Model Estimation and Discriminant Functions

SYDE 372 - Lab 2

March 18th, 2017

Group Members:	Krishn Ramesh - 20521942
	Brady Kieffer - 20517665
	Ramandeep Farmaha - 20516974
	Shubam Mehta - 20483061
Instructor:	Professor Wong

1 Introduction

As is almost always the case in real-life applications, the model and distribution of a particular dataset are rarely known and cannot simply be used. There are a number of methods to estimate the model including parametric and non-parametric estimators. This lab explores the relative advantages and disadvantage of each model estimation method under different circumstances. The latter half of the lab focuses on classifier aggregation i.e. the practice of using multiple simple classifiers (such as linear discriminants) in unison to produce a more powerful classifier. This is the fundamental idea behind deep learning and this lab shows the power of such aggregation methods.

2 Model Estimation: 1-D Case

Two datasets are used for the first sections of this lab. Both are 1 dimensional and follow a different distribution which is assumed to be unknown. The models for each class are estimated using a variety of parametric estimation methods as well as a non-parametric Parzen approach. The two classes are:

- a: Gaussian samples parameterized by $\mu = 5, \sigma = 1$.
- b: Exponential samples parameterized by $\lambda = 1$.

2.1 Parametric Estimation - Gaussian

The Gaussian parametric estimator can be derived using the Gaussian probabilistic distribution:

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

Where μ and σ^2 are the estimated mean and variance. The estimated mean and variance can be computed using the following formulae:

$$\mu_{est} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{2}$$

$$\sigma_{est} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_{est})^2 \tag{3}$$

Finally, to obtain an unbiased variance one simply needs to multiply the resulting variance by $\frac{N}{N-1}$. By applying the two equations to the datasets provided within *lab2_1.mat* the following means and standard deviations were calculated:

$$\mu_{est\_a} = 5.0763,\ \sigma_{est\_a} = 1.0618$$
$$\mu_{est\_b} = 0.9633,\ \sigma_{est\_b} = 0.9297$$

These values then produced the estimations in figures 1 and 2. These have also been overlayed with the actual distributions as provided within the problem statement. As expected the Gaussian parametric estimation approximated the Gaussian data very well while not approximating the exponential distribution particularly well.

Figure 1: Parametric estimation of a with a Gaussian form assumed (true distribution in red, estimated distribution in orange).

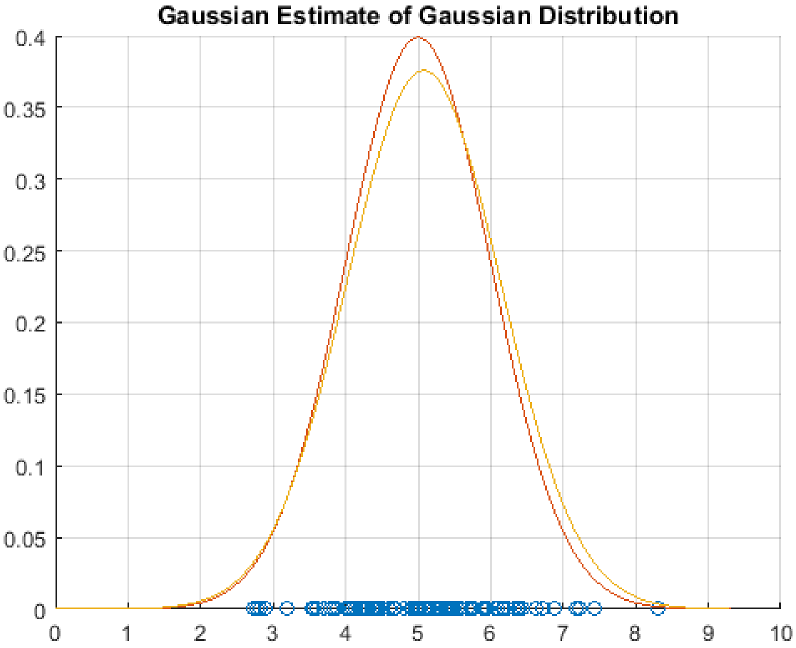
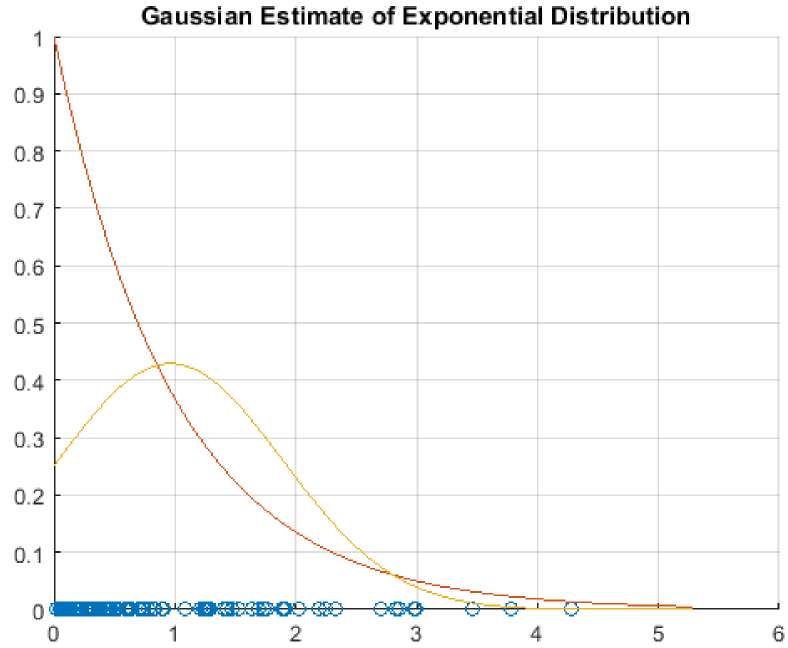


Figure 2: Parametric estimation of b with a Gaussian form assumed (true distribution in red, estimated distribution in orange).



