# SYDE 572 - Lab 2

Model Estimation and Discriminant Functions

**Group 19**

| | |
|---|---|
| Jonathan Chen | 20722167 |
| Ellen Choi | 20707132 |
| Aman Mathur | 20710307 |
| Michael Eden | 20678312 |

April 5, 2022

## 1 Introduction

Lab 2 of SYDE 572 investigates statistical model estimation using parametric and non-parametric estimators and classifier aggregation by combining multiple linear discriminants into one classifier. In the first section of the lab, the provided dataset *lab2_1.mat* was used to plot the estimated probability density function using a Gaussian, exponential, and uniform distribution, as well as using the Parzen method, to compare the estimated PDF against the true PDF. In the second part, the *lab2_2.mat* dataset was used to create parametric and non-parametric estimations of each cluster in the dataset, which were then used to create ML classification boundaries. Finally, in the last part of the lab, the *lab2_3.mat* dataset was used to develop sequential classifiers using a discriminant.

## 2 Model Estimation 1-D Case

For all the parametric estimations, the parameters of each distribution (Gaussian, Exponential, and Uniform) were determined using Maximum Likelihood estimation.

### 2.1 Parametric Estimation - Gaussian

For the Gaussian distribution, there were two parameters to estimate: $\mu$ and $\sigma$. From our class notes, we need to learn the values for $\theta_1$ (which represents the mean) and $\theta_2$ (which represents the variance) from the following equation:

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

After taking the log, taking the derivative, setting it to 0, and solving, the values are:

$$\hat{\theta}_1 = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\hat{\theta}_2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{\theta}_1)$$

These values are the sample mean and sample variance of the dataset. Both of these estimates are implemented in *estimateGaussianParams.m* and were calculated to be $\mu_a = 5.0763$, $\sigma_a^2 = 1.1274$, $\mu_b = 0.9633$, and $\sigma_b^2 = 0.8643$. To create the Gaussian probability distribution function, the calculated values for these parameters were passed into the equation for a Gaussian in *calculateGaussianPDF.m*.

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2)$$

Figure 1 shows the estimated and true PDFs for datasets a and b using Gaussian parametric estimation.

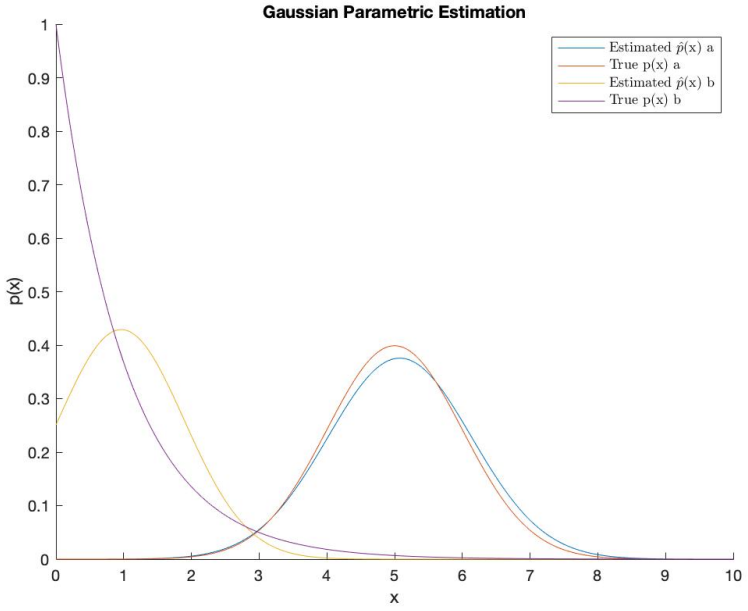



Figure 1: Gaussian Parametric Estimation for datasets a and b

## 2.2 Parametric Estimation - Exponential

For the Exponential distribution, there was one parameter to estimate: $\lambda$. We need to learn the value for $\theta$ (which represents the value for lambda) from the following equation:

$$p(x_i|\theta) = \begin{cases} \prod_{i=1}^{N} \theta \exp(-\theta x_i) & x_i \geq 0 \\ 0 & otherwise \end{cases}$$

Solving for theta gives:

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

This estimate is implemented in *estimateExponentialParams.m* and the values for $\lambda$ were calculated to be $\lambda_a = 0.1970$ and $\lambda_b = 1.0381$. To create the exponential probability distribution function, the calculated values for these parameters were passed into the equation for an exponential in *calculateExponentialPDF.m*.

$$p(x) = \begin{cases} \lambda \exp(-\lambda x) & x \geq 0 \\ 0 & otherwise \end{cases}$$

Figure 2 shows the estimated and true PDFs for datasets a and b using exponential parametric estimation.

