Introduction Machine Learning
Assignment 3

Doga Poyraz TAHAN

March 27, 2018

1. Introduction

1.1 Problem Definition

We were asked to create a parametric classification program based of Bayesian Classification:[1] the tools we had:

$$P(\omega_j|X) = \frac{P(x|\omega_j) * P(\omega_j)}{P(X)} \tag{1.1}$$

$$\log \frac{1}{2 * \pi^{\frac{n}{2}} * |\Sigma|^{\frac{-1}{2}}} * \exp \frac{-1}{2} * (X-\mu)^T * \Sigma^{-1} * (X-\mu) \tag{1.2}$$

$$\mu = \frac{1}{n} * \sum_{k=1}^{n} x_k \tag{1.3}$$

$$\Sigma = \frac{1}{n} * \sum_{k=1}^{n} (X_k - \mu) * (X_k - \mu)^T \tag{1.4}$$

$$g_i(x) = P(\omega_i|x) = \ln p(x|\omega_j) + \ln p(\omega_j) \tag{1.5}$$

1.2 Program Overview

By using these mathematical tools we are able to make a prediction to the new data point. To determine our accuracy and to be sure we are not over-fitting we are going split the data into 2 one being the Train Set and other will be the Test Set. As instructed in the class I will split the given data set first by its class then get the half of the each class data set to form our training set.

2. How To Run the Assignment

1. Open the summited files code into a IDE
2. Find the Main function
3. Change the directory variable according to which data set you want the program to use
4. Put breakpoints on to the functions (release mode works slower)
5. Check the console for the output

[1] Probabilistic approach based on known data

3. Program Structure

3.1 Objects

Program uses 4 self created classes. These classes are related and the DataAll class created it creates the necessary sub classes according to instructions of the instructor.

1. DataAll: holds all of the given data
2. DataSub: holds class based separated data
3. DataTrain: holds the first half of the parent class' data
4. DataTest: holds the second half of the parent class' data

Program highly utilizes the very popular Machine Learning library pandas. Data are hold in data frames [2] to make the use and debugging easier.

3.2 Functions

Program uses total of 3 sub functions and a Main function. These are:

1. getDataSetTable: Reads the given data set
2. probCals: given the inputs finds the Gaussian value
3. gFunc: calculates the probability of the considered point to the respected distribution
4. covMatrix: calculates the covariance Matrix as requested from the instructor
5. plotty1: plots the dataframe to coordinate sytem
6. paintProbArea: paints the spheres of influence by the distribution. To make better shaded area one can increase the increment number but it will take a while.
7. prediction: a utility function for the coder there could have been a better solution without this function but the time constraint forced me to. This function gives the estimated class for each point to shade

4. Solution

4.1 Explanation

The program was highly accurate, every data point has been correctly classified. For the 4 class case, I used the same convention and get the half of the data as Test and the other half as the Train. It classified correctly $\frac{3977}{4000}$. You can see the console outputs in the next sub section. For unknown reasons to me the program runs way faster in the debug mode.

[2] data frame: rectangular indexed format for holding large amount of data

4.2 Console Outputs

```
Class: 1.0 Mean Vector:
          0
0  2.032682
1  3.139692

Class: 1.0 Covarience Matrix:
          c1        c2
c1  0.952139  1.398795
c2  1.398795  3.828466

Class: 2.0 Mean Vector:
          0
0  9.923056
1  3.009518

Class: 2.0 Covarience Matrix:
          c1        c2
c1  1.026438  0.820864
c2  0.820864  0.991811

Data: ./data/two_class/data1.txt > Accuracy (%): 100.0 [Errors:  0 / 1000 ]
```

Figure 1: data set 1 -

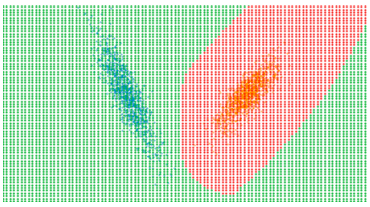

Figure 2: data set 1 - shaded

```
Class: 1.0 Mean Vector:
          0
0  2.015978
1  3.008652

Class: 1.0 Covarience Matrix:
          c1        c2
c1  0.966774 -1.432574
c2 -1.432574  2.864228

Class: 2.0 Mean Vector:
          0
0  9.942320
1  2.947054

Class: 2.0 Covarience Matrix:
          c1        c2
c1  1.000884  0.780257
c2  0.780257  0.953542

Data: ./data/two_class/data2.txt > Accuracy (%): 100.0 [Errors:  0 / 1000 ]
```

