}$$

The estimated values of λ for classes a and b were calculated based on the above equation using the *lab2_1.mat* data set provided and were found to be as follows:

Gaussian samples:

$$\lambda_{est,a} = 0.197$$

Exponential samples:

$$\lambda_{est,a} = 1.038$$

The estimated distribution, $\hat{p}(x)$ was obtained using the above estimated λ. Figure 2 shows the estimated $\hat{p}(x)$ superimposed with the actual $p(x)$.

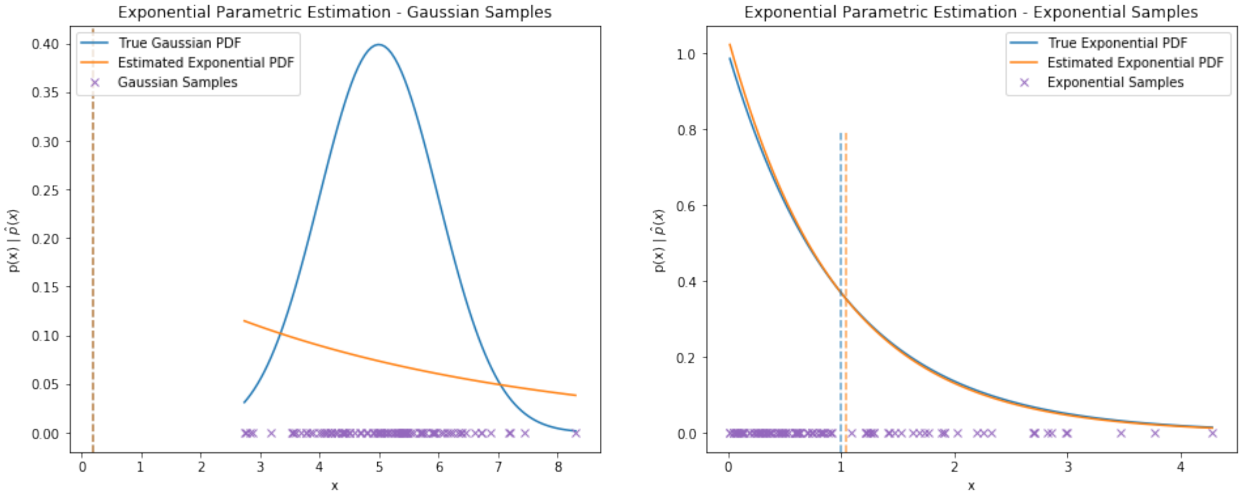

Figure 2: Exponential Parametric Estimation - Class A & B

As expected, assuming an Exponential form for a distribution that is Gaussian in nature wouldn't produce the best estimate and this is reflected in estimation of class A samples. For class B, since the true form of the PDF is also Exponential, the estimate resembles the actual PDF very well.

2.3 Parametric Estimation - Uniform

In this case, the general form of the PDF is assumed to be uniform in nature and hence, two parameters need to be estimated: a and b.

The PDF is of the form

$$p(\underline{x}_i|\hat{a},\hat{b}) = \prod_{i=1}^{N} \frac{1}{\hat{b}-\hat{a}}$$

The log likelihood form of this is

$$l(\hat{a},\hat{b}) = -Nlog(\hat{b}-\hat{a})$$

Taking the derivative with respect to each variable, we get,

$$\frac{\partial l(a,b)}{\partial a} = -\frac{n}{log(b-a)}$$

$$\frac{\partial l(a,b)}{\partial b} = \frac{n}{log(b-a)}$$

Maximizing the first equation would be achieved through minimizing $(b-a)$ by making a as large as possible. Maximizing the second equation would be achieved through minimizing $(a-b)$ by making b as large as possible. In both cases however, the MLE is achieved through making $(b-a)$ as small as possible. As such, for any data set $[X_0, .., X_n]$,

$b_{est} = MAX[X_0, .., X_n]$

$a_{est} = MIN[X_0, .., X_n]$

The estimated values for a and b are as follows:

Gaussian samples:

$$a_{est,a} = 2.741 \;\; b_{est,a} = 8.308$$

Exponential samples:

$$a_{est,b} = 0.014 \;\; b_{est,b} = 4.280$$

The estimated distribution, $\hat{p}(x)$ was obtained using the above estimated a and b. Figure 3 shows the estimated $\hat{p}(x)$ superimposed with the actual $p(x)$.