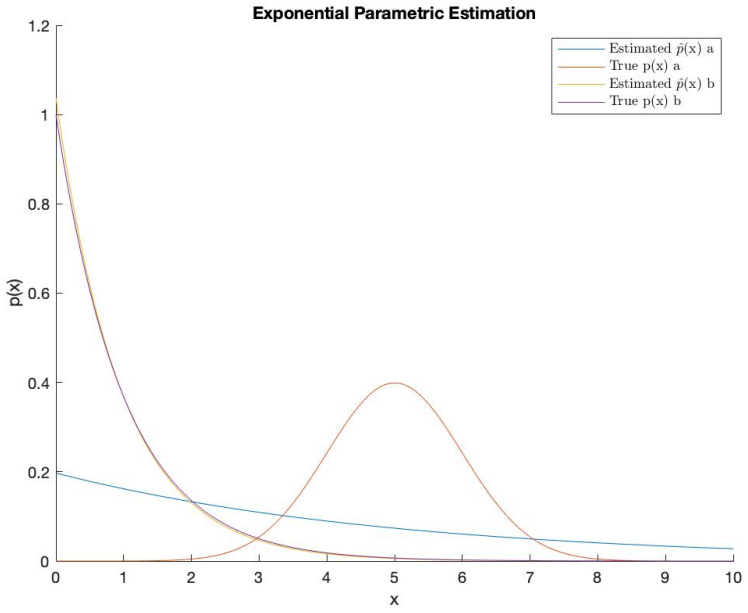



Figure 2: Exponential Parametric Estimation for datasets a and b

## 2.3 Parametric Estimation - Uniform

For the Uniform distribution, there were two parameters to estimate: $a$ and $b$. We need to learn the values for $\theta_1$ (which represents the value for $a$) and $\theta_2$ (which represents the value for $b$) from the following equation:

$$p(x_i|\underline{\theta}) = \begin{cases} \prod_{i=1}^{N} \frac{1}{\theta_2 - \theta_1} & \theta_1 \leq x \leq \theta_2 \\ 0 & otherwise \end{cases}$$

This estimate is implemented in *estimateUniformParams.m* and the values for $a$ and $b$ were calculated to be $a_a = 2.7406$, $b_a = 8.3079$, $a_b = 0.0143$ and $b_b = 4.2802$. To create the uniform probability distribution function, the calculated values for these parameters were passed into the equation for a uniform distribution in *calculateUniformPDF.m*.

$$p(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & otherwise \end{cases}$$

Figure 3 shows the estimated and true PDFs for datasets a and b using uniform parametric estimation.

## 2.4 Non-parametric Estimation

For the non-parametric estimations, the PDFs were estimated using one-dimensional Parzen estimates with a Gaussian window function. For each sample point, all the Gaussian distributions for each data point (in either a or b) were summed up and divided by N (the number of data points). The following equation was implemented in *parzen1D.m*:

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(x_{data_i} - x_{sample}))$$

Figures 4 and 5 show the estimated and true PDFs for datasets a and b using Parzen estimates using Gaussian windows with standard deviations of 0.1 and 0.4, respectively.