2.2 Parametric Estimation - Exponential

The exponential parametric estimator can be derived using the exponential probabilistic distribution:

$$p(x) = \begin{cases} 0 & x < 0 \\ \lambda e^{-\lambda x} & x \geq 0 \end{cases} \tag{4}$$

The parametric estimator is determined by maximizing the sample set probability, which is the product of the individual sample probabilities, as they are considered independent of one another:

$$p(\{x_i\}|\theta) = \prod_{i=1}^{N} p(\underline{x}_i|\theta) \tag{5}$$

Where θ is the estimated parameter applied to the sample set probability:

$$p(x_i|\lambda_{est}) = \prod_{i=1}^{N} \lambda_{est} e^{-\lambda_{est} x} \tag{6}$$

Let $I(\lambda_{est})$ be the the natural log of the equation above:

$$\begin{aligned} I(\lambda_{est}) &= ln(\lambda_{est}^{N} \prod_{i=1}^{N} e^{-\lambda_{est} x}) \\ &= N ln(\lambda_{est}) - \sum_{i=1}^{N} (\lambda_{est} x_i) \end{aligned} \tag{7}$$

The value for λ_{est} that maximizes the sample set probability can be determined by taking the partial derivative with respect to λ_{est}:

$$\frac{\partial I}{\partial \lambda_{est}} = \frac{N}{\lambda_{est}} - \sum_{i=1}^{N}(x_i) = 0$$
$$\lambda_{est} = \frac{N}{\sum_{i=1}^{N}(x_i)} \tag{8}$$

The equation for the estimated λ value can be applied to the datasets provided in the *lab2_1.mat* file to derive the estimated λ values for both the Gaussian and exponential datasets:

$$\lambda_{est\_a} = 0.1970$$
$$\lambda_{est\_b} = 1.0381$$

The value for λ_{est_b} is extremely close to the actual λ value of 1 for dataset b, indicating that the exponential parametric estimator should achieve a curve relatively equivalent to the actual dataset. Both estimated λ values were used to produce the estimations in figures 3 and 4, along with the overlaid actual distributions in orange.

Figure 3: Parametric estimation of a with an Exponential form assumed (true distribution in red, estimated distribution in orange).

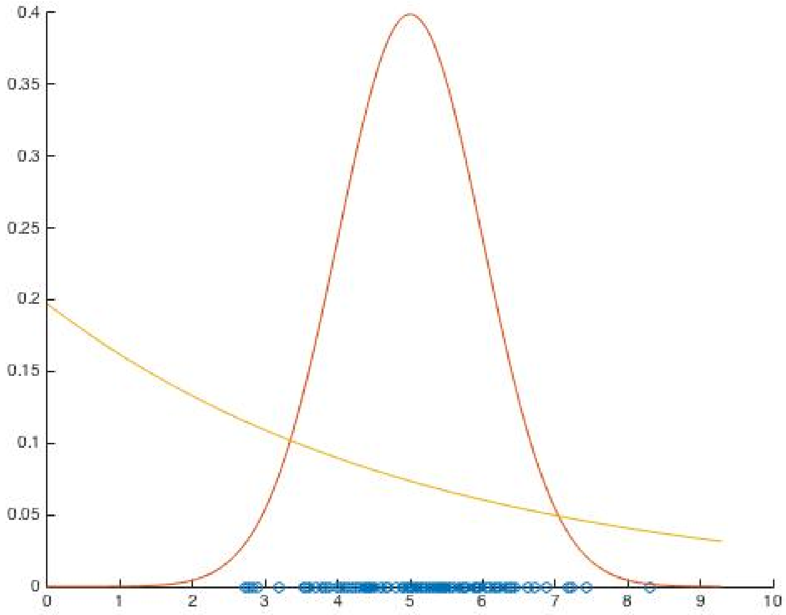
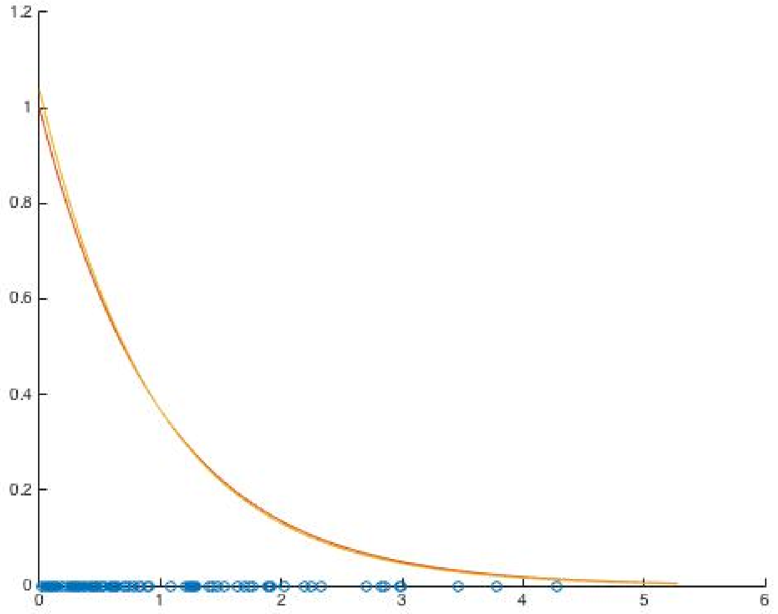


Figure 4: Parametric estimation of b with an Exponential form assumed (true distribution in red, estimated distribution in orange).