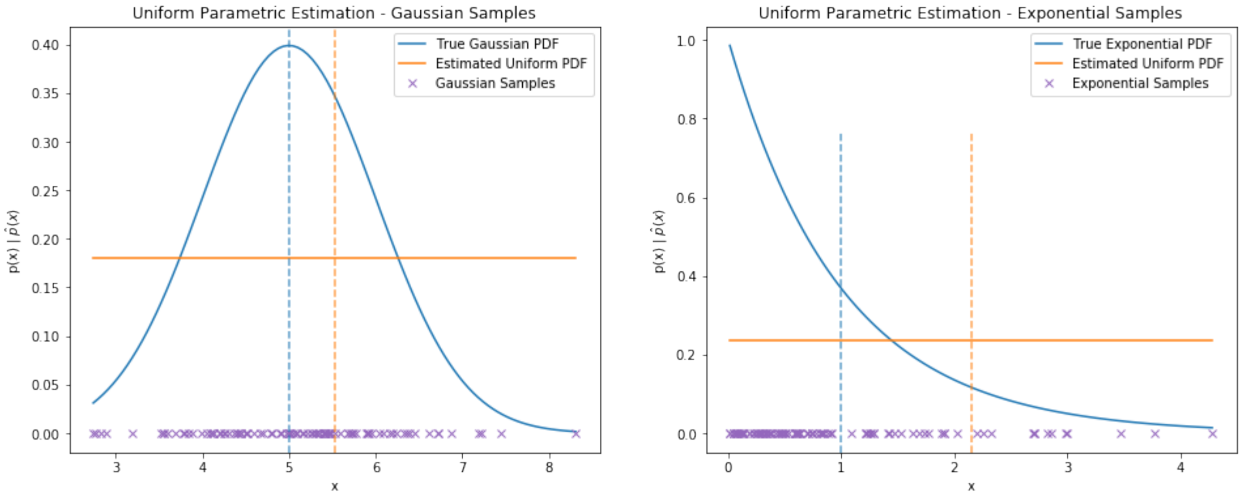

Figure 3: Uniform Parametric Estimation - Class A & B

Assuming a uniform distribution as an estimate for distributions that are Gaussian and Exponential in nature can be expected to give poor results. This is reflected in Figure 3 where the estimated PDF is very different from the actual distributions of class A and B.

2.4 Non Parametric Estimation

Here, we do not know the functional form of the class PDFs and rely on direct estimation of the PDF from the samples x_i. Although there are multiple appraoches to non parametric learning, we adopt the Parzen Window Estimation method for the purpose of this lab.

The main idea behind this approach is that every sample x_i locally influences the estimated PDF in the vicinity of x_i. The estimated PDF is then the sum of the contributions from each sample and is given by:

$$\hat{p}(x) \propto \sum_{i=1} \phi(x - x_i)$$

where ϕ is a window function which controls how each observed sample influences the PDF. We can also stretch or compress the function.

The normalized Parzen PDF estimate based on window function ϕ:

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h^n} \phi \frac{(x - x_i)}{h}$$

Our window function is Gaussian in nature and the scaling factor is taken to be proportional to the standard deviation of the Gaussian window. Two different values were investigated namely: $\sigma = 0.1$ and $\sigma = 0.4$.

The estimated densities for the classes A and B for different values of σ are shown in Figures 4 and 5.

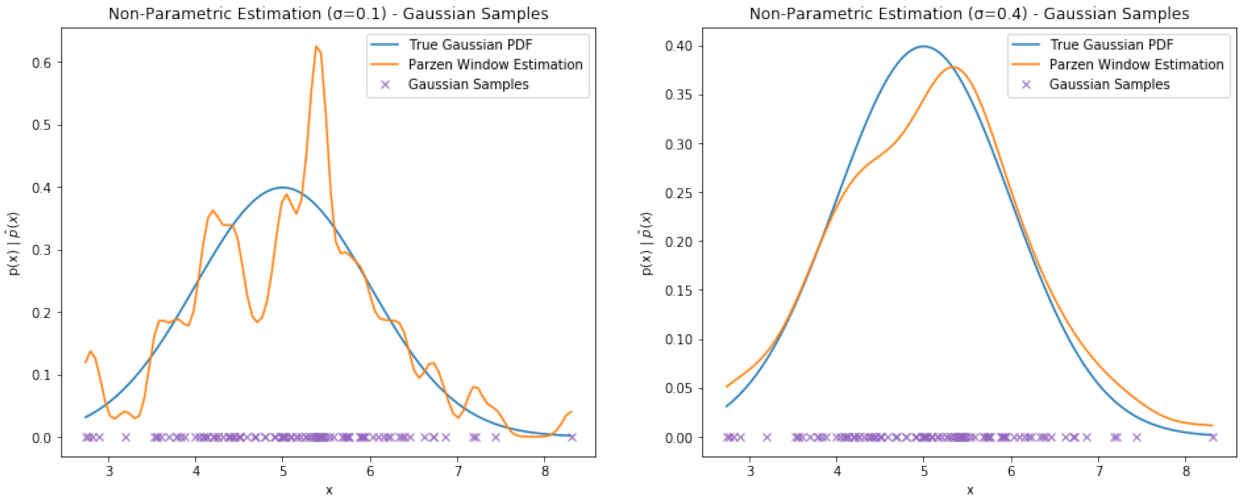

Figure 4: Non Parametric Estimation - Class A with $\sigma = 0.1$ and $\sigma = 0.4$

