SYDE-572: Introduction to Pattern Recognition

Lab 2: Model Estimation and Discriminant Functions - **Report**

Group 17: Marta Zheplinska - 21050969
mzheplin@uwaterloo.ca
Professor - Alexander Wong

Introduction

In Lab 2, methods of estimation of distribution parameters are considered for the purpose of approximate reproduction of the probability distribution of data. In the first part of the Lab 2, parametric and non-parametric estimation of data in 1-D space was considered. The assessment was carried out on the data of classes a and b, where the real distribution of elements is known in advance. In the second part, parametric and non-parametric estimations of data distribution of three given classes in 2-D space were carried out. In the third part, all the beauty of the aggregation of discriminant functions for the construction of decision boundaries for the classification of clusters was considered.

Model Estimation 1-D case

In the first part, two sets of data (class a and class b) where each element is represented as a number on a 1-D plane were considered and investigated.

Class *a*

It is known that the elements of class a are normally distributed

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_a^2}} \cdot e^{-\frac{(x-\mu_a)^2}{\sigma_a^2}}$$

with parameters:

$$\mu_a = 5 \ , \ \sigma_a = 1$$

Below, the construction of all plots related to data set a is implemented in the file **class_a_estimations.m** in the directory **/scripts for plots**.

Parametric Estimation – Gaussian

We assume that a given data is distributed normally and our goal is to reproduce the parameters of the distribution based on the data set. In the case of the Gaussian distribution, we have the following estimates:

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

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

The code for the function that calculates the values of the parameters of the normal distribution can be found in file **Gaussian_estimation_1D.m** of directory **/1D_estimators**. As a result of the program execution, the following estimates of the parameters of the Gaussian distribution were obtained for the data set of class a:

$$\mu_{est} = 5.0763 \quad \sigma_{est} = 1.0618$$

Figure 1 shows plots of the data distribution with true parameters (green) and with estimated ones (blue):

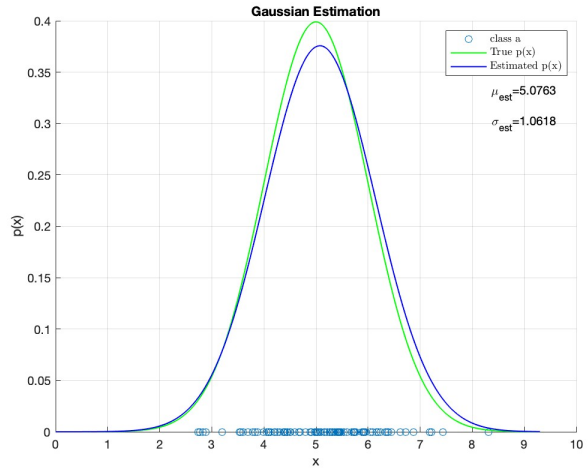

Figure 1: Gaussian estimation of Gaussian distribution

Parametric Estimation – Exponential

Assume that the data is exponentially distributed.

$$p_{est}(x) = \begin{cases} \lambda e^{-\lambda x} \ , \ x \geq 0 \\ 0 \ , \ x < 0 \end{cases}$$

There is an estimate of the parameter λ :

$$\lambda_{est} = \frac{N}{\sum_{i=1}^{N} x_i}$$

The code for the function that calculates the values of the parameters of the exponential distribution can be found in file **Exponential_estimation_1D.m** of directory **/1D_estimators**. As a result of the execution of the program, the estimated parameter of the exponential distribution for class a was calculated

$$\lambda_{est} = 0.19699$$

and a chart of estimated distribution (blue) was plotted on top of the real normal distribution (green) (Figure 2).