As expected, the exponential parametric estimator performed extremely poorly for the Gaussian distributed dataset. However, as predicted using the estimated λ value, the the exponential parametric estimator performed almost perfectly with the exponential dataset.

2.3 Parametric Estimation - Uniform

The uniform distribution is an extremely simplistic probability distribution where all samples within the range of a dataset are given equal probabilities:

$$p(x) = \begin{cases} 0 & x < a \\ \frac{1}{b-a} & a \leq x \leq b \\ 0 & b < x \end{cases} \tag{9}$$

The Maximum Likelihood Estimate (MLE) can be derived using a similar method to the exponential case:

$$\begin{aligned} p(x_i|[a_{est}, b_{est}]) &= \prod_{i=1}^{N} \frac{1}{b_{est} - a_{est}} \\ &= \frac{1}{(b_{est} - a_{est})^N} \end{aligned} \tag{10}$$

Since this is a uniform distribution, all samples must be included within the parametric estimation, i.e. the distribution must be at least the size of the range of the dataset. This imposes two additional constraints on the MLE:

$$\begin{aligned} a_{est} &\leq min(x_i) \\ b_{est} &\geq max(x_i) \end{aligned} \tag{11}$$

The log-likelihood of the MLE can be determined as follows:

$$\begin{aligned} I([a_{est}, b_{est}) &= ln(\frac{1}{(b_{est} - a_{est})^N}) \\ &= Nln(\frac{1}{b_{est} - a_{est}}) \end{aligned} \tag{12}$$

Similarly, the partial derivative with respect to both estimated values of the log-likelihood equation can be taken and set to 0, in order to determine the maximal value of the MLE:

$$\frac{\partial I}{\partial a_{est}} = \frac{N}{b_{est} - a_{est}} \ , \ \frac{\partial I}{\partial b_{est}} = \frac{-N}{b_{est} - a_{est}} \tag{13}$$

Since the partial with respect to a_{est} is always increasing (as $b_{est} > a_{est}$), it can be minimized by choosing the largest value possible for a_{est}, which is $min(x_i)$ from equation 11. Similarly, the partial with respect to b_{est} is always decreasing, so it can be maximized by choosing the smallest value for b_{est}, which is $max(x_i)$. This can be summarized as:

$$a_{est} = min(x_i) \ , \ b_{est} = max(x_i) \tag{14}$$

These values for the estimates of a and b provide the largest MLE possible as it provides an interval that's equivalent to the range of the dataset. Equation 12 can be applied to the datasets to derive the estimated values of a and b:

$$a_{est\_a} = 2.7406 \ , \ b_{est\_a} = 2.7406 \ a_{est\_b} = 0.0143 \ , \ b_{est\_b} = 4.2802$$

As expected, the uniform distribution estimated parameters perform poorly in attempting to recreate the Gaussian and exponential distributed datasets. However, the estimated distributions do provide a nice visualization of the spread or range of the datasets, as shown in the figures below:

Figure 5: Parametric estimation of a with a Uniform form assumed (true distribution in red, estimated distribution in orange).