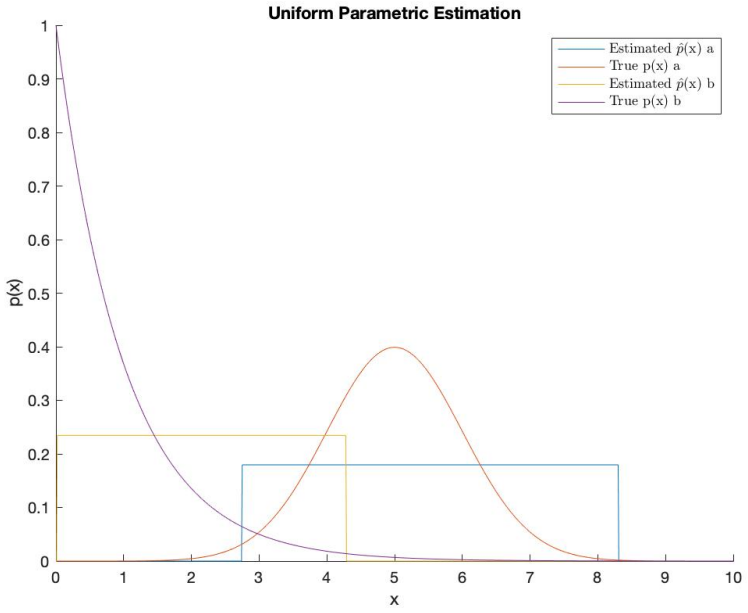



Figure 3: Uniform Parametric Estimation for datasets a and b

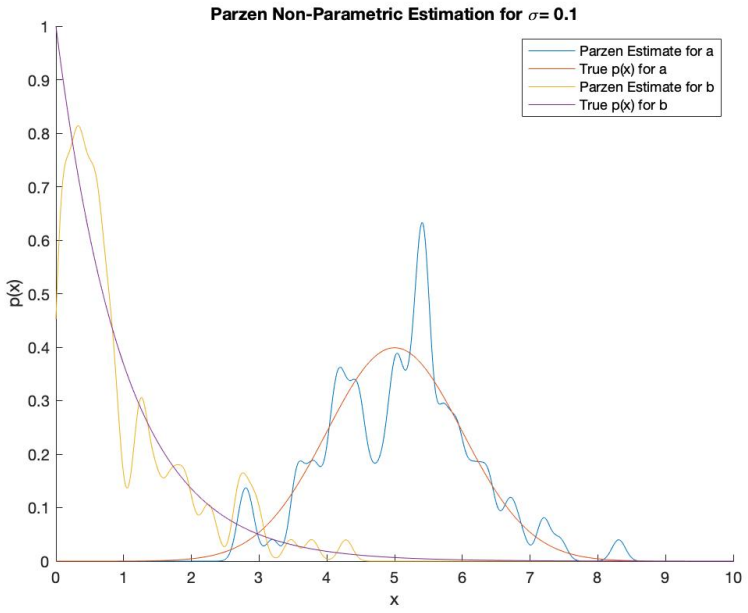



Figure 4: Parzen Estimation for datasets a and b using $\sigma = 0.1$

***For each of the two data sets, which of the estimated densities is closest to the original? Give a qualitative comparison of the results.***

Based on the figures for the PDF estimations of the different assumptions for types of distributions, the parametric estimation assuming a Gaussian distribution most closely resembles the original distribution of data set $a$, while the parametric estimation assuming an exponential distribution most closely resembles the original distribution of data set $b$. In particular, the blue curve for the Gaussian estimate of data set $a$ in

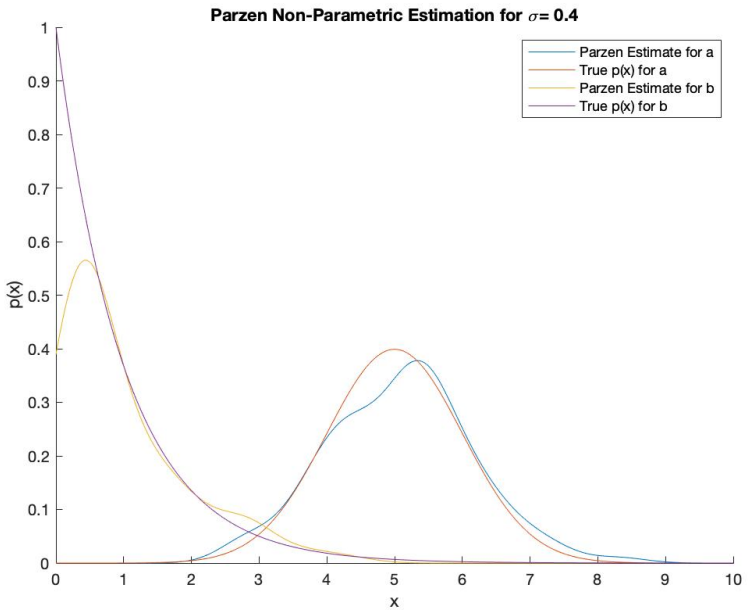



Figure 5: Parzen Estimation for datasets a and b using $\sigma = 0.4$

Figure 1 is very close to the true distribution (orange curve), while the yellow and purple curves in Figure 2 show that the exponential estimated density is almost identical to the original. This makes sense given that the original/true distributions for $a$ and $b$ are known to be a Gaussian and exponential, respectively. Figure 3 shows that the uniform distribution does not do a good job of approximating either data set and Figures 4 and 5 show that the Parzen estimates do a relatively good job of approximating the general shape of each dataset, where the estimate using a Gaussian window with standard deviation of 0.4 is slightly more accurate.