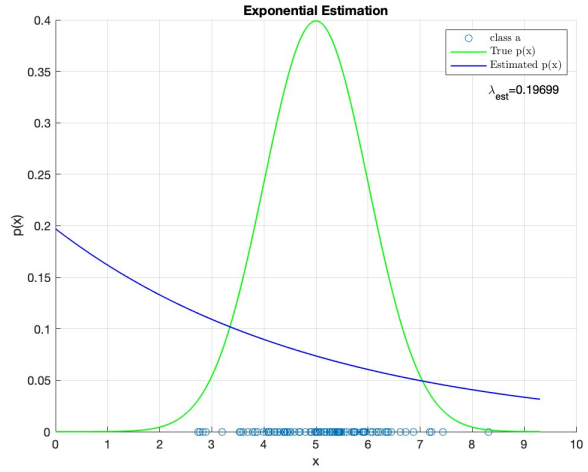

Figure 2: Exponential estimation of Gaussian distribution

Parametric Estimation – Uniform

We assume that the data are uniformly distributed on the interval [a, b]

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

and our task is to find the ends of the interval. The parameters that will maximize the reliability of the distribution will be as follows:

$$a = min(x_1, ..., x_N)$$

$$b = max(x_1, ..., x_N)$$

The code for the function that calculates the values of the parameters of the uniform distribution can be found in file **Uniform_estimation_1D.m** of directory **/1D_estimators**. As a result of the execution of the program, the estimated parameters of the uniform distribution for class a were obtained

$$a_{est} = 2.7406$$

$$b_{est} = 8.3079$$

Figure 3 below shows plots of the true distribution (green) and of estimated uniform distribution (blue):

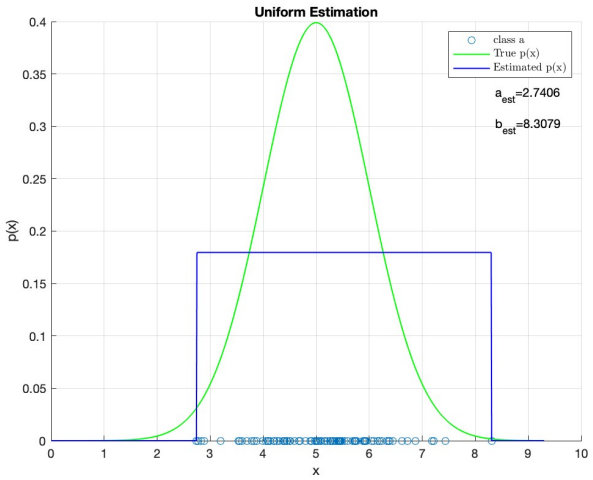

Figure 3: Uniform estimation of Gaussian distribution

Non-parametric estimation

More than real are situations when it is difficult or impossible to guess the probability distribution of data set, and then non-parametric methods of estimating the data distribution come to the rescue. In the Lab 2 we considered the Parzen Window estimation method. The non-parametric method is based on the density of data set clusters in certain intervals. In the method of Parzen Windows, the value of the received PDF will be larger on those intervals, where, accordingly, the actual density of data is greater. In Lab 2, the Gaussian Parzen windows method was used with h = 0.1 and h=0.4 as standard deviations. The distribution construction function has the form:

$$p_{est}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi h^2}} \cdot exp \frac{(x - x_i)^2}{2h^2}$$

The code for the function that generates the PDF with Gaussian Parzen Windows can be found in file **Parzen_estimation_1D.m** of directory **/1D_estimators**. Below are graphs of the estimated density (blue) of the distribution over the true one (green) for the parameter h =0.1 (Figure 4) and h = 0.4 (Figure 5):