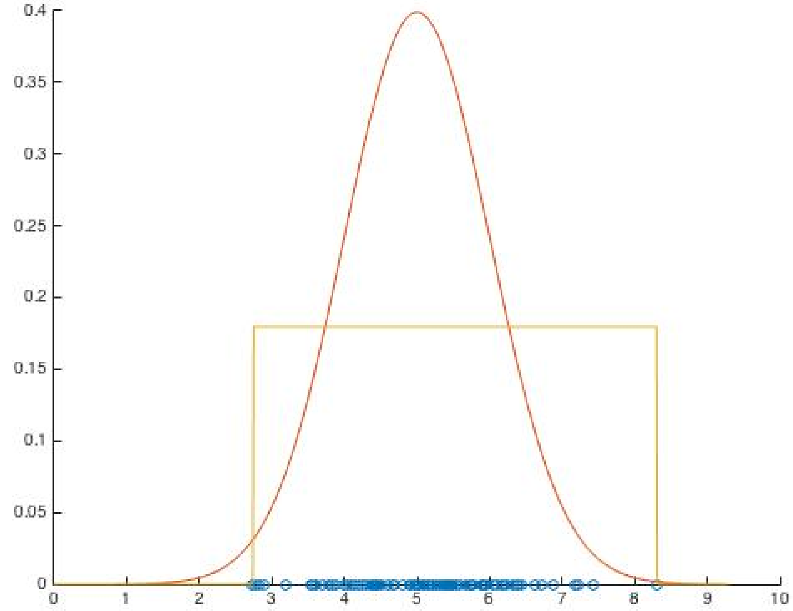
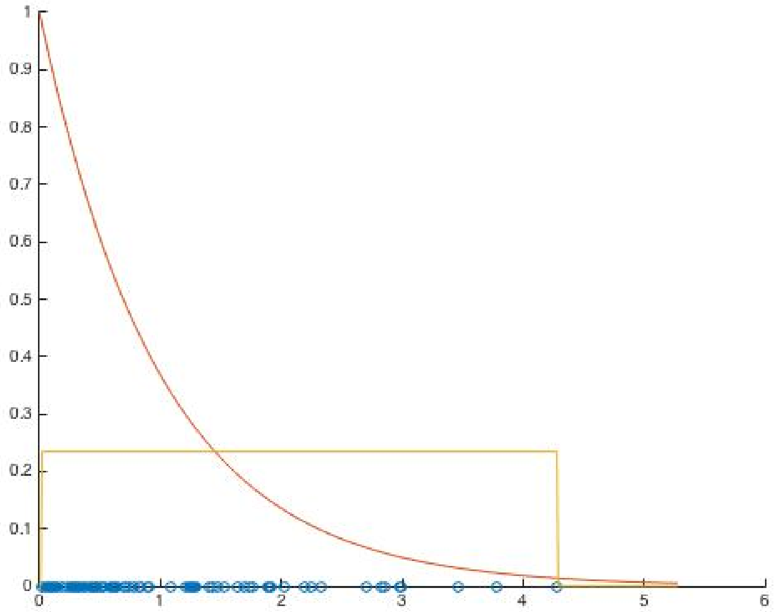


Figure 6: Parametric estimation of b with a Uniform form assumed (true distribution in red, estimated distribution in orange).



2.4 Non-Parametric Estimation

The fundamental idea behind non-parametric estimation is to come up with an estimated PDF for a cluster of points based on their density. The Parzen method is based on the premise that every point x_i influences the value of the PDF in the vicinity of x_i. That is to say that the more points there are in a given region, the higher the PDF value in that region i.e. higher density ⇒ higher probability. The estimated PDF can thus be written as:

$$\hat{p} \propto \phi(x - x_i)$$

where ϕ is a normalized window function (such as Gaussian, exponential, rectangular or triangular). To change the locality of influence of a sample, the window function can be stretched by a factor of h (i.e. the standard deviation):

$$\hat{p} = \frac{1}{N}\sum_i \frac{1}{h}\phi(\frac{x - x_i}{h})$$

The class densities were estimated using the Parzen method with a Gaussian window having standard deviations of 0.1 and 0.4. The results can be seen in the figures below.

Figure 7: Non-Parametric estimation of a with a Gaussian window with std dev = 0.1 (true distribution in red, estimated distribution in orange).

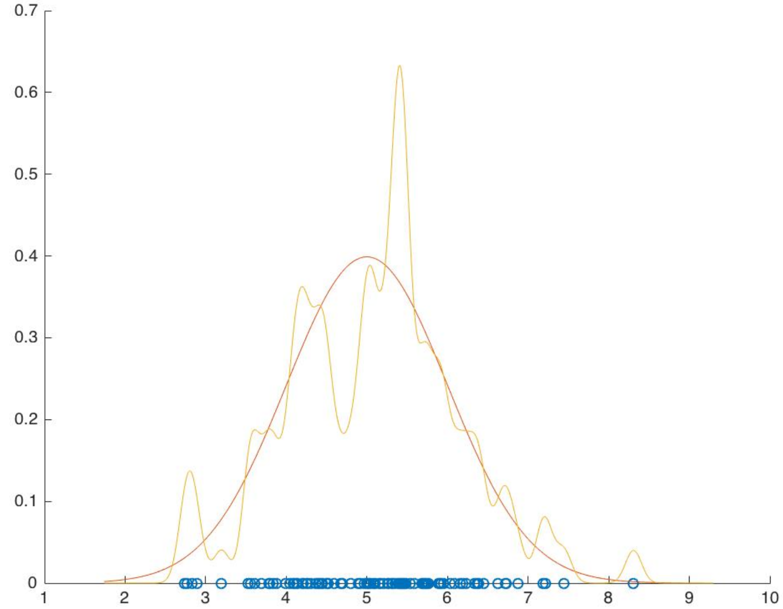
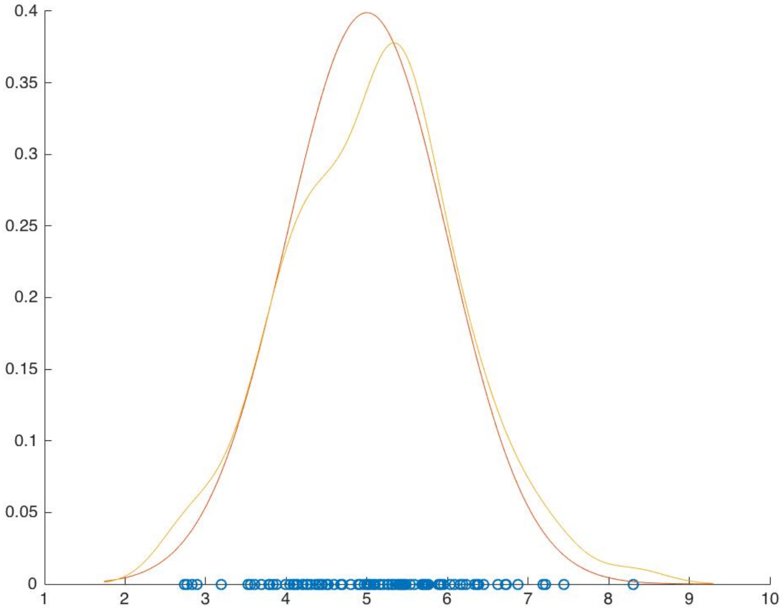


Figure 8: Non-Parametric estimation of a with a Gaussian window with std dev = 0.4 (true distribution in red, estimated distribution in orange).