***In general, is it possible to always use a parametric approach? When is it better to use a parametric method? When is the non-parametric approach preferred?***

In general, it is always possible to use a parametric approach to determine the estimates for the parameters of an assumed distribution, but the parametric approach may not always be accurate. Since the parametric approach involves an assumption of the functional form of the PDF in order to determine what parameters to estimate, the parametric approach may yield very inaccurate results if the assumption is incorrect, as evidenced in Figures 1, 2, and 3. On the other hand, with no prior assumption for the data, the non-parametric approach (Figures 4 and 5) created fairly accurate estimates of the PDFs of each data set. Therefore, it is better to use a parametric method when the true distribution of the dataset is already known or can be reasonably assumed, while the non-parametric approach is preferred when the the functional form of the PDF is not known or the statistical distribution cannot be well modeled using parametric PDF models.

# 3 Model Estimation 2-D Case

For Part 3, decision boundaries were generated for 3 classes. Both parametric and non-parametric estimation methods were performed.

## 3.1 Parametric Estimation

For the parametric estimation method, it was assumed that each cluster is normally distributed. To create the Maximum Likelihood classification boundaries, a modified version of the *MAP3.m* function from Lab 1 was used where the probability of each prior distribution is assumed to be equal. In order to create the ML classifier, the sample mean and covariance found for each cluster were found to be:

$$\bar{x}_{al} = \begin{bmatrix} 347.16 \\ 131.2 \end{bmatrix}, S_{al} = \begin{bmatrix} 1766.6 & -1610.6 \\ -1610.6 & 3343.5 \end{bmatrix}$$

$$\bar{x}_{bl} = \begin{bmatrix} 291.84 \\ 224.02 \end{bmatrix}, S_{bl} = \begin{bmatrix} 3315.7 & 1176 \\ 1176 & 3414 \end{bmatrix}$$

$$\bar{x}_{cl} = \begin{bmatrix} 119.55 \\ 346.67 \end{bmatrix}, S_{cl} = \begin{bmatrix} 2738.5 & -1327.2 \\ -1327.2 & 1699.3 \end{bmatrix}$$

Figure 6 shows the 2D parametric estimation for the maximum likelihood classification boundaries.