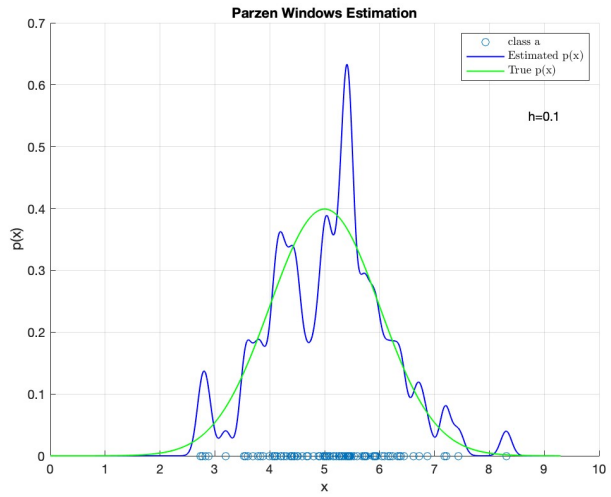

Figure 4: Gaussian Parzen windows of Gaussian distribution with h=0.1

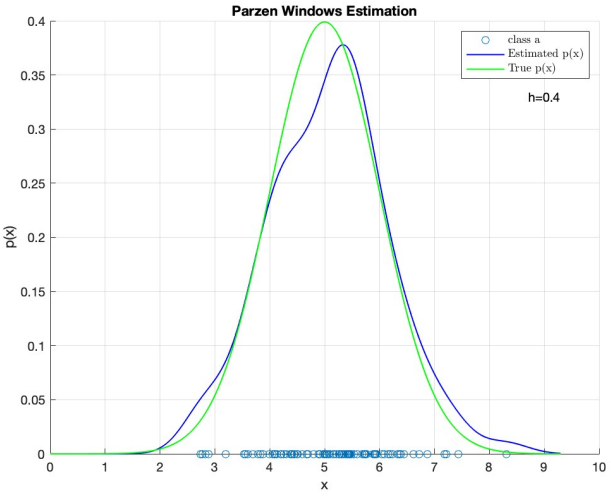

Figure 5: Gaussian Parzen windows of Gaussian distribution with h=0.4

Conclusions for class *a* estimations

Question

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

Answer for data set *a*

Obviously, **knowing the distribution of the data in advance**, the best solution was to reproduce the parameters of this particular distribution and the parametric approach is preferred. So, the Gaussian distribution was the most accurate for data set a. The parameters of the distribution turned out to be very close to the true parameters. The assumption that the data are distributed exponentially and uniformly did not give high accuracy, and the use of these discriminants would give many errors as a result. The non-parametric method turned out to be quite successful and accurate, and the obtained discriminant function makes sense in future uses when **we do not reliably know the density of the data distribution**. As a result, the Gaussian Parzen Windows method turned out to be more accurate for h = 0.4 than for h = 0.1.

Class b

Another data set b that was estimated in Lab 2 was exponentially distributed with parameter $\lambda_b = 1$:

$$p(x) = \begin{cases} \lambda_b \cdot e^{-\lambda_b x} \ , \ x \geq 0 \\ 0 \ , \ x < 0 \end{cases}$$

The logic of parametric and non-parametric estimations of the model for data set b does not differ from the logic for class a. Therefore, repetitive details and parameter estimation formulas are omitted to avoid repetition.
The code for plotting the charts of the considered estimations of data set **b** is written in the file **class_b_estimations.m** in the directory **/scripts for plots**.

Parametric Estimation – Gaussian

Under the assumption that the data are normally distributed, the following estimated parameters were obtained:

$$\mu_{est} = 0.9633$$

$$\sigma_{est} = 0.92967$$

A chart of estimated distribution (blue) was plotted on top of the real exponential distribution (green) (Figure 6).

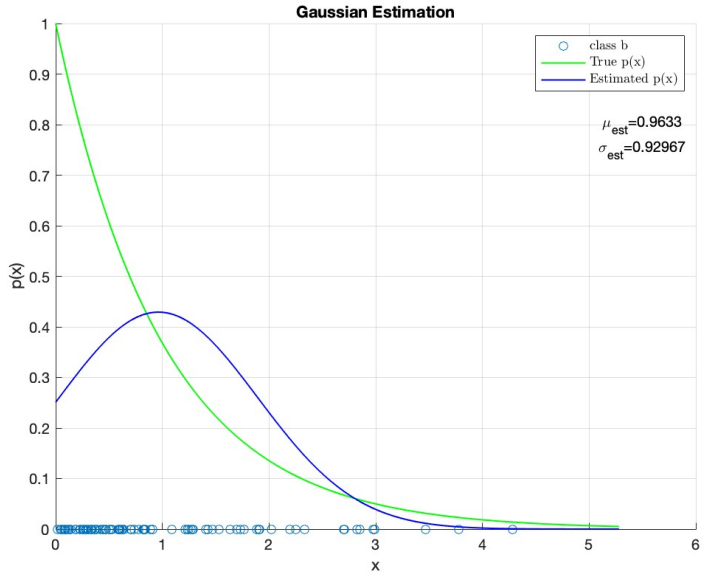

Figure 6: Gaussian estimation of Exponential distribution

Parametric Estimation – Exponential

Given that we know that the true PDF of the data is exponential, we expect high accuracy from the estimated PDF, which is also exponential. The estimated parameter was calculated and obtained:

$$\lambda = 1.0381$$

Figure 7 shows plots of the data distribution with true parameter (green) and with estimated one (blue):