The lower standard deviation of 0.1 results in a noisier estimated distribution with sharp peaks at higher densities of sample points. The estimator with the higher standard deviation of 0.4 provides better resolution and a fairly decent estimate of the true distribution given the limited number of samples.

Figure 9: Non-Parametric estimation of b with a Gaussian window with std dev = 0.1 (true distribution in red, estimated distribution in orange).

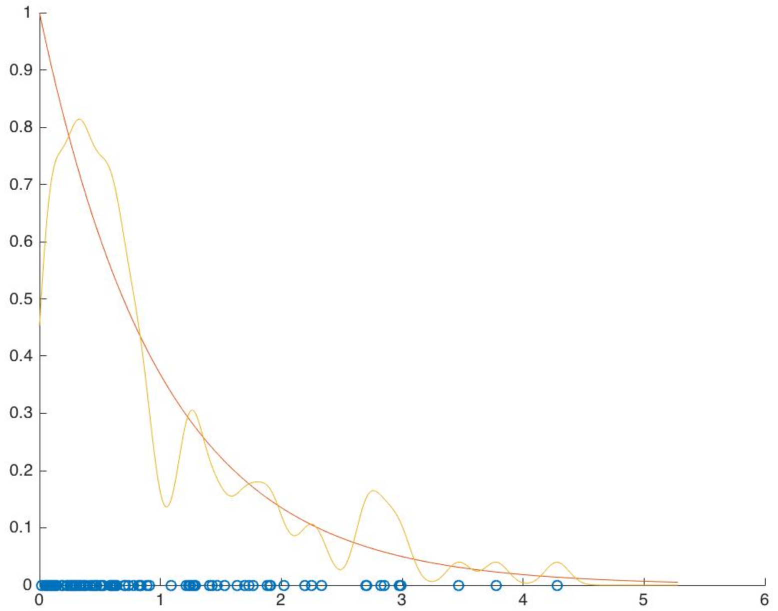
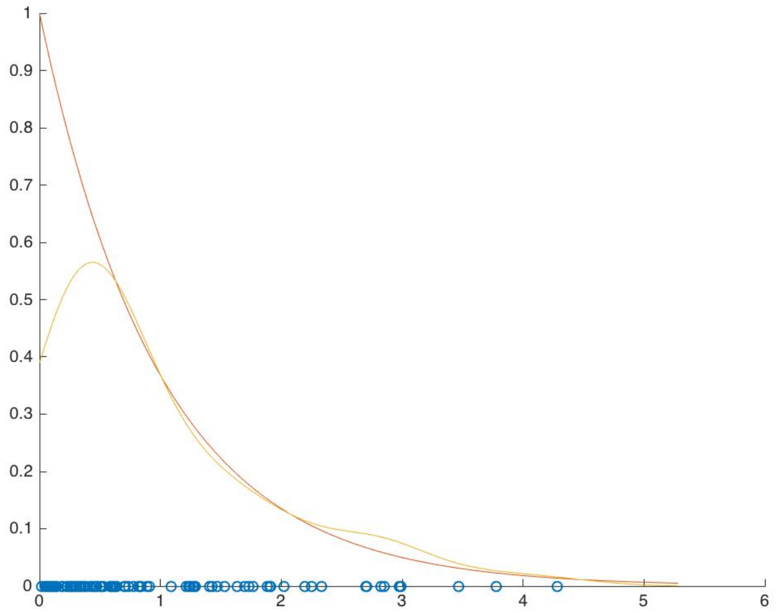


Figure 10: Non-Parametric estimation of b with a Gaussian window with std dev = 0.4 (true distribution in red, estimated distribution in orange).



Even given an exponential distribution of samples, the Parzen window method provides decent results in estimating the true distribution. The estimator with the 0.1 standard deviation was once again noisy with sharp peaks but still follows the trend of the true distribution. The estimator with the 0.4 standard deviation is once again the better of the two, however it is significantly lower than the true distribution for values less than 1. This is due to the fact that a Gaussian window is being used to estimate an exponential distribution.

2.5 Comparison

For the Gaussian dataset, the best performing estimated density was the Gaussian estimator in Figure 1, as the curve fit the closest to the dataset Gaussian curve. Both the uniform and exponential estimated densities were wildly, off, as they're inherently restricted by the nature of their functions: since the exponential density function is always decreasing, it is impossible for it to fit the bell-shaped curve of a Gaussian distribution. Similarly, a uniform distribution will always have a square curve, as it must maintain equivalent probabilities for all samples within a dataset, thus it also cannot emulate a bell-shaped curve. It is interesting to note that the non-parametric Parzen method produced estimated densities that fit the general shape of the Gaussian distribution with a degree of noise. As the standard deviation of the Gaussian window increased, so did the noise in the Parzen-estimated density.

