Department of Systems Design Engineering

SYDE 372: Pattern Recognition

Lab 2: Model Estimation and Discriminant Functions

Syeda Zainab, 20550724
Abhinav Grover, 20557610
Lalit Lal, 20572296

March 29, 2019

Introduction

As mentioned in the lab manual, this lab examines the areas of statistical model estimation and classifier aggregation. 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, as seen in the sequential discriminant section in part three.

The first section discusses model estimation and its implementation in the univariate case, and the second section discusses model estimation and its implementation in the bivariate case (2D) case. Finally, the third part discusses the implementation, results, and statistics of implementing a sequential discriminants (aggregation) classifier.

1. Model Estimation 1-D Case

As learned in class, model estimation consists of two variants - parameterized and non-parameterized. The data used for this section comes from the course website, a file named “lab2_1.mat” as mentioned in the lab manual.

The parameterized method involves assuming a probability density function form to fit the data, and trying to estimate the parameters of the probability density function. In this case, a Maximum Likelihood (ML) estimation technique will be used to estimate the PDF parameters of each dataset. Three different PDFs will be used to estimate the datasets - Gaussian, Exponential, and Uniform. The estimation and their true distributions will be overlayed with one another to show accuracy of the results. The overall process of estimating the PDF parameters is as follows:

a) Assume the sample are independent of each other, such that
$p(\{x_i\}|\theta) = p(x_1, x_2, ... , x_N|\ \theta) = \prod_{i=1}^{N} p(x_i|\theta)$

b) Get the log form of the the conditional probabilities such that
$l(\theta) = log[p(\{x_1\}|\theta)] = \sum_{i=1}^{N} logp(x_i|\theta)$

c) Take the derivative of the log form and set it to zero such that
$\frac{\partial}{\partial\theta} l(\theta)|_{\theta=\hat{\theta}_{ML}} = 0$

For the non-parameterized model estimation, the Parzen window technique will be used with two different standard deviations, 0.1 and 0.4, as mentioned in the lab manual.

1.1 Gaussian Parametric Estimation

The form of a univariate Gaussian PDF is shown below as a function of mean μ and standard deviation, σ.

$$F_{Gauss}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

For a Gaussian distribution, the parameters are setup as $\bar{\theta} = [\theta_1\ \theta_2]^T = [\mu\ \sigma^2]^T$. The log function is given as

$$l(\bar{\theta}) = log[\sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\theta_2}} e^{-\frac{1}{2}\frac{(x-\theta_1)^2}{\theta_2}}].$$

Using the generalized approach above, and as shown in the course notes, the parameter estimations can be determined from the samples as

$$\mu \ = \ \widehat{\theta}_{1ML} = \ \frac{1}{N}\sum_{i=1}^{N} x_i \, ,$$

$$\sigma^2 \ = \ \widehat{\theta}_{2ML} = \ \frac{1}{N}\sum_{i=1}^{N} (x_1 - \widehat{\theta}_{1ML})^2$$

where N is the total number of samples in the dataset.

Using the above outlined Maximum Likelihood Estimation on two datasets, A and B, their approximate Gaussian distributions were plotted and can be seen in Figure 1.1(a) and 1.1(b). Built-in MATLAB functions were used to estimate the PDFs.

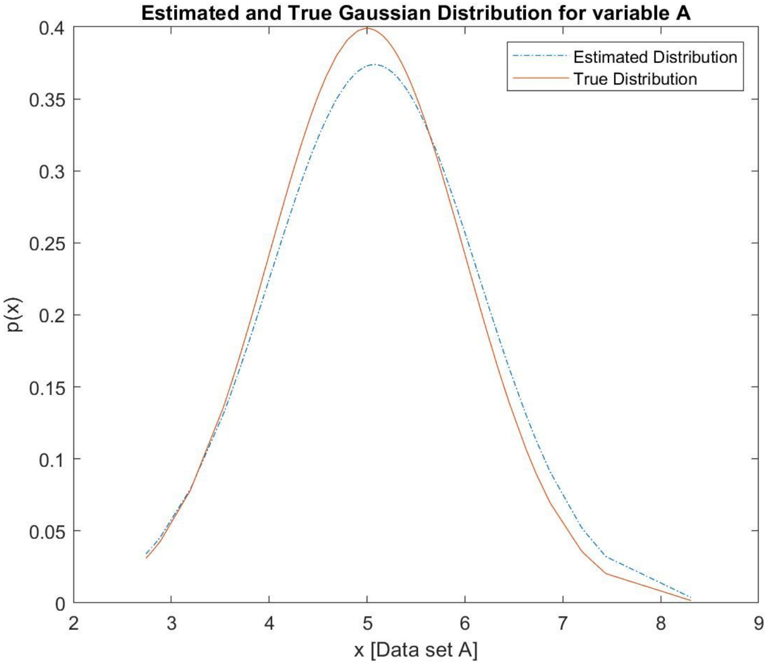

Figure 1.1(a): Estimated and True Gaussian distribution for dataset A

Since dataset A was Gaussian distributed with a true mean of 5 and a standard deviation of 1, the estimated Gaussian was a close fit to it. There is slight deviation from the true distribution caused by the noise and outliers in the dataset.