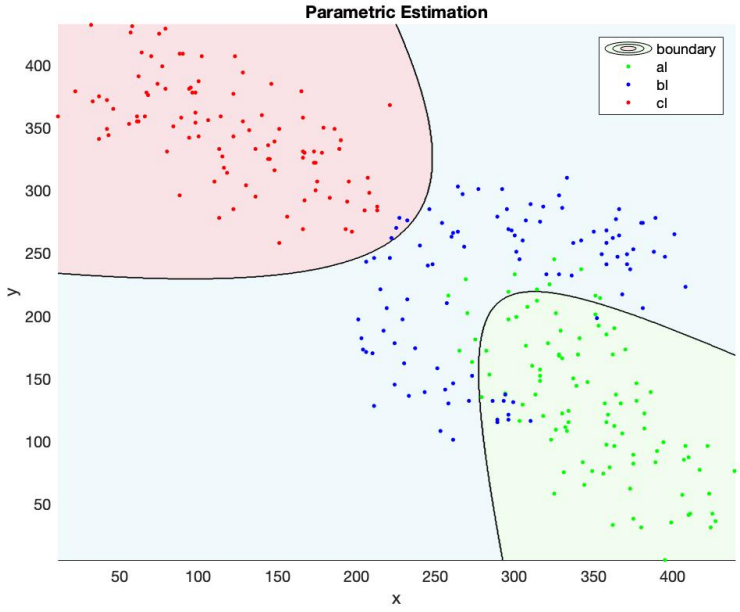



Figure 6: Parametric Estimation for datasets al, bl, cl

## 3.2 Non-parametric Estimation

To create the non-parametric estimations, the provided *parzen.m* function was used to construct the probability density functions for each class. A Gaussian Parzen window with variance of $\sigma^2 = 400$ was used. To create the Parzen window, Gaussian estimates were calculated at x values across the entire range of the data. The step size between each x sample was $\pm resolution$ with the overall kernel size being $2n_c + 1$, where $n_c$ is the number of cells in each direction. From a trial and error approach, $n_c$ was set to $12 * h/resolution = 480$ to ensure that there was at least one class defined at each coordinate in the range of values, leading to a smooth decision boundary. Once the PDFs for each cluster were obtained from the provided Parzen function, an ML classifier was used to determine the decision boundaries. Figure 7 shows the 2D non-parametric estimation for the classification boundaries.

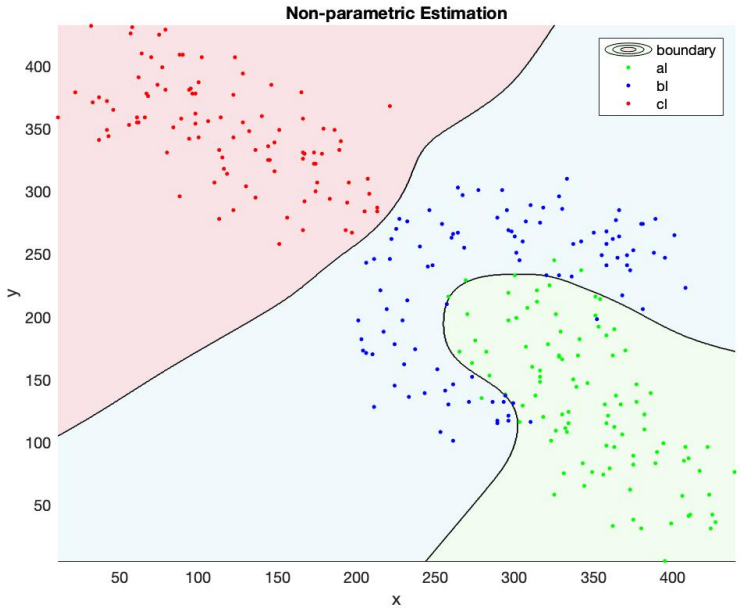



Figure 7: Non-parametric Estimation for datasets al, bl, cl, $\sigma^2 = 400$

### 3.3 Comparison

***Give a qualitative comparison of the classification results. In general, is it possible to always use a parametric approach? When is it better to use a parametric method? When is the non-parametric approach preferred?***

From analyzing Figures 6 and 7, we can compare the parametric and non-parametric estimation approaches. The parametric approach in Figure 6 does a reasonable job of estimating the boundaries but has a group of samples from *bl* that are misclassified within the *al* bound and another group of samples from *al* that are likewise misclassified within the *bl* bound. It is evident that Figure 7 presents better fitting decision boundaries given our sample data. Particularly for points in *al* represented by the green area on the plot, the decision boundary curves around data points to capture more *al* points within the boundary.

The two main categories of statistical learning, parametric and non-parametric estimation, are both effective in certain instances. In parametric estimation, the form of the PDF is assumed and the necessary parameters are estimated. In non-parametric estimation, the PDF itself is estimated directly. In many pattern recognition problems, the functional form of the PDF may not be known, making parametric estimation difficult. When data is not normally distributed or the underlying distribution of the data is not known, it may be more effective to use a non-parametric method. However, parametric methods are preferable when assumptions can be made about the data because they are simpler to compute.

## 4 Sequential Discriminants

### 4.1 Implementation

The sequential classifier was implemented as described in the lab manual. During the learning process, one random point from each class is selected, and an MED classifier is constructed using those points as prototypes. The resulting classifier tested against all points to determine whether it classifies some part of the problem perfectly (i.e., for at least one class, no points belonging to other classes are incorrectly classified

into that given class). If so, the discriminant is saved by storing the two prototypes, and the points correctly classified into the the class corresponding to the discriminant are removed. The process is repeated until all points have been removed, or the desired number of iterations has been reached.

For a given point $\vec{x}$, the inference process is sequential. For each discriminant, if the discriminant classifies $\vec{x}$ into a class into which no other points were incorrectly classified during the learning process, then the classification is obtained and the process terminates. Otherwise, the process repeats for the subsequent discriminant. In the case where $J$ is limited, the discriminants may be exhausted before a definitive classification has been made. In such a case, the classification defers to that made by the final discriminant.