Similarly, the exponential estimator performed the best against the exponential dataset, as it was able to emulate the distribution almost exactly, unlike the Gaussian and uniform distributions, which were again limited by the shape of their parametric functions. The Parzen method again produced densities that fit the general trend of the exponential function but with added noise that decreased as the standard deviation of the Gaussian window increased.

In general, the parametric approach should be ideally used if the distribution of the dataset is already known. For example, in a supervised learning model where the dataset is known to fit a Poisson distribution, it is recommended to use the parametric approach to derive the estimated density. However, in the case where the dataset's distribution is not known, i.e. in an unsupervised model, nonparametric estimator, such as the Parzen method, work much better at approximating the density function of the distribution.

3 Model Estimation: 2-D Case

Extending estimation to 2 dimensions, datasets for 3 classes were provided and divided into training and test sets.

3.1 Parametric Estimation

The first estimation performed assumed each cluster to be normally distributed. From this a sample covariance and mean were calculated. The values for each cluster were:

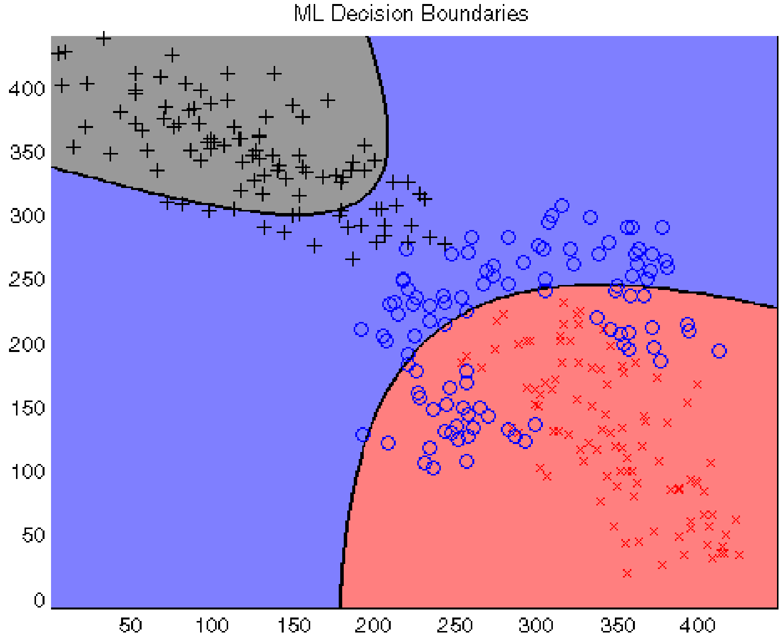
$$\mu_{est\_al} = \begin{bmatrix} 347.1600 & 131.2000 \end{bmatrix}^T \Sigma_{est\_al} = 10^3 \cdot \begin{bmatrix} 1.7666 & -1.6106 \\ -1.6106 & 3.3435 \end{bmatrix}$$
$$\mu_{est\_bl} = \begin{bmatrix} 291.8400 & 224.0200 \end{bmatrix}^T \Sigma_{est\_bl} = 10^3 \cdot \begin{bmatrix} 3.3157 & 1.1760 \\ 1.1760 & 3.4140 \end{bmatrix}$$
$$\mu_{est\_cl} = \begin{bmatrix} 119.5500 & 346.6700 \end{bmatrix}^T \Sigma_{est\_cl} = 10^3 \cdot \begin{bmatrix} 2.7385 & -1.3272 \\ -1.3272 & 1.6993 \end{bmatrix}$$

These values were then used to build a Maximum Likelihood decision boundary with the following discriminant function:

$$(\vec{x} - \vec{\mu_b})^T \cdot \Sigma_b^{-1} \cdot (\vec{x} - \vec{\mu_b}) - (\vec{x} - \vec{\mu_a})^T \cdot \Sigma_a^{-1} \cdot (\vec{x} - \vec{\mu_a}) = 0$$

By plugging in the calculated values the decision boundaries within figure 11 were obtained. These follow an expected pattern and it should be noted that the middle cluster was allocated a large decision boundary while it was composed of relatively few points.

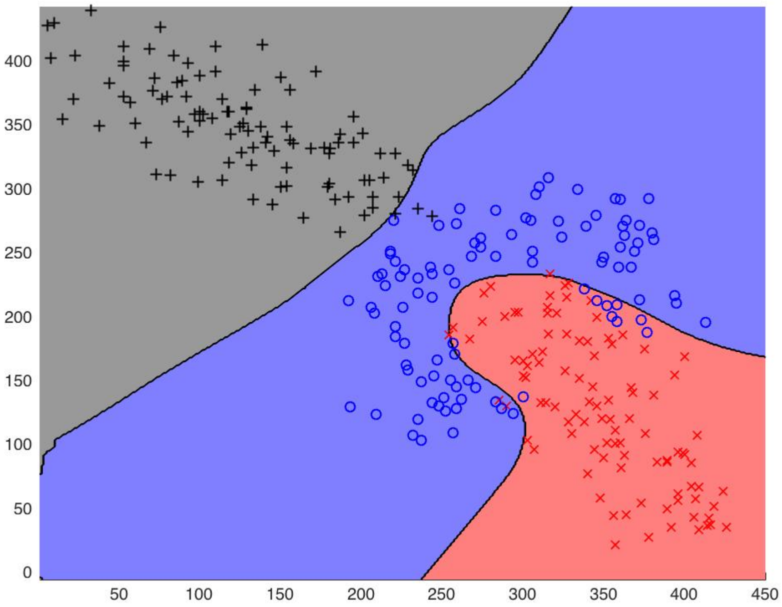
Figure 11: ML decision boundary of *at (red), bt (blue), ct (black)* (test sets) where each is assumed to be normally distributed.