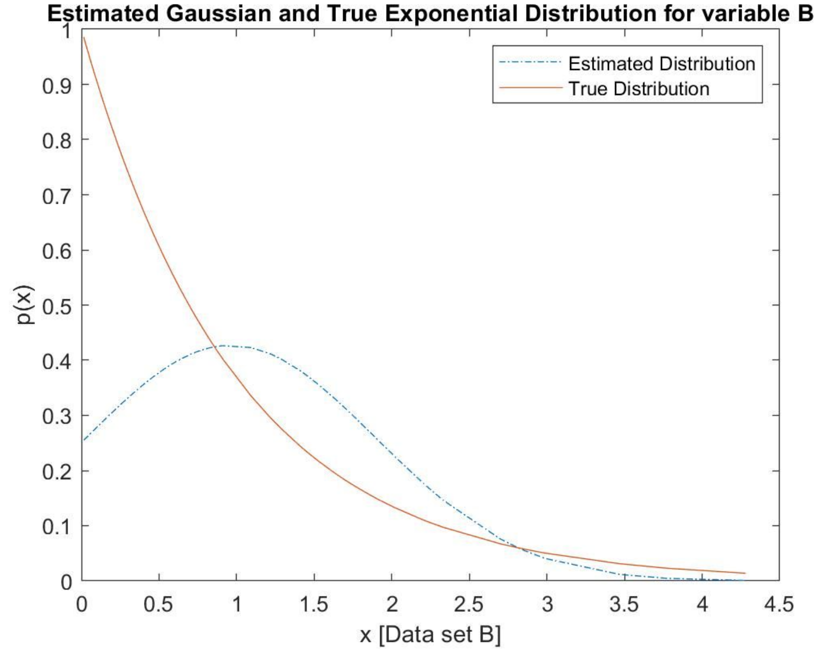

Figure 1.1(b): Estimated Gaussian over True Exponential distribution for dataset B

dataset B had a true exponential distribution with $\lambda = 1$. As seen in figure 1.1(b), the estimated Gaussian distribution is not a good representation of the data. This highlights the shortcoming of the Maximum likelihood Estimation method which requires for the probability distribution to be known at the start. If this assumption is incorrect, the results can be completely incorrect.

1.2 Exponential Parametric Estimation

The form of the exponential PDF is given as

$$F_{Exp}(x) = \lambda e^{-\lambda x}$$

For an exponential distribution, the parameters are set up as $\bar{\theta} = [\theta_1] = [\lambda]$. The log function is given as

$$l(\bar{\theta}) = log[\sum_{i=1}^{N} \theta_1 e^{-\theta_1 x}] .$$

Using the generalized approach above, the parameter estimation is given as

$$\widehat{\theta}_{1ML} = \frac{N}{\sum_{i=1}^{N} x_i}$$

where N is the number of samples.

The exponential parametric estimation for each dataset is shown below in figures 1.2(a) and (b). It can be seen that variable B estimation fits the true distribution as expected, whereas variable a, which is truly Gaussian, does not.

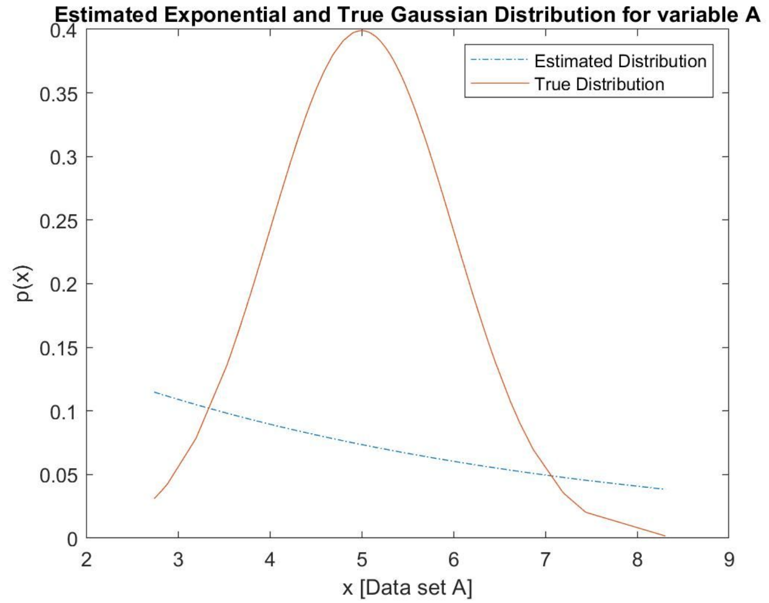

Figure 1.2(a): Estimated Exponential on True Gaussian distribution for dataset A

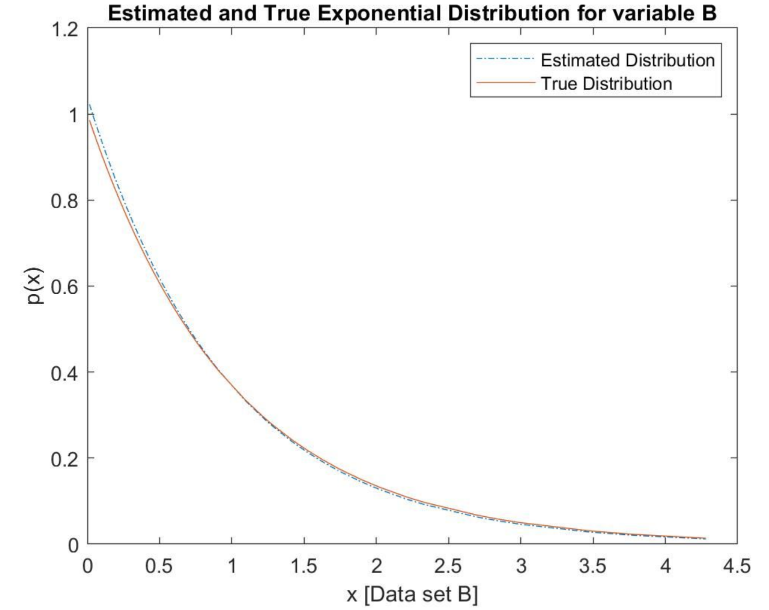

Figure 1.2(b): Estimated and True Exponential distribution for dataset B

There can be seen a slight offset for the Estimated distribution for dataset B compared to the true. This, again, is attributed to the noise in the dataset and the outliers.