Three sequential classifiers were generated with no limit imposed on the number of iterations. The resulting decision boundaries of each classifier are depicted in 8.

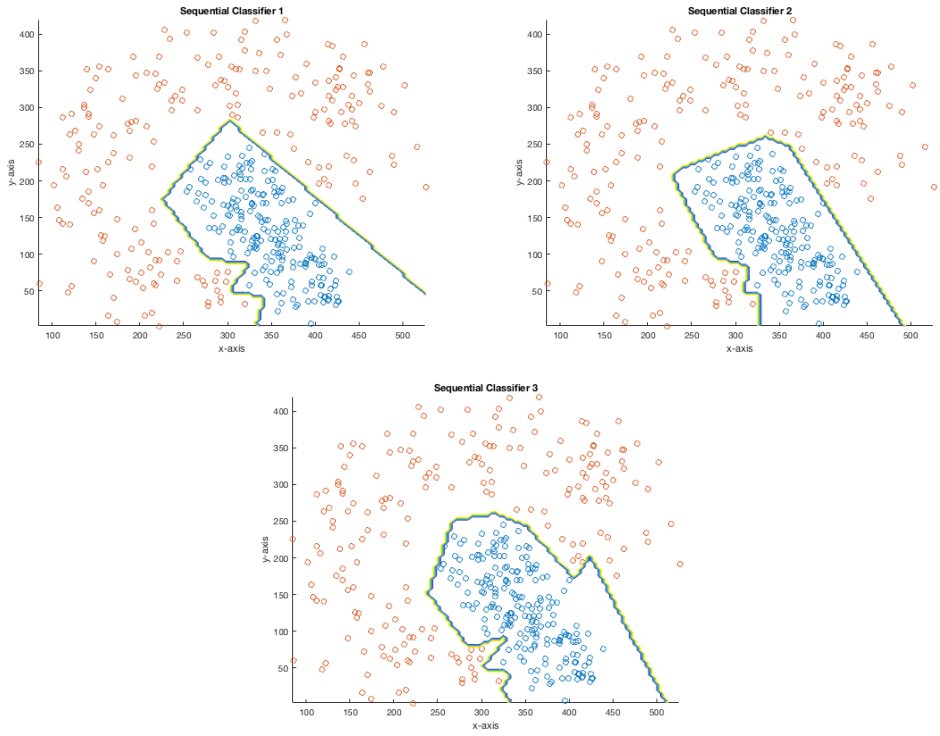



Figure 8: Sequential classifiers with no limit imposed on $J$

## 4.2 Error Analysis

***If we test our classifier on the training data, what will its probability of error be? Discuss.***

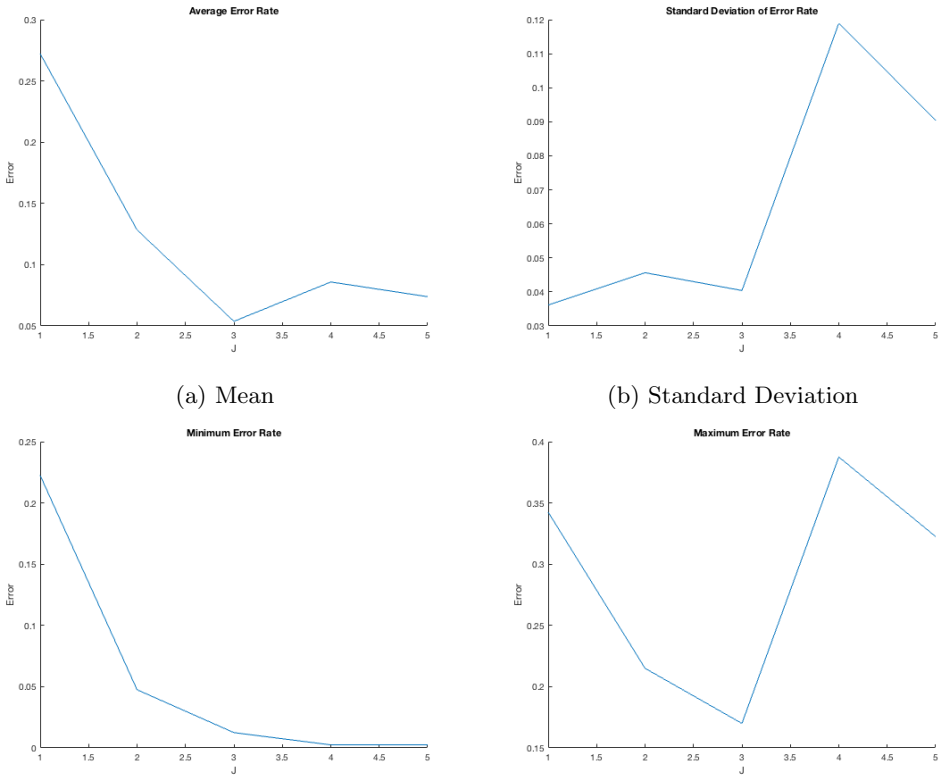
If tested on training data, the probability of error will be 0.