3.2 Non-Parametric Estimation

In the 2D case, the Gaussian window is constructed as a matrix with a variance of 400. This window is passed into the *parzen2* function along with the training sets for each class to construct estimated 2D distributions for each class. A Maximum Liklihood classifier was then constructed by picking the class with the maximum estimated probability for each point in the 2D meshgrid. The ML decision boundary is plotted below along with the sample points from the test sets.

Figure 12: ML decision boundary of *at (red), bt (blue), ct (black)* (test sets) where each class distribution is estimated using a Gaussian window with variance of 400.



3.3 Comparison

The 2D case is where the non-parametric approach really shines. It seems like class A and C are normally distributed so both the parametric and the non-parametric estimations can classify them well. However, class B is definitely not normally distributed and has more of a half-crescent shape. This is where the parametric estimation approach fails and has a high rate of misclassification for class B. A lot of class B test points are parametrically classified as class A and class C points are classified as class B.

The non-parametric approach handles class B really well and produces a nicely curved ML decision boundary to separate class B from class A. This results in fairly low levels of misclassified points and a high level of accuracy.

In general, it is not always possible to use a parametric approach because bad assumptions might be made about the true underlying distributions of the classes. If they do not happen to be Gaussian or Exponential or whatever parametrization is chosen, a classifier built on this would result in high levels of misclassification and error. That being said, if the distributions of the classes can be reasonably approximated to be close to a known distribution such as Gaussian or Exponential through prior knowledge, then a parametric approach would work well.

If no assumptions or prior knowledge can help understand the distribution of the sample data, the non-parametric approach is the more powerful one as it is agnostic or the true distribution and produces good estimates. Playing around with the Parzen window can estimate any data sample reasonably well and produce a classifier with relatively low rates of error and misclassification.

4 Sequential Discriminants

This section of the lab deals with developing a sequence of classifiers such that the aggregation of the individual classifiers becomes a more powerful overall classifier. Two classes *a* and *b* are provided with sample points stored as (x,y).

First by randomly selecting discriminants until all training samples were classified the results in figure 13, 14 and 15 were obtained. If one were to then test the training error using these discriminants the probability of error would be zero for one of the classes. This is due to the fact that during training discriminants are selected until each point within one class is correctly classified regardless of the error within another class (it could select an arbitrarily bad discriminant!).

Figure 13: Sequential classifier separating one class perfectly.

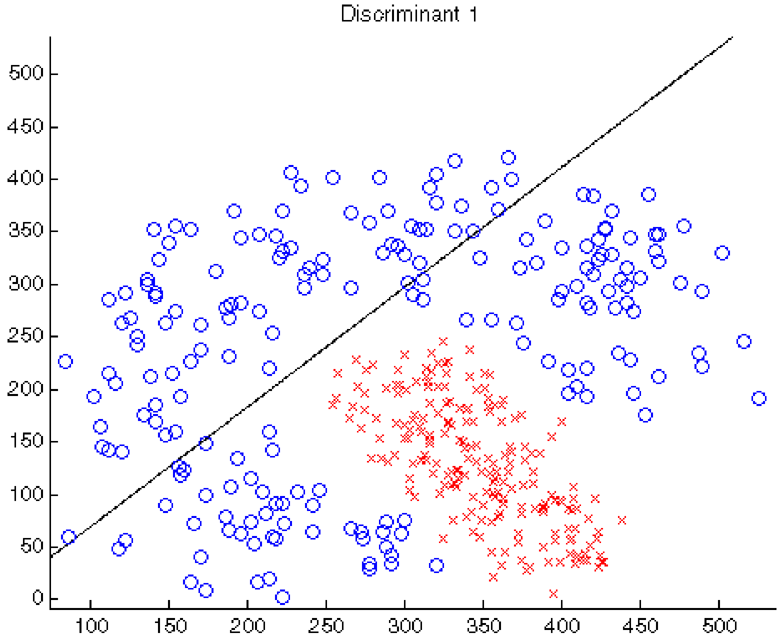


Figure 14: Sequential classifier separating one class perfectly.