Figure 3: data set 2 - output

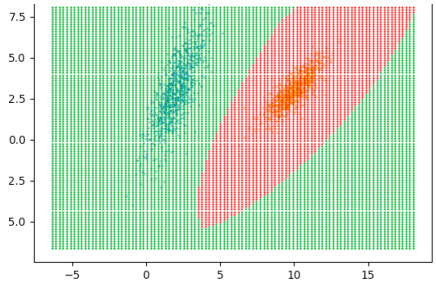

Figure 4: data set 2 - shaded

```
Class: 1.0 Mean Vector:
          0
0  2.044158
1  3.021696

Class: 1.0 Covarience Matrix:
          c1        c2
c1  0.937321 -0.037423
c2 -0.037423  3.117510

Class: 2.0 Mean Vector:
          0
0  7.082960
1  3.027272

Class: 2.0 Covarience Matrix:
          c1        c2
c1  1.010234  0.026823
c2  0.026823  0.902132

Data: ./data/two_class/data3.txt > Accuracy (%): 99.5 [Errors:  5 / 1000 ]
```

Figure 5: data set 3 - output

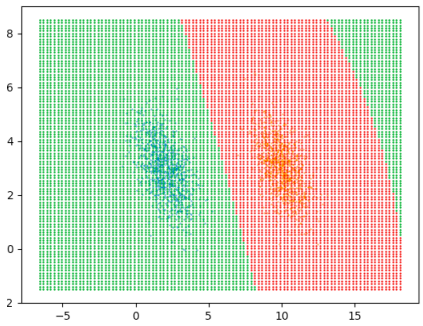

Figure 6: data set 3 - shaded

```
Class: 1.0 Mean Vector:
          0
0  1.953756
1  3.025828

Class: 1.0 Covarience Matrix:
          c1        c2
c1  1.012603 -0.518514
c2 -0.518514  0.956618

Class: 2.0 Mean Vector:
           0
0  10.045476
1   3.024334

Class: 2.0 Covarience Matrix:
          c1        c2
c1  0.952370 -0.518088
c2 -0.518088  1.084563

Data: ./data/two_class/data4.txt > Accuracy (%): 100.0 [Errors:  0 / 1000 ]
```

Figure 7: data set 4 - output

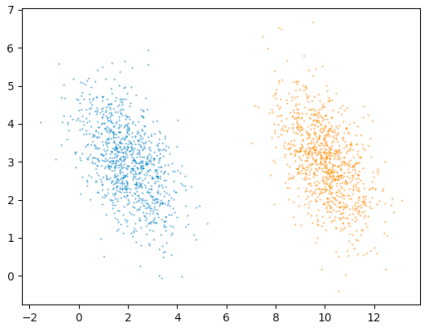

Figure 8: data set 4 - plain view

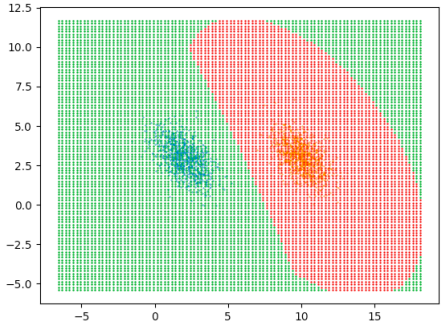

Figure 9: data set 4 - shaded

```
Class: 1.0 Mean Vector:
           0
0  2.070160
1  2.965666

Class: 1.0 Covariance Matrix:
          c1        c2
c1  1.749892  0.082419
c2  0.082419  0.485993

Class: 2.0 Mean Vector:
           0
0  9.987644
1  3.032178

Class: 2.0 Covariance Matrix:
          c1        c2
c1  1.983730 -0.020666
c2 -0.020666  0.483153

Data: ./data/two_class/data5.txt > Accuracy (%): 99.9 [Errors:  1 / 1000 ]
```