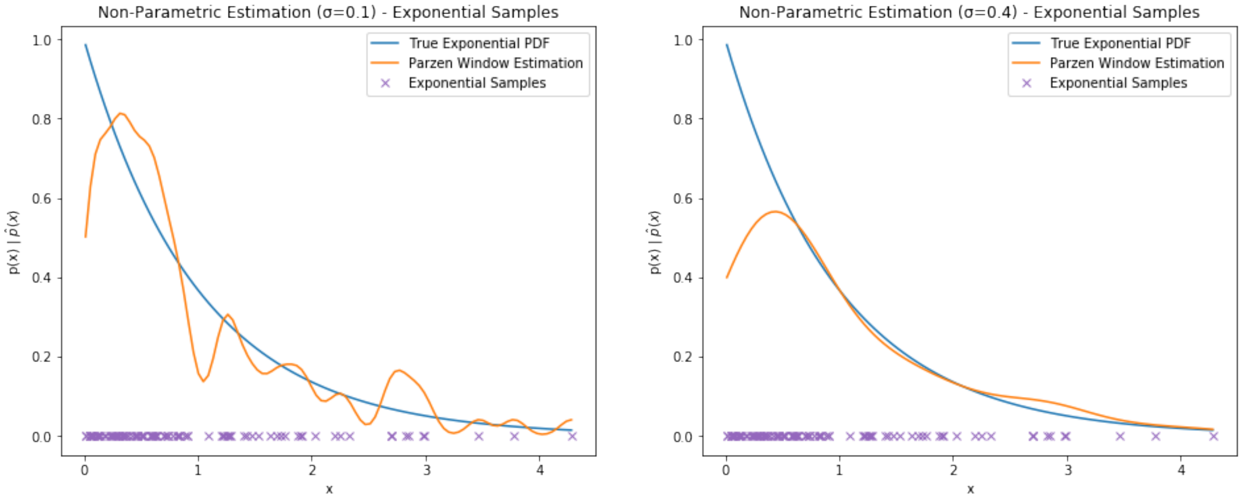

Figure 5: Non Parametric Estimation - Class B with $\sigma = 0.1$ and $\sigma = 0.4$

The expansion factor h influences how good the estimate is and is usually chosen by trial and error. If h is too small, then the estimate will be noisy. If it is too large, the estimate is smeared and has low resolution. As seen in both classes A and B, the smaller value of σ results in a relatively noisy estimated density as compared to $\sigma = 0.4$ which gives us a smoother estimate. In class A (Gaussian samples), since the window is also Gaussian in nature, the estimate performs well especially in the case of $\sigma = 0.4$. In class B, even though the true distribution is exponential in nature and the window is Gaussian, the Parzen window method still manages to give a decent estimate.

2.5 Discussion

From the above results, it can be seen that the performance of the parametric approach is heavily influenced by the assumption of the form of the PDF. For class A, where the true density is Gaussian, the best estimation was done when the form of the PDF was also assumed to be Gaussian. Similarly for class B, the best estimation was done when the form of the PDF was assumed to be exponential. In cases where the assumption did not align with the true density like in the case of the Uniform distribution, the estimation is sub optimal. For both classes, the Parzen window method produced a decent estimate but was not as good as parametric case where the assumption matched the form of the actual PDF.

In general, it is better to use the parametric approach when the form of the PDF of the data set is known. In the case it is not known, then it is advisable to use the non parametric approach.

3 Model Estimation: 2D Case

A similar investigation of different methods was carried out with the 2D data sets *al, bl* and *cl* provided. In this case, the true PDF of the different data sets is not known.

3.1 Parametric Estimation

It is assumed that the PDF is Gaussian in nature. Similar to section 2.1, the estimated mean and covariance are just the sample mean and covariance of the data set.

$$\underline{\mu}_{est} = \frac{1}{N}\sum_{i=1}^{N}(\underline{x}_i)$$

$$\underline{\Sigma}_{est} = \frac{1}{N}\sum_{i=1}^{N}(\underline{x}_i - \underline{\mu}_{est})(\underline{x}_i - \underline{\mu}_{est})^T$$

The calculated sample mean and covariances for the three data sets are given as follows:

$$\mu_{est,a} = \begin{bmatrix} 341.6 \\ 131.2 \end{bmatrix} \Sigma_{est,a} = \begin{bmatrix} 1748.9 & -1594.4 \\ -1594.4 & 3310.1 \end{bmatrix}$$

$$\mu_{est,b} = \begin{bmatrix} 291.8 \\ 224.0 \end{bmatrix} \Sigma_{est,b} = \begin{bmatrix} 3282.5 & 1164.2 \\ 1164.2 & 3379.8 \end{bmatrix}$$

$$\mu_{est,c} = \begin{bmatrix} 119.5 \\ 346.6 \end{bmatrix} \Sigma_{est,a} = \begin{bmatrix} 2711.1 & -1313.9 \\ -1313.9 & 1682.3 \end{bmatrix}$$

Instead of plotting the estimated PDFs, ML classification boundaries were calculated for 2 classes at a time and plotted.