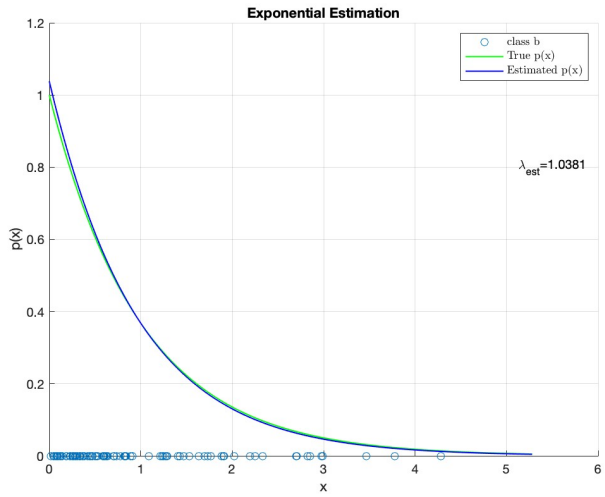

Figure 7: Exponential estimation of Exponential distribution

Parametric Estimation – Uniform

We assume that the data are uniformly distributed on the interval [a, b] and our goal is to find the ends of the interval. Calculations of the parameters of the uniform distribution for class b were obtained:

$$a_{est} = 0.014322$$

$$b_{est} = 4.2802$$

Figure 8 below shows plots of the true exponential distribution (green) and of estimated uniform distribution (blue):

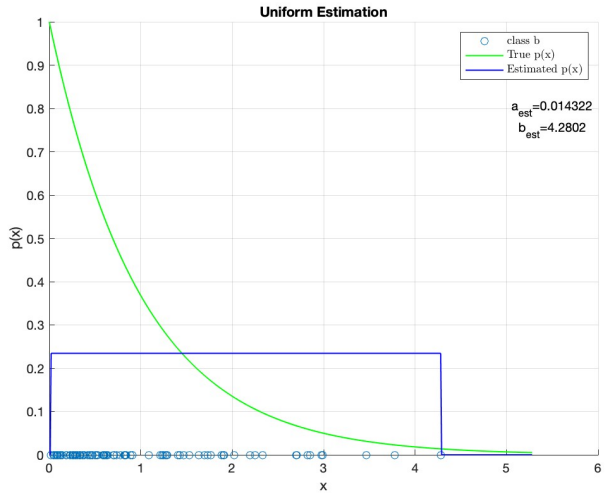

Figure 8: Uniform estimation of Exponential distribution

Non-parametric Estimation

As for class a, non-parametric estimations were also performed for class b with the help of Gaussian Parzen windows with the parameters h = 0.1 and h=0.4 as standard deviations. Below are plots of the estimated density (blue) of the distribution for the parameter h =0.1 (Figure 9) and h = 0.4 (Figure 10) over the true one (green) :