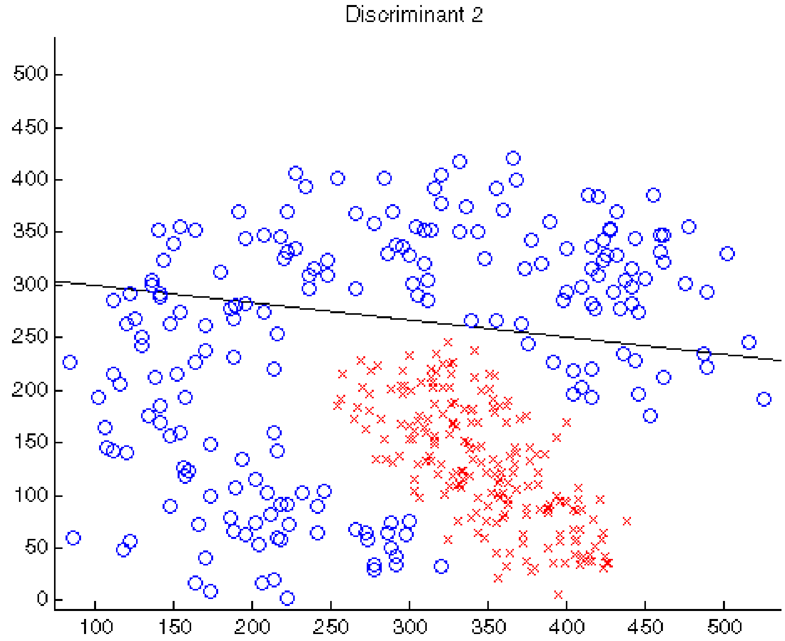
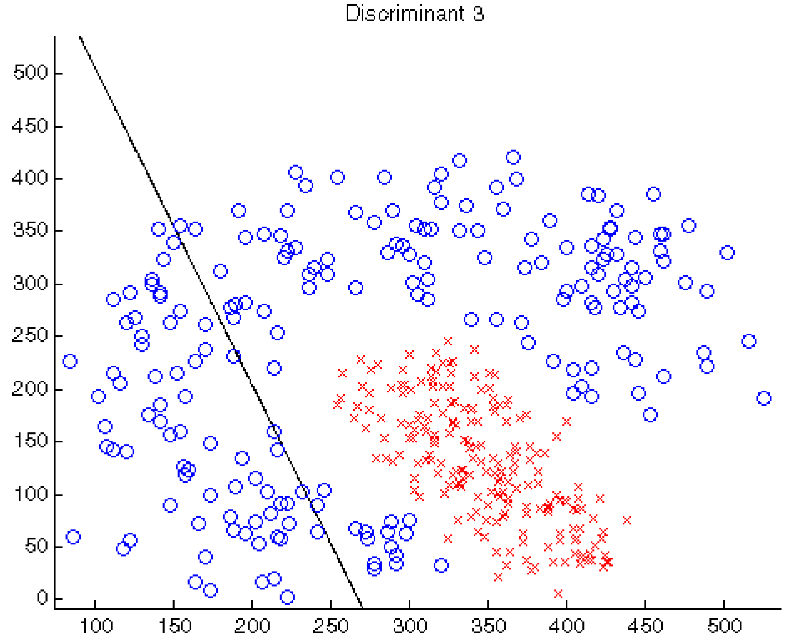
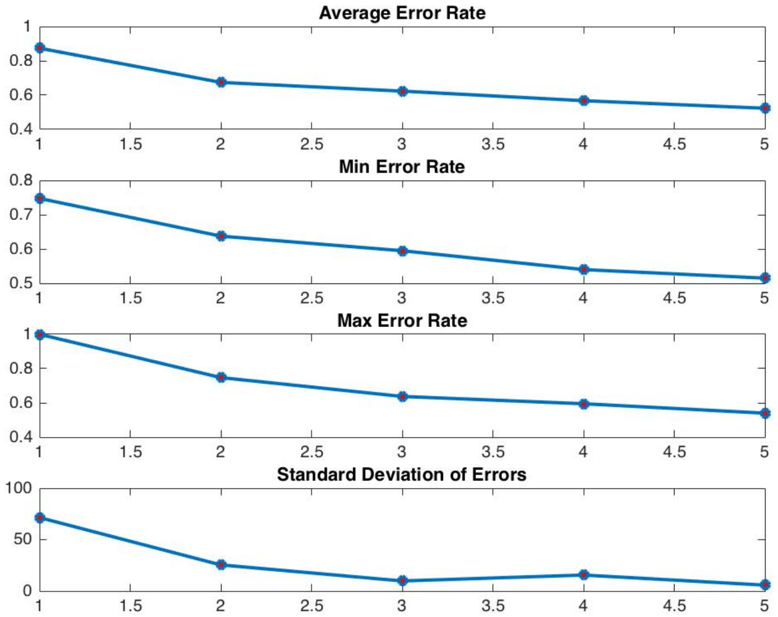


Figure 15: Sequential classifier separating one class perfectly.



By limiting the sequential classifier to only twenty tries and learning J=5 discriminants, the results within Figure 16 were obtained. As expected, the average error rate decreased as well as the minimum and maximum error rates and standard deviation. If one were to limit the number of training point pairs, the error rate during training would increase. However, the discriminants would not over-fit the training data as drastically and thus the performance of the classifier may generally improve.

Figure 16: Error Analysis for J=5 classifiers with a limit of 20 tries



5 Conclusion

Parametric estimation works well if one assumes the correct distribution type for the data provided. In general, if the distribution type is not known, it is better to use a non-parametric approach and tweak the window size to obtain a decent fit.

When fitting sequential classifiers, if one does not limit either the number of discriminants or the number of training points then they will always obtain a perfect fit to the training data. This is an undesirable effect due to the high potential to over-fit to the training data.