For a 2-class classification problem, the discriminant function for ML is as follows:

$$(\underline{x} - \mu_B)^T \Sigma_B^{-1} (\underline{x} - \mu_B) - (\underline{x} - \mu_A)^T \Sigma_A^{-1} (\underline{x} - \mu_A) \underset{B}{\overset{A}{\gtrless}} \left[\frac{|\ \Sigma_A\ |}{|\ \Sigma_B\ |} \right]$$

The decision boundary is given by:

$$\underline{x}^T Q_0 \underline{x} + Q_1 \underline{x} + Q_2 + Q_4 = 0$$

where

$$Q_0 = S_A^{-1} - S_B^{-1}$$
$$Q_1 = 2[\underline{m}_B^T S_B^{-1} - \underline{m}_A^T S_A^{-1}]$$
$$Q_2 = \underline{m}_A^T S_A^{-1} \underline{m}_A - \underline{m}_B^T S_B^{-1} \underline{m}_B$$
$$Q_3 = \left[ln \frac{P(B)}{P(A)} \right]$$
$$Q_4 = \left[\frac{\mid S_A \mid}{\mid S_B \mid} \right]$$

The resultant plot obtained using the estimated means and covariances is shown in Figure 6.

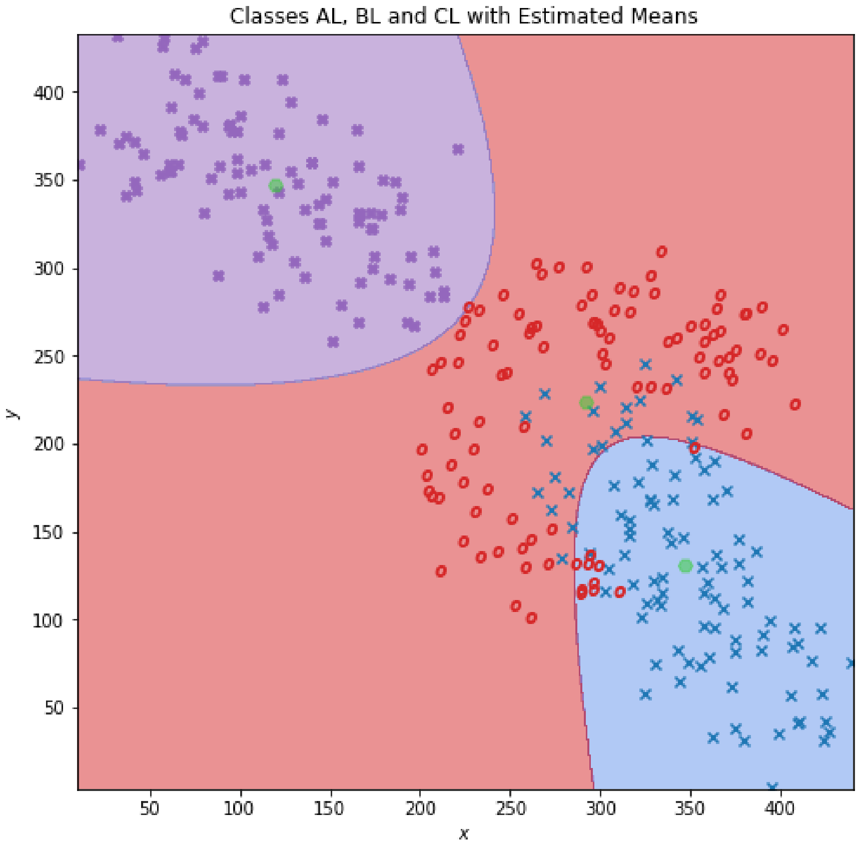

Figure 6: ML Decision Boundary based on Estimated PDF

3.2 Non Parametric Estimation

For this part, we extend the implementation of Gaussian Parzen window in section 2.4 to 2D. The estimation was done using SciPy's gaussian kernel density estimation feature. The PDF was estimated for each class which was then used to build a ML classifier. The resultant plot is shown in Figure 7.