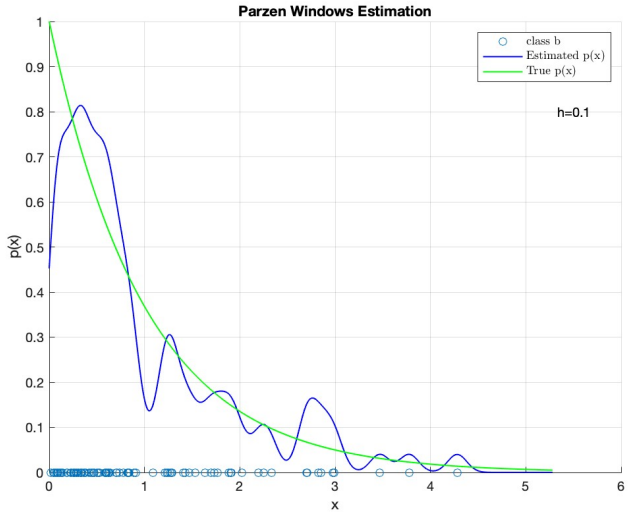

Figure 9: Gaussian Parzen window of Exponential distribution with h=0.1

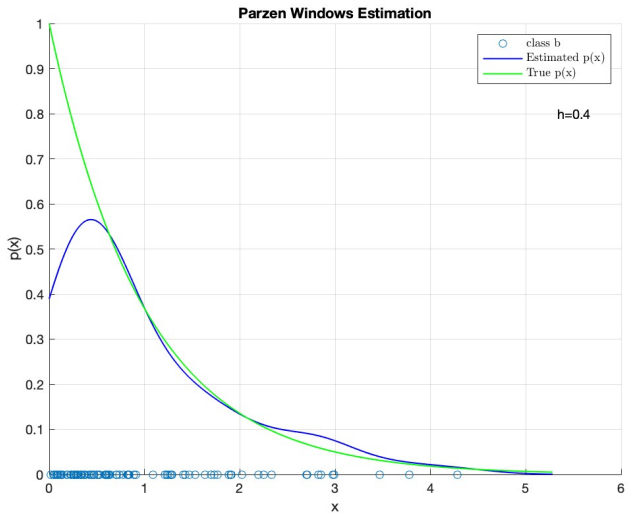

Figure 10: Gaussian Parzen window of Exponential distribution with h=0.4

Conclusions for class *b* estimations

Question

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

Answer for data set *b*

Undoubtedly, the estimated discriminant function of the exponential distribution most accurately reproduced the distribution of the data, since they themselves are exponentially distributed. Therefore, the two PDFs (figure 7) practically overlapped each other. It is obvious that using the estimate discriminant function of a known distribution is the best solution in such a case. Gaussian estimation and uniform estimation did not cope with their task, and their further use will lead to large errors and incorrect classification of clusters. Non-parametric estimation of the model using the method of Parzen windows showed relative accuracy and can be used when the data distribution is unknown. With parameter h = 0.4, the resulting curve is closer to the true exponential PDF that with h =0.1.

Model Estimation 2-D case

In this part of the laboratory work, parametric and non-parametric evaluations of three classes on the 2D space were performed. Here, the clusters of each class are represented as the points on a plane. The code for plotting the charts of the considered estimated decision boundaries on 2D space of classes *al, bl, cl* is written in the file **estimations_2d.m** in the directory **/scripts for plots**.

Parametric Estimation

Suppose that the data of three classes are distributed according to the normal law. Then the estimation of the parameters of the normal distribution of each class has the form:

$$\bar{\mu_{est}} = \frac{1}{N}\sum_{i=1}^{N}\bar{x_i} \quad S_{est} = \frac{1}{N}\sum_{i=1}^{N}(\bar{x_i} - \bar{\mu_{est}})(\bar{x_i} - \bar{\mu_{est}})^T$$

The file **Gaussian_estimation_2D.m** in the directory **/2D_estimators** implements the calculation of the estimated mean and the covariance matrix. To perform the task, the parameters of three classes *al, bl, cl* were estimated:

$$\bar{\mu_{est,al}} = [347.16\ 131.2] \quad S_{est,al} = 10^3 \cdot \begin{bmatrix} 1.7666 & -1.6106 \\ -1.6106 & 3.3435 \end{bmatrix}$$

$$\bar{\mu_{est,bl}} = [291.84\ 224.02] \quad S_{est,bl} = 10^3 \cdot \begin{bmatrix} 3.3157 & 1.176 \\ 1.176 & 3.414 \end{bmatrix}$$

$$\bar{\mu_{est,al}} = [119.55\ 346.67] \quad S_{est} = 10^3 \cdot \begin{bmatrix} 2.7385 & -1.3272 \\ -1.3272 & 1.6993 \end{bmatrix}$$