Learning a sequential classifier involves computing a series of discriminants each of which classifies some part of the problem perfectly. A discriminant is saved if it correctly predicts all points for at least one class, after which these correctly classified points are removed from the learning process for subsequent iterations. The process is repeated until all points are correctly classified.

Therefore, when the sequential classifier is tested on its training data, it is guaranteed to correctly classify all points.

***Produce plots showing error rates as a function of $J$.***

As shown by 9a, the classifier error tends to decrease as $J$ increases. This result is expected since a greater




(a) Mean

(b) Standard Deviation

(c) Minimum

(d) Maximum

Figure 9: Behaviour of sequential classifier error rate as a function of J

number of discriminants implies that more of the problem is perfectly classified. Similarly, the best-case performance also increases with $J$ as depicted in 9c.

One unexpected result is that the maximum error and standard deviation are both significantly larger for $J = 4, J = 5$. This observation may be related to the behaviour of the classifier that when all discriminants have been exhausted, the classification made by the last discriminant is deferred. It may be possible that the fourth and fifth discriminants tend to yield high error rates if, for example, each one perfectly classifies only a very small part of the problem.

***In our sequential classifier, we assumed that we could keep looking indefinitely for a classifier that would classify elements of some class perfectly. How might the results of the sequential classifier differ if I limited the number of point pairs that you could test?***

If the number of point pairs that could be tested were limited, the sequential classifier would no longer classify the training set perfectly. The error rate would likely increase as the number of point pairs that could be tested decreases, since there is a reduced probability of identifying a discriminant which perfectly classifies some part of the problem. It can be speculated that when evaluating on non-training data, such an implementation may better handle outliers since a discriminant which classifies almost all points correctly, with the exception of that outlier, would be learned.

# 5 Summary and Conclusions

In summary, parametric and non-parametric estimations were made for one dimensional and two dimensional datasets. For the one dimensional case, the parametric estimations did a good job of approximating the

original distribution when the assumed distribution matched the actual distribution of the dataset. For example, the Gaussian parametric estimation closely matched the dataset with Gaussian samples and the exponential parametric estimation closely matched the dataset with exponential samples. However, when the wrong assumption for the density of the samples was used, the parametric estimations were not very accurate. On the other hand, the non-parametric estimation using a Parzen window produced relatively accurate approximations of the underlying distribution without requiring any assumptions. In particular, the Parzen estimate with a Gaussian window using a larger standard deviation was more accurate than the one with a lower standard deviation.

As for the two-dimensional case, visually the non-parametric estimation method appeared to produce a more accurate decision boundary than the parametric method. The non-parametric decision boundary curved to fit the dataset better, so there were fewer misclassified datapoints compared to the parametric method. Nevertheless, the parametric decision boundary was still reasonably accurate.

Finally, sequential discriminants were also calculated to create a sequential classifier. A sequential classifier may overfit to the data on which it is trained, as evidenced by perfect classification results when evaluated on the training set. Therefore, this particular implementation of a sequential classifier may be poorly suited to data sets with outliers. Moreover, when the number of discriminants is bounded to some finite integer $J$, the sequential classifier was found to perform poorly when evaluated on the testing set. As $J$ is increased, the classifier performance generally improves. An alternative implementation of a sequential classifier may involve generating discriminants with a finite number of pairs of points. These sequential classifiers may yield non-zero error-rate when tested on the training data, but they could possibly better handle outliers.