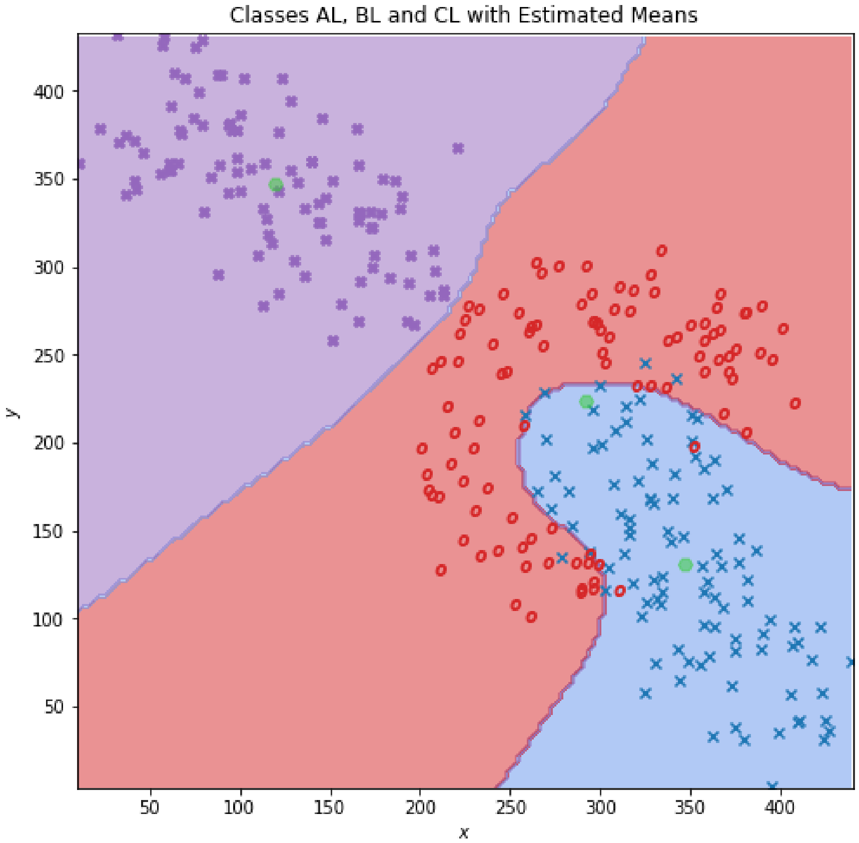

Figure 7: ML Decision Boundary based on Estimated PDF

3.3 Comparison

For the 2D case, we do not have information about the true nature of the class distributions. To do parametric estimation, we assumed a gaussian form for the class densities. As we've seen in section 2, parametric approach works best when we have prior knowledge of the class PDF so that we can make the right assumption and estimate the right parameters. Despite not knowing the true PDF, parametric estimation works quite well for our three data sets. However, some of the blue data points are wrongly classified as red and some of the red wrongly classsified as blue. This misclassification is definitely reduced in the case of non parametric estimation. By changing the window size, we were able to fit the data more accurately. However, it is important to note that it is a trade-off between getting better accuracy and also preventing overfitting of the training set. In cases like this where the prior PDFs are not known, it is better to use non parametric estimation. There is a higher chance of misclassification opting for a parametric approach and making assumptions that might not reflect the actual form of the PDFs.

4 Sequential Discriminants

The training and classification algorithms described in the question were implemented using Python. New discriminants were added to a list until both sets, a and b, were empty or until the limit of the J value was reached.

4.1 Learning Three Sequential Classifiers

Three classifiers were trained on the training set provided and since the algorithm entails using random points in each set as prototypes for the MED classifier, all three classifiers had slightly different decision boundaries, which are plotted below for each classifier:

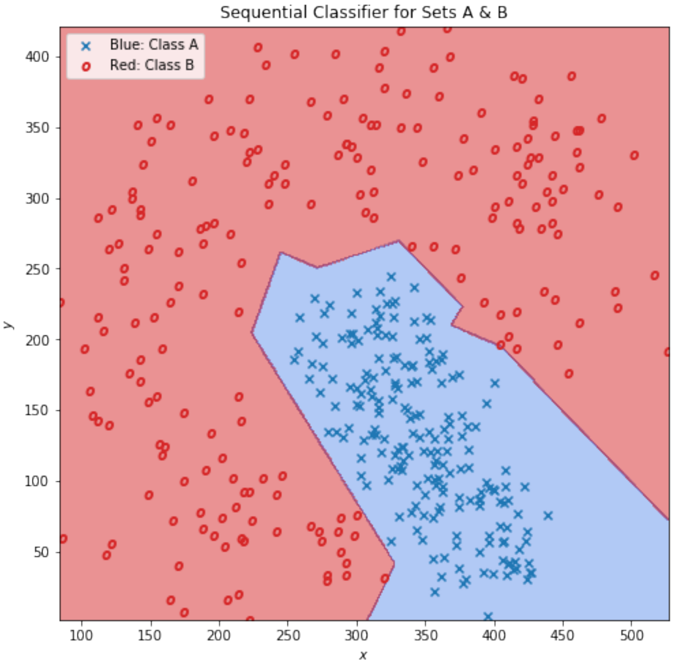

Figure 8: 1st Classifier

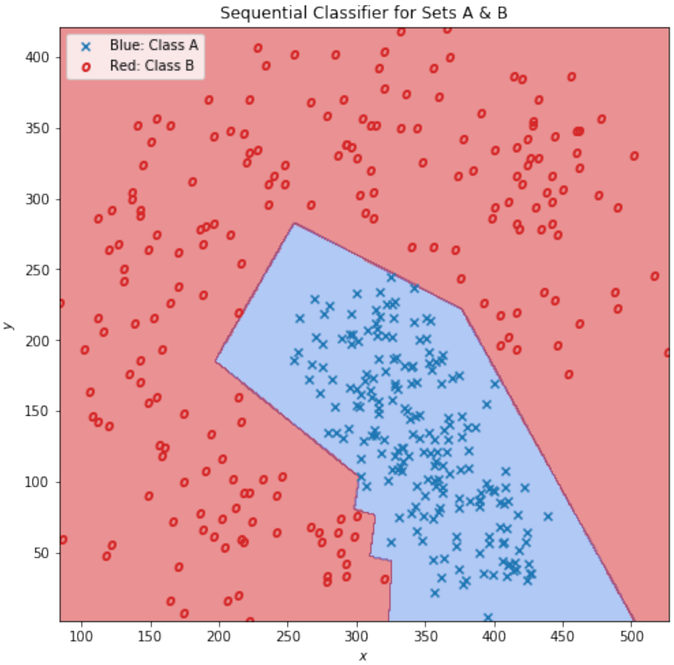

Figure 9: 2nd Classifier

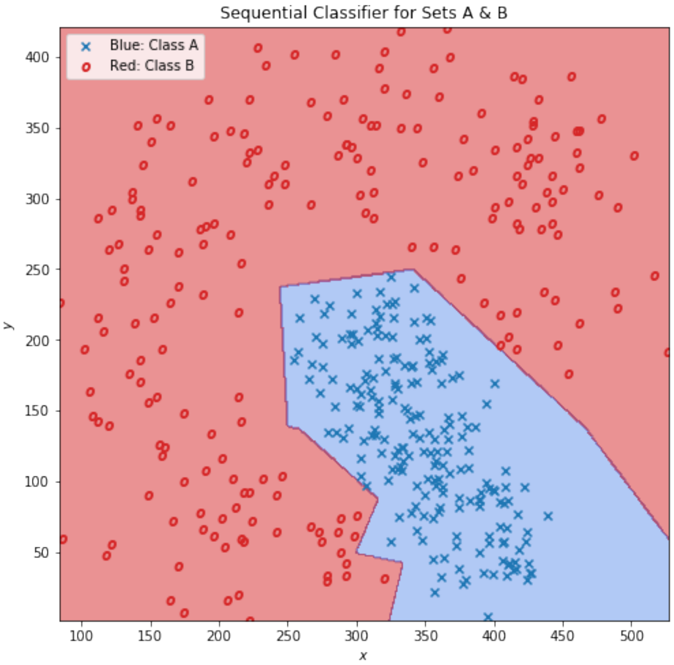

Figure 10: 3rd Classifier

In order to see the inner workings of such a sequential classifier, the decision boundaries of the first three determinant classifiers are also plotted below:

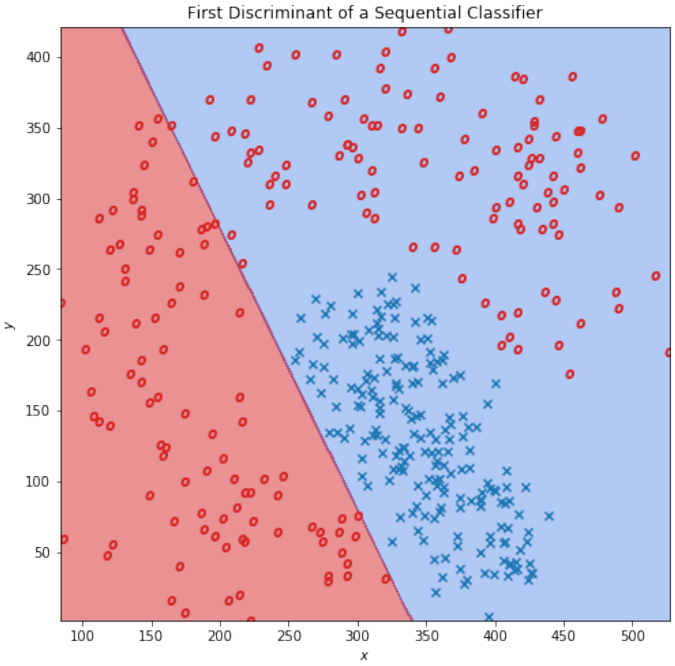

Figure 11

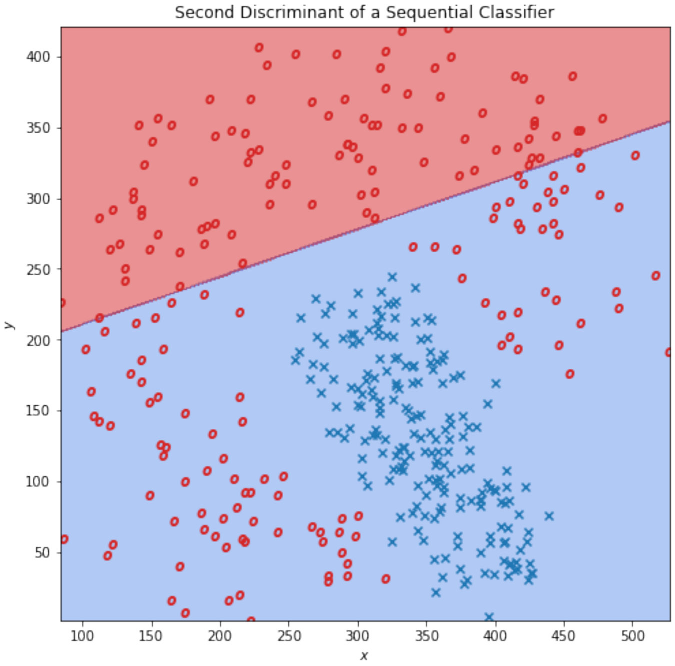

Figure 12

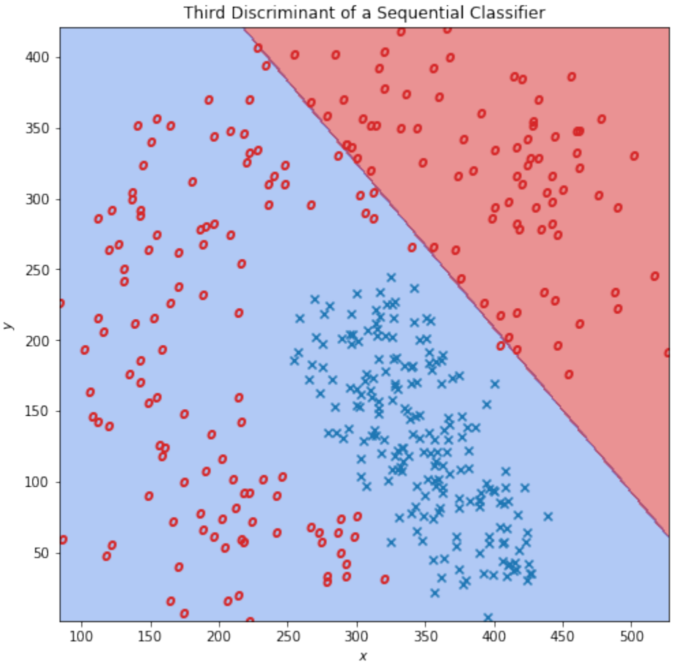

Figure 13

It can be seen from the plots above that each of the first three determinants classifies all samples in Class A correctly, as required. Moreover, it can also be seen that the final decision boundaries that are shown in Figures 8, 9 and 10 are an amalgamation of multiple discriminants such as the three in Figures 11, 12 and 13.

4.2 Testing Classifiers on Training Data

If the sequential classifiers from the section above are tested on training data, the probability of error should be 0 because each classifier was built by making sure that every sample in the training set could be classified by at least one of the J discriminants. Since there was no limit to J (i.e. how many discriminants can be added to the classifier) and since the training algorithm terminated with empty a and b sets, any sample from the training data would be classified correctly. If, however, a limit were to be imposed on J, then the classifiers we train might fail to classify some samples from the training set as the algorithm may not terminate with empty a and b sets.

4.3 Limiting the Number of Discriminants

Instead of allowing the sequential classifier to use as many discriminants as needed, the algorithm from section 4.1 is modified to limit the sequential classifier to J discriminants, where J is varied from 1 to 5. The resulting error rates are plotted below:

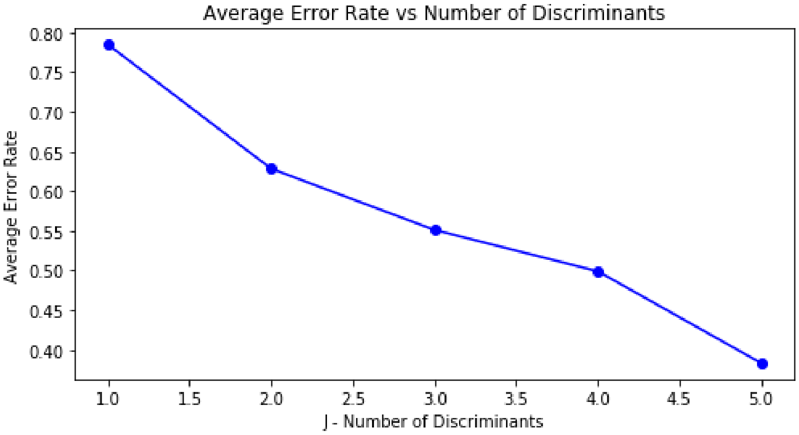

Figure 14: Average Error Rate

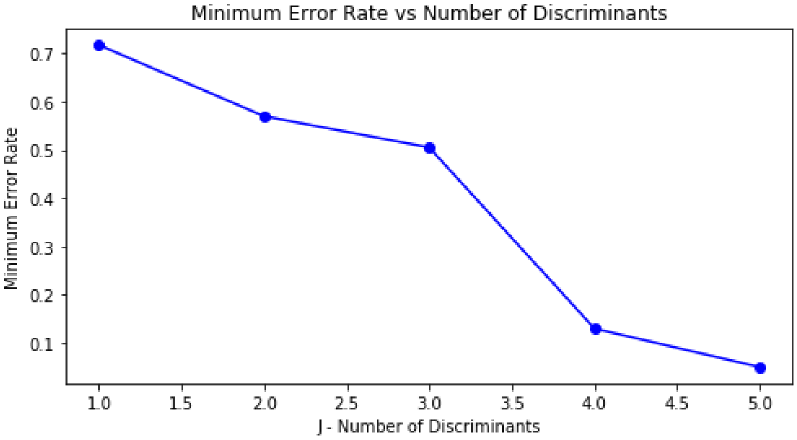

Figure 15: Min Error Rate

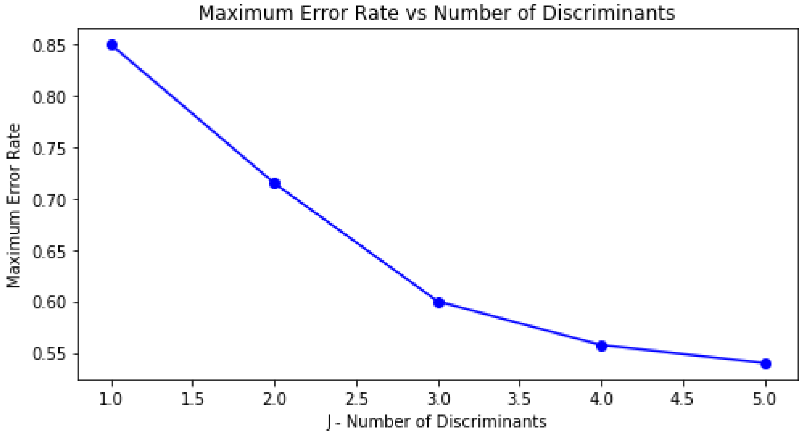

Figure 16: Max Error Rate

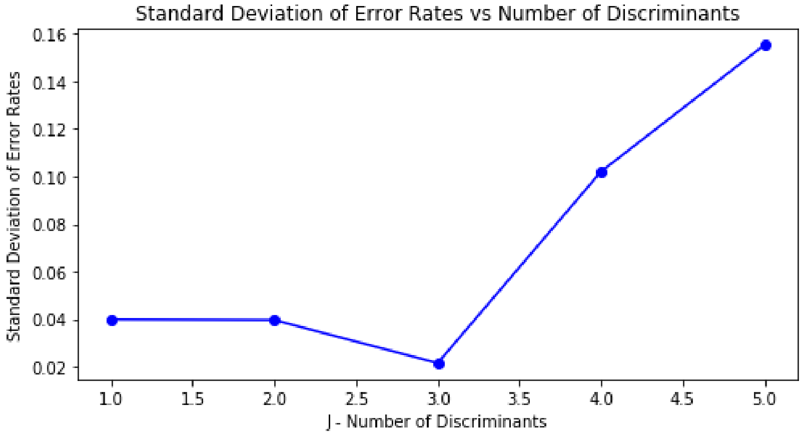

Figure 17: Standard Deviation of Errors

The results above show that as the number of discriminants, J, is increased, the average, minimum and maximum error rates all decrease. This is expected because as more discriminants are added, the sequential classifier gains the ability to classify an additional group of samples which it couldn't classify earlier. It can also be noticed that when J is small the difference between the maximum and minimum error rates is small (around 15 %); however, when J is increased, this difference increases to around 45 %. This is the reason behind the standard deviation of the error rates increasing as J is increased. Since the training process relies on randomness, the minimum and maximum error rates deviate from each other more as the classifier gets more and more accurate. When the accuracy of the classifier is low (i.e. the error rate is around 0.8) then the randomness doesn't change this error rate by much because it is already very high, thus causing a relatively low standard deviation of errors when J=1, J=2 and J=3.

4.4 Effect of Limiting the Number of Samples in the Training Process

If the number of points that can be used is limited, then some elements of each class would be left out and thus, the sequential classifier may not have a discriminant which can correctly classify those point pairs. This can also be demonstrated by Figure 8, Figure 9 and Figure 10 in section 4.1 where some of the samples can be seen to be very close to the decision boundaries. If we imagine a case where some of the point pairs of Class A were left out of this data set and those point pairs happened to be on the red side of the decision boundaries, then these classifiers would fail to classify them correctly and the error rate would increase.

However, in practice, the classifiers should be tested and trained on different data sets of the same classes so that over-fitting can be avoided. In such cases, a perfectly accurate sequential classifier is a very unrealistic achievement for most purposes.

5 Conclusion

Through the results and findings presented in this report, it can be concluded that parametric estimation is appropriate when the type of distribution of the Probability Density Function (PDF) is known, or if a safe assumption can be made about the form of the PDF. However, if this information is entirely unknown, then non-parametric estimation should be the preferred method. However, when using non-parametric estimation, much experimentation should be done in order to determine the variance of the Gaussian window or the bandwidth of the estimator that is being used.

Moreover, when using sequential classifiers, a reasonable limit should be imposed on both the number of determinants that the classifier uses and the number of data points that are left out of the training set in order to test the accuracy of the classifier. By maintaining this balance well, over-fitting can be avoided and a reasonably accurate sequential classifier can be constructed. A good way to select how many discriminants to use would be to plot the accuracy of the classifier as a function of the number of discriminants and determine when the plot exceeds a reasonable threshold of accuracy.