1.3 Uniform Parametric Estimation

The form of the uniform PDF is given as

$$F_{Exp}(x) = \{\frac{1}{b-a}\} \quad a \le x \le b$$

The PDF yields zero outside the bounds of a and b. The parameters are setup as $\overline{\theta} = [\theta_1\ \theta_2]^T = [a\ b]^T$. The log function is given as

$$l(\overline{\theta}) = log[\sum_{i=1}^{N} \frac{1}{\theta_2 - \theta_1}].$$

Using the generalized approach above, the parameter estimation is given as

$$\widehat{\theta}_{1ML} = min(x)$$
$$\widehat{\theta}_{2ML} = max(x)$$

Figures 1.3(a) and (b) illustrate the uniform parametric estimation for datasets A and B respectively.

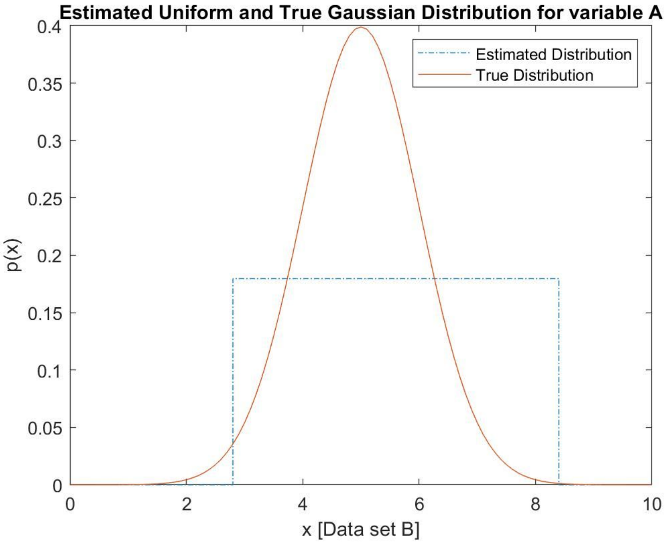

Figure 1.3(a): Estimated Uniform on True Gaussian for dataset A

It can be seen that the dataset above has a Gaussian distribution and thus, does not fit well with the uniform parameters. There are large areas where the uniform distribution gives a high pdf while the Gaussian shows low likelihood of the value occurring. One such value would be x = 8.2. While x = 5 has a higher probability than x = 8.2, as seen from the Gaussian distribution, the uniform estimation would return the same p(x) for both.

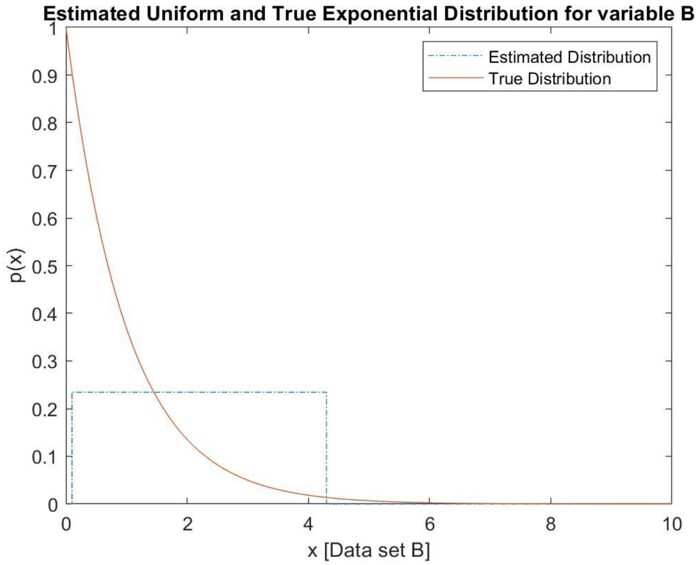

Figure 1.3(b): Estimated Uniform on True Exponential for dataset B

Similar to the previous dataset, the true distribution for dataset B is not uniform. Therefore, fitting the uniform parameters give a distribution very far from the true distribution, as seen in Figure 1.3(b).

It is possible to use a parametric approach in most cases, but it may not be wise. Specifically, it is better to use a parametric method when distribution of the dataset is well known (Gaussian, Exponential, etc). On the contrary, the non-parametric approach is preferred when the shape of the dataset is not well known and there are ample data points.

1.4 Non-Parametric Estimation

Unlike Parametric estimation, non-parametric estimation does not require the functional form of the PDF to be known. Instead, the approach directly estimates the class distribution p(x) from the samples available.

The lab requires the parzen window method be used for estimation. The parzen window estimation p(x) is:

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

Where N is the number of samples, h is the window width and $\phi(\frac{x-x_i}{h})$ is the window function.

If a sample, x_i, is found at a point, the PDF at that location cannot be too low. If multiple samples are found close to each other, the PDF would be higher. The window function accounts of this and h defines the width of the window function - the larger the h, the more samples would be considered for PDF at a point. If h is too small, potentially only one sample would fall within the window and be considered for the PDF - the overall y resolution would decrease.