The next step was the construction of ML decision boundaries. A point on the plane belongs to the class with the highest probability of belonging. ML classification is a subversion of MAP classification, when it is assumed that the probability of each class is equal:

$$P(A) = P(B)$$

The formula for ML classification in the case of normal distributions of classes has the form:
If

$$(\bar{x} - \bar{\mu_B})^T\Sigma_B^{-1}(\bar{x} - \bar{\mu_B}) - (\bar{x} - \bar{\mu_A})^T\Sigma_A^{-1}(\bar{x} - \bar{\mu_A}) > \ln\frac{|S_A|}{|S_B|}$$

then $\bar{x}$ belongs to class A. The implementation of this classification method is saved in the files **ML_calc.m** and **ML_classifier.m** in the directory **/Classifiers**. Figure 11 shows the decision boundaries formed as a result of the execution of the program for 3 classes on the plane:

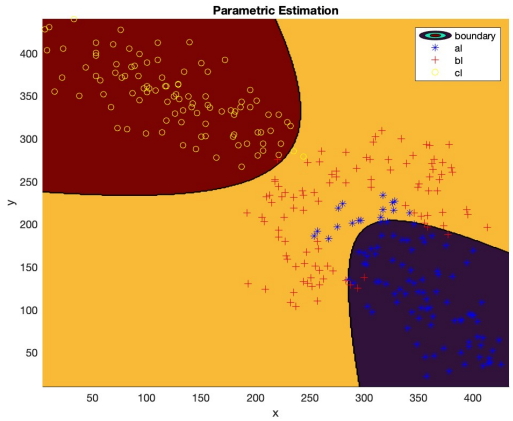

Figure 11: 2D parametric Gaussian estimation

Non-parametric Estimation

A ready-made function in the file **parzen2.m** was used for non-parametric estimation of the distribution into three classes in 2D space. For this, the global minimum and maximum values of the first and second coordinates were determined among the three classes and a **resolution** of 0.5 was set. A *win* function defining the normal distribution was also specified:

$$\frac{1}{\sqrt{2\pi h^2}} \cdot exp\frac{-x^2}{2h^2}$$

As a result of the execution of the program, decision boundaries for classes *al, bl, cl* were built using the method of Parzen windows (Figure 12):

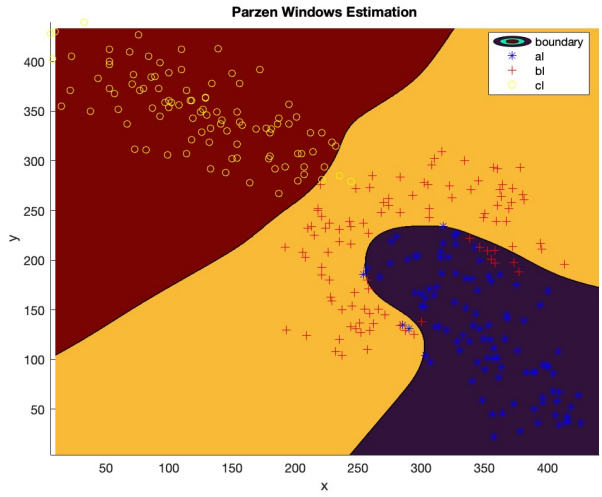

Figure 12: 2D non-parametric Parzen Windows estimation

Comparison

Questions
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?

Answer
The parametric estimation in 2D space showed good results, but many clusters of class *al* and class *bl* were incorrectly classified. Therefore, it can be assumed that the data of both or one of the two classes are not distributed according to a normal distribution. It can be seen visually that the data set of class *bl* with the lowest probability is distributed according to the Gaussian law.
Non-parametric estimation and construction as a result of the decision boundary showed a clear and more accurate separation of clusters by classes. And relatively few clusters were misclassified. Using the example of these three classes, the full power of non-parametric estimation was shown.
In general, preference should be given to parametric estimation of distributions only when the real distribution of the data is reliably known. However, in real-world scenarios, this is rarely the case. If we are not sure how the data are actually distributed, or if we do not know anything about their distribution at all, non-parametric estimation should be preferred. It shows a high level of accuracy in more complex cases, such as the three classes considered in this Lab 2.