Usually, h is chosen as follows:

$$h = \frac{k}{\sqrt{N}}$$

Where k is user defined and N is the number of samples.

Using Gaussian distribution as the window function, h was chosen as 1 and the standard deviations were used to control the effect the samples had on the PDF at any point - samples further away would contribute less.
The distributed approximated is as follows:

$$\widehat{p}(x) \;=\; \frac{1}{N}\sum_{i}\frac{1}{h}\left(\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\frac{(x-\mu)^2}{(h\sigma)^2}}\right)$$

The results are illustrated in the figures below.

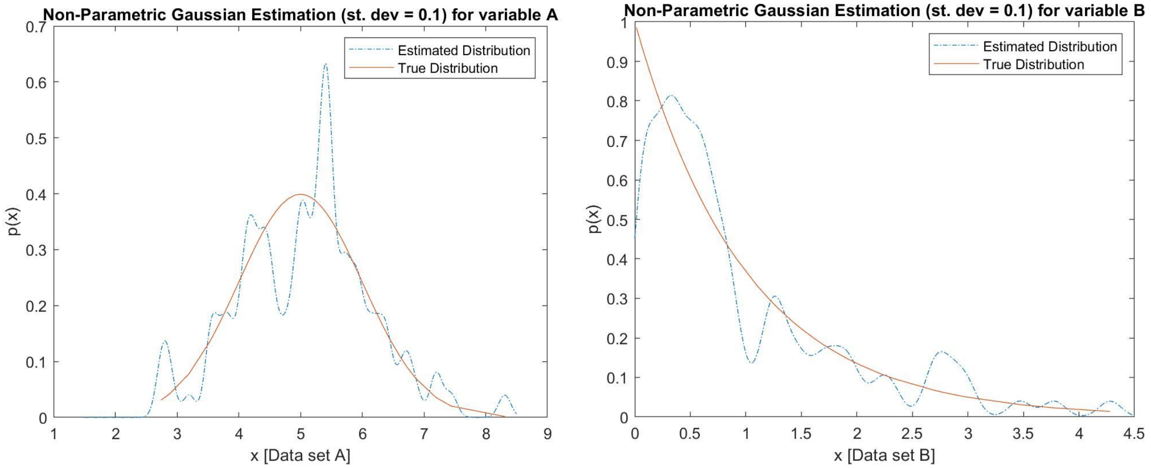

Figure 1.4.1(a) & (b): Non-Parametric Gaussian Estimation for $\sigma = 0.1$ and $h = 1$

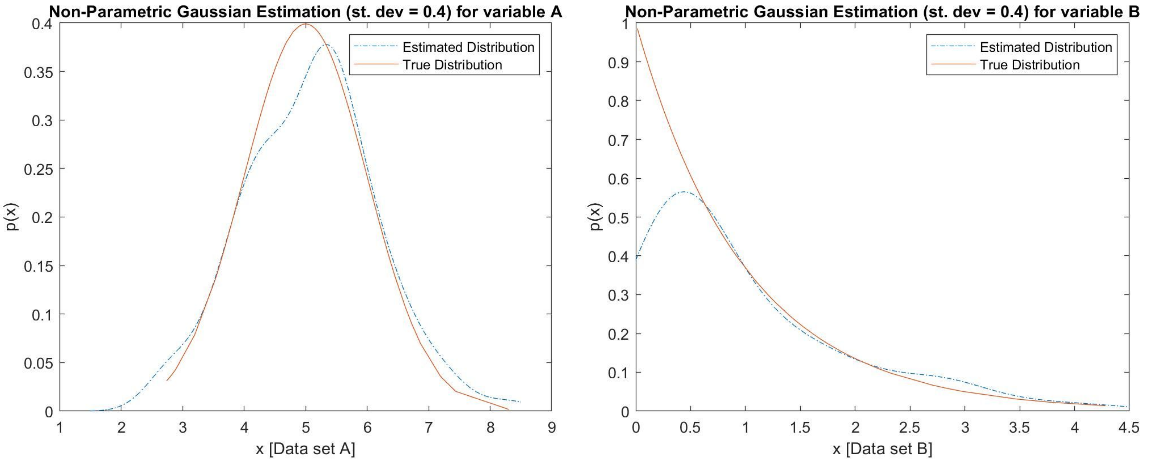

Figure 1.4.2: Non-Parametric Gaussian Estimation for $\sigma = 0.4$ and $h = 1$

Since the distribution is being estimated directly from the data, it can be seen that the overall trend of the estimation follows the true distribution for both datasets A and B. It can also be seen that with the h set to one, the standard deviation has a similar effect - a larger standard deviation leads to a larger window with more data points contributing to the PDF at any given point. Thus, figure 1.4.2 shows smoother estimated pdfs than figure 1.4.1.

This was further verified by choosing $k = 25$ value, which gives $h = 2.5$:

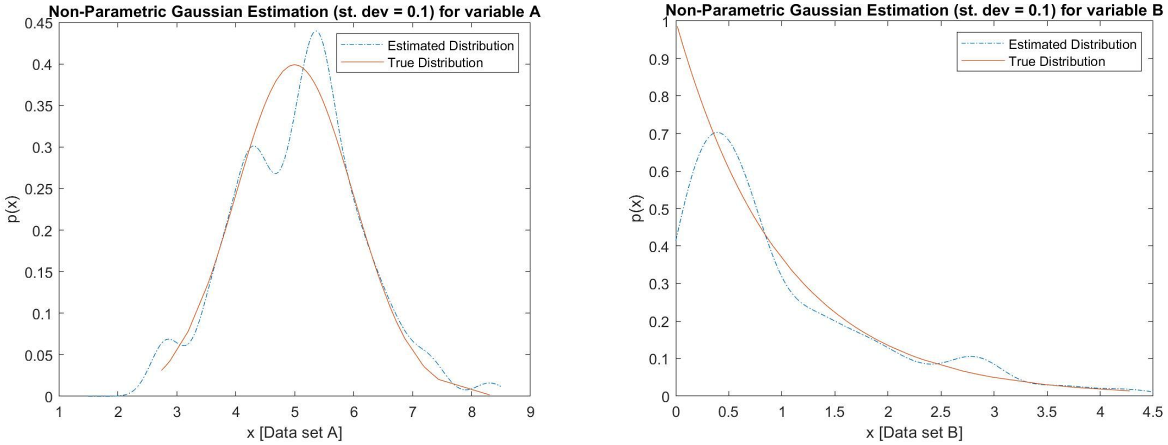

Figure 1.4.3(a) & (b): Non-parametric Gaussian estimation with $\sigma = 0.1$ and $h = 2.5$

Comparing figure 1.4.1 and 1.4.3, the latter is much smoother but also has smaller maximum p(x) values. For example, while the maximum estimated p(x) seen in figure 1.4.1(a) is beyond 0.6, the maximum in 1.4.3(a) falls short of 0.45!
Therefore, as the resolution on the x axis is increased, it decreases on y axis, as expected.

In conclusion, the effectiveness of the parametric or non-parametric approach depends heavily on the information available regarding the dataset. If the distribution for the dataset is known, the parametric approach performs better as seen in Figure 1.1(a) and figure 1.2(b). In both of these cases, the parametric approach was used with the correct distribution of the data and the estimated pdfs were almost identical to the true distributions of the data.
However, as seen in figure 1.1(b), 1.2(a), and 1.3(a) and (b), there is a high cost associated with using the parametric approach with incorrect distributions. In such cases, it is better to use the non-parametric approach with carefully thought out parameters as seen in figures 1.4.1, 1.4.2 and 1.4.3.

2. Model Estimation 2-D Case

This part of the lab deals with parametric and non-parametric estimation for two dimensional cases. The data used for this section comes from the course website, a file named “lab2_2.mat” as mentioned in the lab manual.

The data consists of three clusters of which two, al and cl, are of Gaussian nature while the third, bl, is not. The purpose of this section is to use different estimation methods to find decision boundaries between the classes and identify which type of estimation - parametric or otherwise - works better.

2.1 Parametric Estimation

Parametric Estimation for the 2 dimensional case is identical to that of the one dimensional case. It assumes that the distribution of the dataset is known and then attempts to estimate the distribution parameters.

As outlined in the lab, the clusters of data are assumed to be normally distributed. Thus, the PDF function becomes:

$$p(\bar{x}) \ = \ \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{0.5}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu})^T \Sigma^{-1} (\bar{x}-\bar{\mu})}$$

Where n is the number of variables (2 in this case) and $\bar{x}$ is the input vector.

For a Gaussian distribution, the parameters are setup as $\bar{\theta} = [\theta_1 \, \theta_2]^T = [\bar{\mu} \, \Sigma^{-1}]^T$.

Using the approach as shown in the course notes, the parameter estimations can be determined from the samples as

$$\theta_1 \ = \hat{\mu}_{ML} \ = \ \frac{1}{N} \sum_{i=1}^{N} \bar{x}_i \ ,$$

$$\theta_2^{-1} \ = \ \hat{\Sigma}_{ML} = \ \frac{1}{N} \sum_{i=1}^{N} (\bar{x}_i - \hat{\mu}_{ML}) \, (\bar{x}_i - \hat{\mu}_{ML})^T$$

where N is the total number of samples in the dataset.

It was decided to use an iterative approach to find the ML classification boundaries as was done in Lab 1.

For ML classification:

$$\bar{x} \ \varepsilon \ A \ iff \ p(\bar{x}|A) \ > \ p(\bar{x}|B)$$

Thus, generalizing further, $\bar{x}$ belongs to the class with the highest p(x) for that class.

The above equations were used to approximate the mean and covariances of each cluster in the dataset. Then, for each (x,y) value on the grid, a p(x) was calculated for each of the three classes. The grid point was assigned to the class which scored the highest p(x). Each classification was stored in the matrix and by running edge detection on the matrix, the classification boundaries were found. The classification boundaries can be seen in figure 2.1.

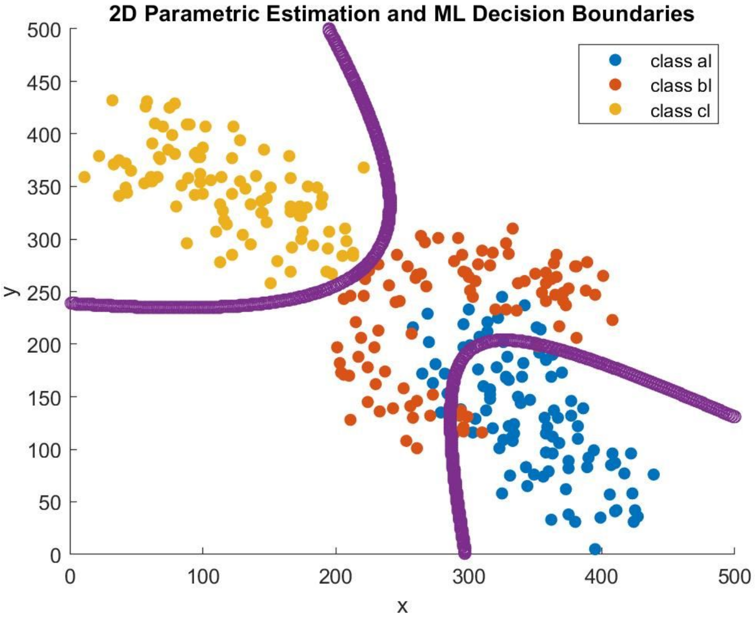

Figure 2.1: 2D Parametric Estimation and ML classification boundaries

As can be seen from the figure, the classification boundaries are very accurate around classes al and cl which are true Gaussian distributions. The boundaries are unable to identify class bl well due to the assumption it is normally distributed though it is clearly not.
This is the weakness of Parametric estimation - the correct distribution must be known.