Sequential Discriminants

In Lab 2, the method of building sequential discriminants was considered on the example of data from two classes a and b. At each step, two representatives were randomly selected. Based on them, the MED discriminant was built and the clusters of both classes were classified. If the MED (taken from Lab 1) classified the clusters of at least one class completely correctly, then it was added to the sequence, and correctly classified clusters of another class were removed. The action continued until not a single cluster remained in any of the classes. As a result, we obtained a sequence of classifiers, each of which accurately classified a certain part of the clusters of both classes. Then, for the construction of the decision boundary, the classifier was selected from the sequence, which assigned a certain class to the point, and had no errors in assigning clusters to this class.

Question

Learn three sequential classifiers, and for each one plot the resulting classification boundary along with the data points.

Answer

The implementation of such method for constructing classifiers sequence is stored in files **Classify_Point_Seq.m** , **Sequential_Discriminant.m** in the directory **/Sequential_classifier**. The method of classifying plane points and constructing decision boundaries is stored in the file **/Classify_Sequentially**. The results of program execution for 3 discriminants is shown on Figures 13-15:

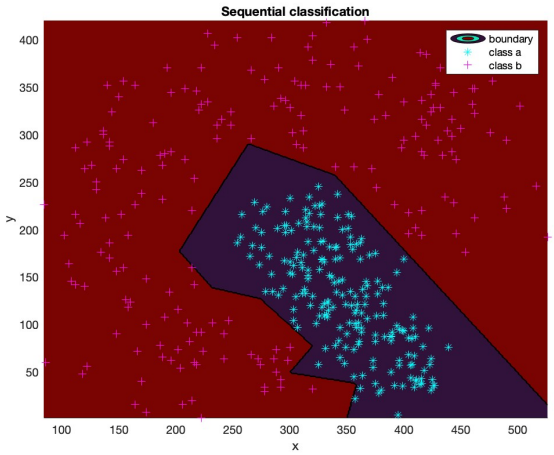

Figure 13: 1 Sequentioal discriminat's decision boundary

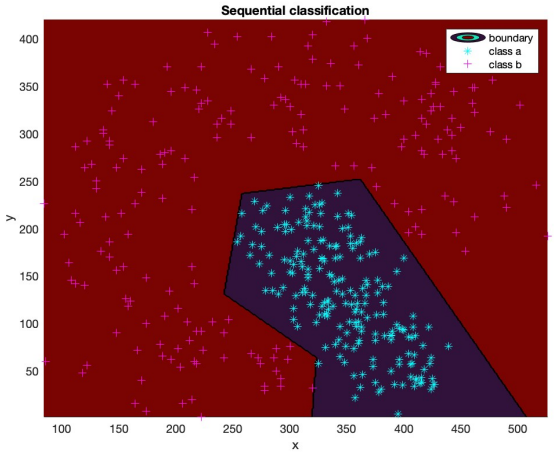

Figure 14: 2 Sequentioal discriminat's decision boundary

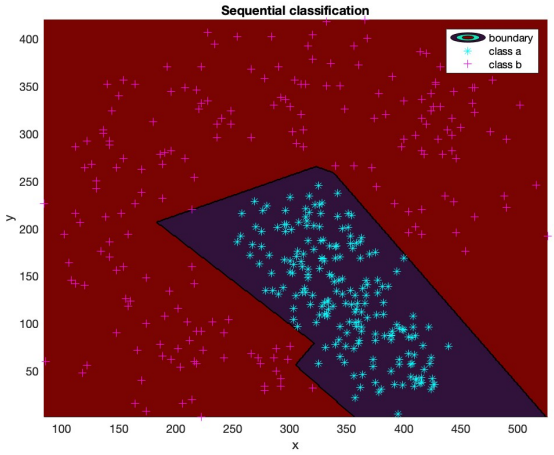

Figure 15: 3 Sequentioal discriminat's decision boundary

Question

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

Answer

The probability of error will be 0. As described above, each classifier from the sequence accurately separates a certain set of clusters, so as a result, decision boundaries will perfectly separate clusters of two classes. If a certain classifier assigns a point to one of the classes, its decision will be accepted only if it has not mistakenly assigned clusters of another class to this class. Thus, the accuracy of the sequence of classifiers is 100%.