2.2 Non-Parametric Estimation

For non-parametric class estimation, the parzen window technique is used, as learned in class. Specifically, the window is used with a variance, $\sigma^2 = 400$ and a mean centred at 200 for both the x and y values. With the help of a parzen window function given from the course website, the implementation was fairly simple:

a) Determine the resolution of the grid, which is essentially the minimum value of all classes in the x and y direction to the maximum value of all lasses in the x and y direction. The resolution will be the minimum value to the maximum value with a step size of 1.
b) Develop a Gaussian window using the MATLAB function, MVNPDF, which generates a multivariate pdf in matrix form using a mean and covariance as inputs. The covariance is 2x2 matrix, and as mentioned above, 400 in the diagonals, zero for non-diagonal elements.
c) Use the parzen function as given in the course website: parzen(data, resolution, window)

d) Create contours of the pdfs and overlay the testing data
e) Create a new grid that spans the length of the x and y axis of the estimated pdfs and evaluate each of the pdfs at each (x,y) point in the grid, assign the matrix location (x,y) to be the class with the highest probability (as per the maximum likelihood classifier):
   Class A = 1, class B = 2, class C = 3;

In figure 2.2.1 below, the contour plot indicates that each estimated pdf using the parzen window is reasonable.

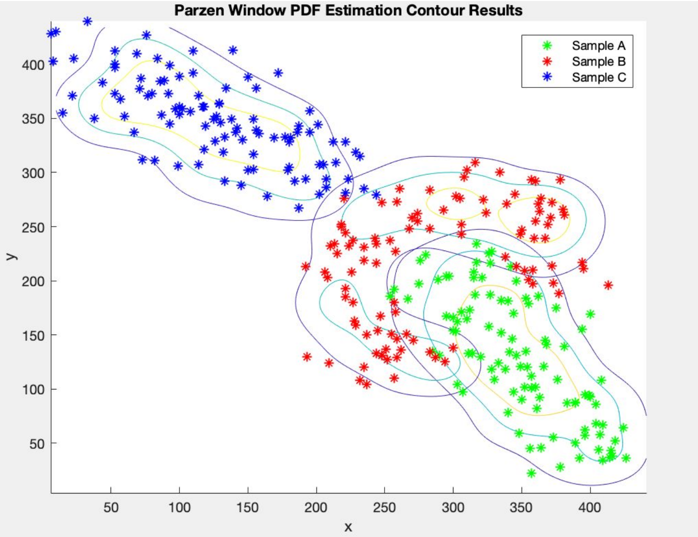

Figure 2.2.1: Parzen Window PDF estimation contour plots

The results of the ML classifier using the estimated PDFs is shown below in figure 2.2.2**.**

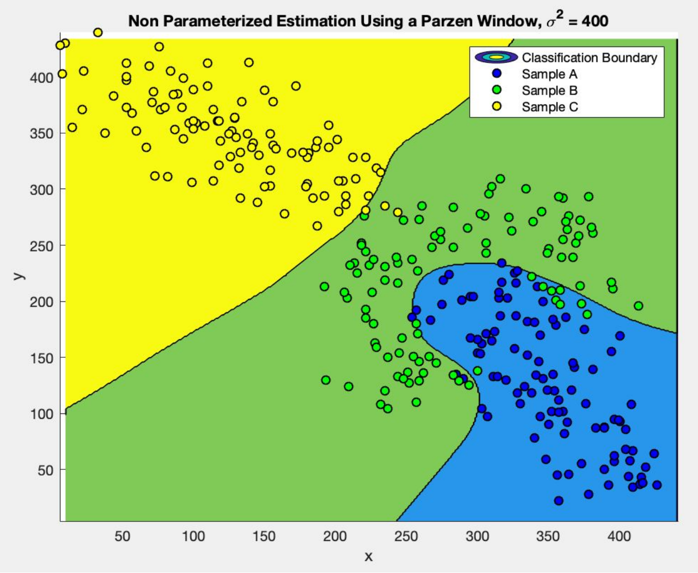

Figure 2.2.2: Non-parametric estimation with Parzen Window $\sigma^2 = 400$

As mentioned above, it is preferred to use a non-parametric approach when the distribution of the dataset is not known and there are enough data points. Comparing from the results above, the nonparametric estimation is fairly accurate, but a trial and error process is used to determine the appropriate window function and its properties; after tuning for a little while, it can be seen that without having the true densities of each dataset, great separability between classes was achieved using non parametric estimation. Analytically, above, about 16 out of the 300 test data points were misclassified, yielding about 95% accuracy. The largest error came from the tendril-like cluster (green).

Overall, it is always better to use the non-parametric approach unless the distribution of all the clusters in the dataset are known. Comparing figures 2.1 and 2.2.2, it is clear that the non-parametric approach has better classified the data samples into their respective classes. The parametric approach may have performed well had the class bl also been normally distributed, as it was assumed to be in the calculations.

3. Sequential Discriminants

3.1 Implementation

Sequential classification is a stochastic method of machine learning that uses multiple low cost discriminants to achieve high classification accuracy. The Mean Euclidean Distance based discriminant has been used in this experiment.

Learning

The following algorithm is used to learn various discriminants on a given set of data.

1. Let a and b represent the data points in classes A and B. Let j = 1.
2. Randomly select one point from a and one point from b
3. Create a discriminant G using MED with the two points as prototypes
4. Using all of the data in a and b, work out the confusion matrix entries
    a. n_{aB} = #times G classifies a point from a as class B
    b. n_{bA} = #times G classifies a point from b as class A
5. If $n_{aB} \neq 0$ and $n_{Ba} \neq 0$ then no good, go back to step 2.
6. This discriminant is good; save it as $G_j = G$, $n_{aB,j} = n_{aB}$, $n_{bA,j} = n_{bA}$ Let j = j + 1.
7. If $n_{aB} = 0$ then remove those points from b that G classifies as B.
8. If $n_{bA} = 0$ then remove those points from a that G classifies as A.
9. If a and b still contain points, go back to step 2.

Classification

The following algorithm is used for classifying any points using the learnt set of discriminants

1. Let j = 1
2. If G_j classifies x as class B and $n_{aB,j} = 0$ then “Say Class B”
3. If G_j classifies x as class A and $n_{bA,j} = 0$ then “Say Class A”
4. Otherwise j = j + 1 and go back to step 2.

3.2 Results

This section illustrates in the results of training a sequential classifier on a given set of 2D data. Figure 3.1 shows the results of the sequential classification algorithm, where the said classification boundary has been trained and tested in the same data. In the following cases, the sequential classification algorithm was run till completion.

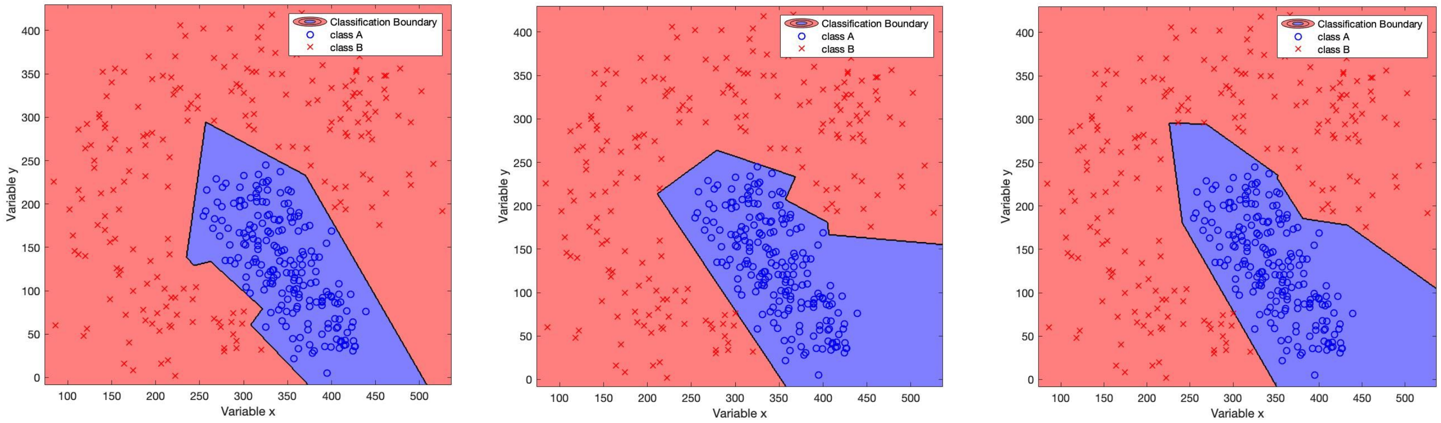

Figure 3.1: Sequential Classification boundaries with different random initializations (trained and overlaid on the same training data)

It can be seen that the plots shown in figure 3.1, where the discriminants have been trained until they converge, lack any unclassified regions. Moreover, the classified regions appear to be completely accurate, i.e., none of the training points have been incorrectly classified. This leads to the conclusion that the probability of error is zero for a sequential classifier with no limit on the number of discriminants. Convergence of such a sequential classifier implies a 100% accuracy on the training data. Clearly, the reason for these characteristics is the ability to use infinite MED discriminants.