Question

We want to see how the experimental error rate varies with J. For each value of J = 1, 2, . . . , 5, learn a sequential classifier 20 times to calculate the following:

- *(a) the average error rate*
- *(b) minimum error rate*
- *(c) maximum error rate*
- *(d) standard deviation of the error rates*

Answer

As described in the task, we estimated the errors that occur when we use a sequence with a fixed number of classifiers and examined the change in error when increasing their number from 1 to 5. For this, 20 classifiers were examined for each J. Below are the plots of changes in the minimum (Figure 16), maximum (Figure 17), and mean values (Figure 18) of the error, as well as the standard deviation of the error (Figure 19):

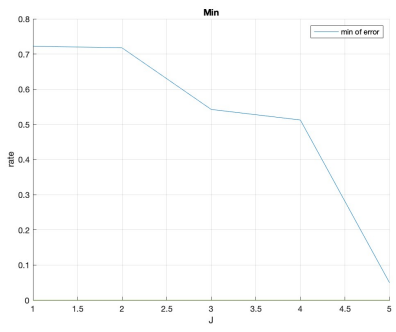

Figure 16: Min error of sequential classifiers as function of J

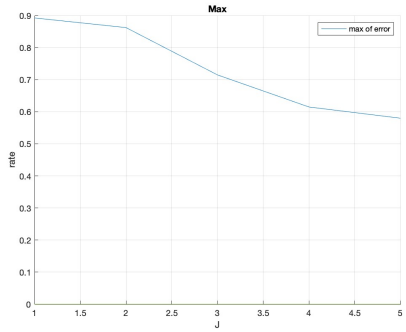

Figure 17: Max error of sequential classifiers as function of J

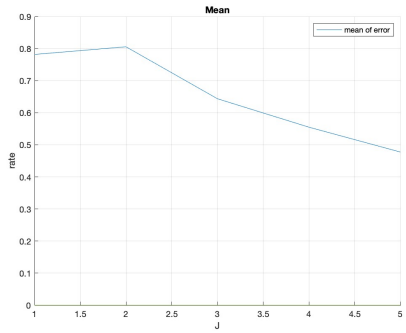

Figure 18: Mean error of sequential classifiers as function of J

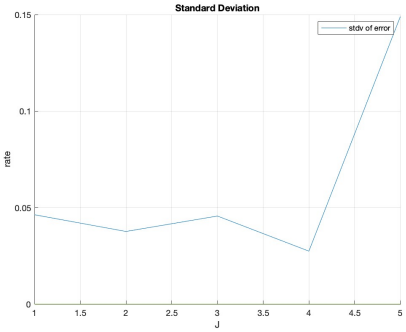

Figure 19: Standard deviation of error of sequential classifiers as function of J

As expected, the minimum, maximum, and mean values of the errors reached their maxima with a small number of classifiers in the sequence and decreased with increasing J. However, the value of the standard deviation increased sharply upon reaching the value j = 5, which means that the resulting errors vary more strongly around their average value. Probably, each of the classifiers well evaluated a rather small set of clusters.

Question

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

Answer

Clearly, by limiting the pairs of points that can be tested, the accuracy of the sequential classifier would decrease because the probability of selecting perfect points would be lower. However, a classifier trained on a limited number of points may eventually show better performance if we choose the training data correctly, since the sequence of classifiers will not over-fit the data so much.

Conclusions

In summary, parametric and non-parametric estimations of datasets in 1D and 2D planes were carried out in the Lab 2. As a result, it can be concluded that parametric estimation makes sense in application only when the real distribution of clusters is known. Otherwise, the wrong selection of the parametric estimation will lead to significant errors, such as the use of an exponential estimation, when the data is normally distributed. Therefore, non-parametric estimation comes to the rescue here. In Lab 2, we got acquainted with the method of Parzen windows, which showed a significant level of efficiency, especially on the example of three classes of 2D space, where the real distribution of data is not known. sequential discriminators trained on the training data give us a 100% accuracy rate, but by limiting their number or the amount of training data, their error will increase, but will not significantly exceed the training data, which will lead to their higher level of fit.