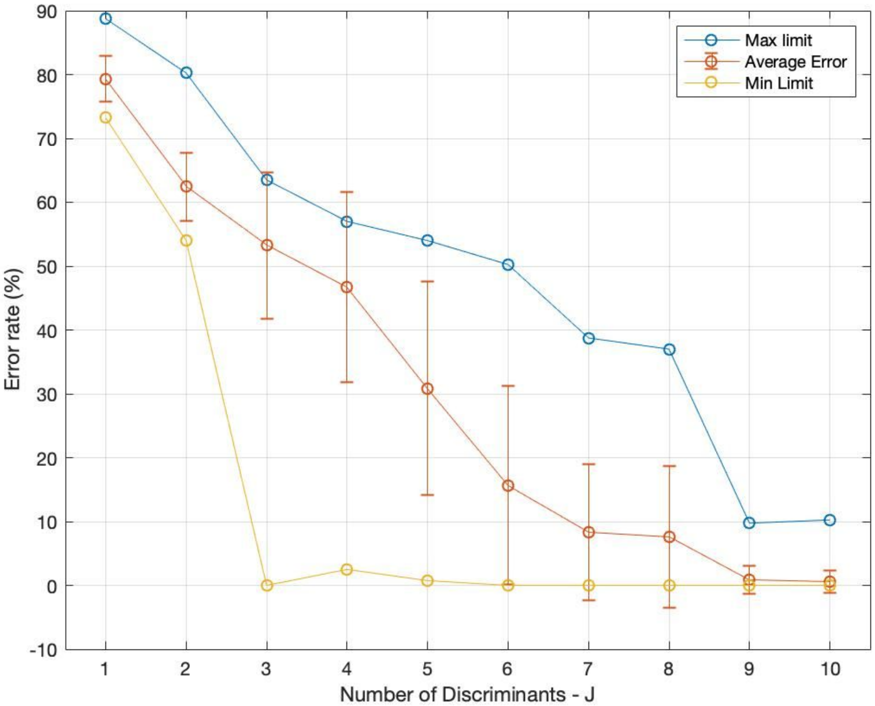

Figure 3.2: Error rate vs Number of discriminants used for sequential classification (trained and tested on the same data 20 times for each J)

In the case where the number of discriminants have been the limited, the error rate are much higher. Figure 3.2 shows the results of training and classifying on the above data using the sequential classification algorithm, while limiting the number of MED discriminants (J) to a finite value. The results show an inverse relationship between the error rate and number of discriminants, i.e., the error rate falls drastically as the number of discriminants are increased. The error rate also seems to plateau close to the higher values of J, indicating a possible saturation if the J is increased further. The maximum and the minimum error rates appear to follow the same trend. On the other hand, the standard deviation of the error rates follows a peculiar trend; it increases from J = 1 to J = 5 and then decreases to a very low values as the J gets higher.

In the sequential classification algorithm the discriminant searching process is stochastic, which implies that limiting the number of discriminants does not guarantee a limit on the number of MED classifiers tested. Figure 3.3 shows the number of MED iterations vs the number of discriminants used. The result is coherent with the intuition of higher MED iteration for more discriminants. If the number of MED iteration were limited, it would establish an indeterminate cap on the number of discriminants that can be used. It would become very hard to predict the accuracy or the error rate of the sequential classification since the there is a relationship between average error rates and the number of discriminants - J.

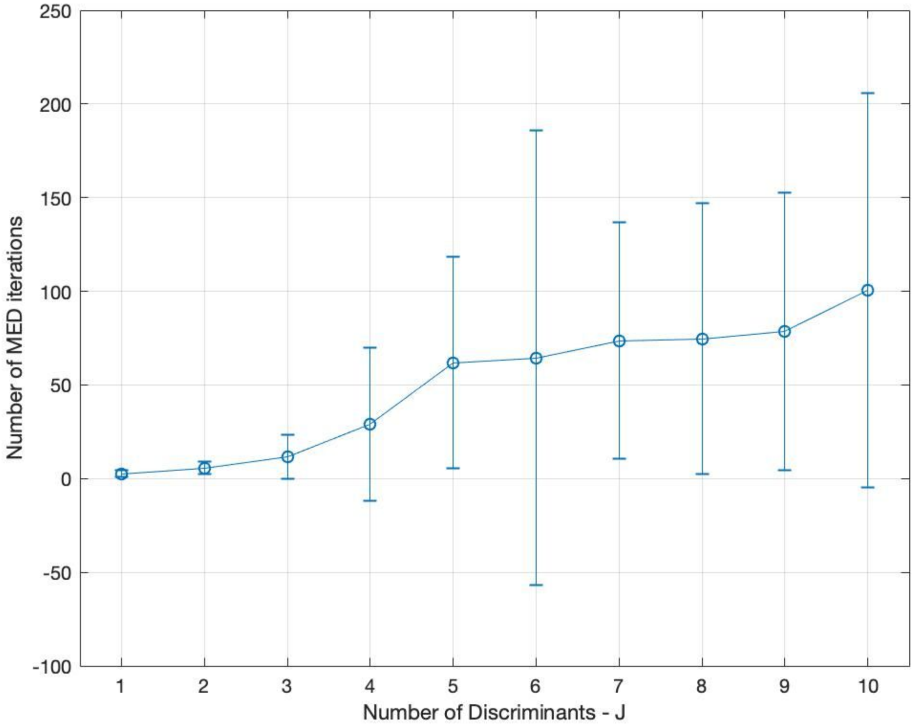

Figure 3.3: Average number of MED iteration vs Number of discriminants-J

4. Conclusions

The first half of the lab was concerned with comparing the outputs of parametric and non-parametric estimation approaches on different data sets. For the 1D case, it can be concluded that the parametric approach is preferred when the overall PDF of the dataset is known and only the parameters of the PDF need to be estimated. The non-parametric approach performs better when the PDF is not known or when the data does not conform to a parametric distribution at all. In the 2D case, non-parametric estimation performs better when the data does not conform to a PDF or when the PDFs for all the data sets are not known - that is to say, in cases where the distribution must be estimated directly from the data. However, there are user defined variables in non-parametric estimation which can significantly change the resultant PDF. Thus, it is important to choose those carefully.

The second half of the lab was concerned with the implementation and the evaluation of the sequential classification approach. This approach uses a combination of multiple Euclidean classifiers to achieve high accuracy. It is found that the accuracy of classification for a sequential classifier is dependent on the number of discriminants used. The accuracy appears to increase as the number of discriminants increase.