Figure 10: data set 5 - output

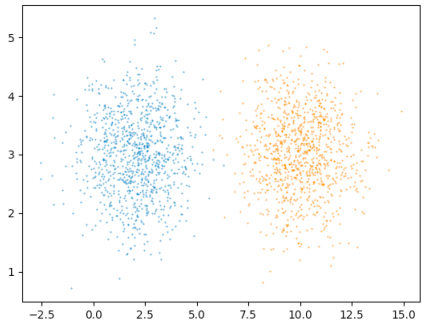

Figure 11: data set 5 - plain view

```
Class: 1.0 Mean Vector:
           0
0  1.893316
1  3.070810

Class: 1.0 Covariance Matrix:
          c1        c2
c1  0.954488 -0.070486
c2 -0.070486  1.005676

Class: 2.0 Mean Vector:
           0
0  10.072870
1   2.942782

Class: 2.0 Covariance Matrix:
          c1        c2
c1  1.113091 -0.075734
c2 -0.075734  0.965138

Data: ./data/two_class/data6.txt > Accuracy (%): 100.0 [Errors:  0 / 1000 ]
```

Figure 12: data set 6 - output

```
Class: 1.0 Mean Vector:
          0
0  2.046906
1  2.903180

Class: 1.0 Covarience Matrix:
          c1        c2
c1  0.961078 -1.351759
c2 -1.351759  2.586535

Class: 2.0 Mean Vector:
          0
0  4.937686
1  1.957730

Class: 2.0 Covarience Matrix:
          c1        c2
c1  0.890506  0.694622
c2  0.694622  0.892728

Data: ./data/two_class/data7.txt > Accuracy (%): 95.5 [Errors:  45 / 1000 ]
```

Figure 13: data set 7 - output

```
Class: 1.0 Mean Vector:
          0
0  2.112842
1  3.024152

Class: 1.0 Covarience Matrix:
          c1        c2
c1  6.355614 -1.362574
c2 -1.362574  5.106024

Class: 2.0 Mean Vector:
          0
0  0.001556
1  8.017904

Class: 2.0 Covarience Matrix:
          c1        c2
c1  1.177852  0.763210
c2  0.763210  0.998029

Data: ./data/two_class/data8.txt > Accuracy (%): 97.8 [Errors:  22 / 1000 ]
```

Figure 14: data set 8 - output

```
Class: 1.0 Mean Vector:
          0
0  1.881014
1  3.146992

Class: 1.0 Covarience Matrix:
          c1        c2
c1  4.421913 -0.054189
c2 -0.054189  6.195426

Class: 2.0 Mean Vector:
          0
0  9.991668
1  3.025162

Class: 2.0 Covarience Matrix:
          c1        c2
c1  0.984843  0.007742
c2  0.007742  0.912557

Data: ./data/two_class/data9.txt > Accuracy (%): 99.3 [Errors:  7 / 1000 ]
```

Figure 15: data set 9 - output

```
Class: 8.0 Mean Vector:
          0
0  1.989656
1  3.042244

Class: 8.0 Covarience Matrix:
          c1        c2
c1  1.671593  0.519637
c2  0.519637  1.620192

Class: 4.0 Mean Vector:
          0
0  2.017978
1  2.976406

Class: 4.0 Covarience Matrix:
          c1        c2
c1  4.027109 -0.089080
c2 -0.089080  3.483958

Data: ./data/two_class/data10.txt > Accuracy (%): 67.8 [Errors:  322 / 1000 ]
```

Figure 16: data set 10 - output

```
Class: 8.0 Mean Vector:
          0
0  1.946756
1  2.977318

Class: 8.0 Covarience Matrix:
          c1         c2
c1  9.457185   7.129685
c2  7.129685  10.788028

Class: 4.0 Mean Vector:
          0
0  1.787372
1  3.028242

Class: 4.0 Covarience Matrix:
          c1        c2
c1  3.859816 -0.083663
c2 -0.083663  3.737398

Data: ./data/two_class/data11.txt > Accuracy (%): 67.8 [Errors:  322 / 1000 ]
```

Figure 17: data set 11 - output

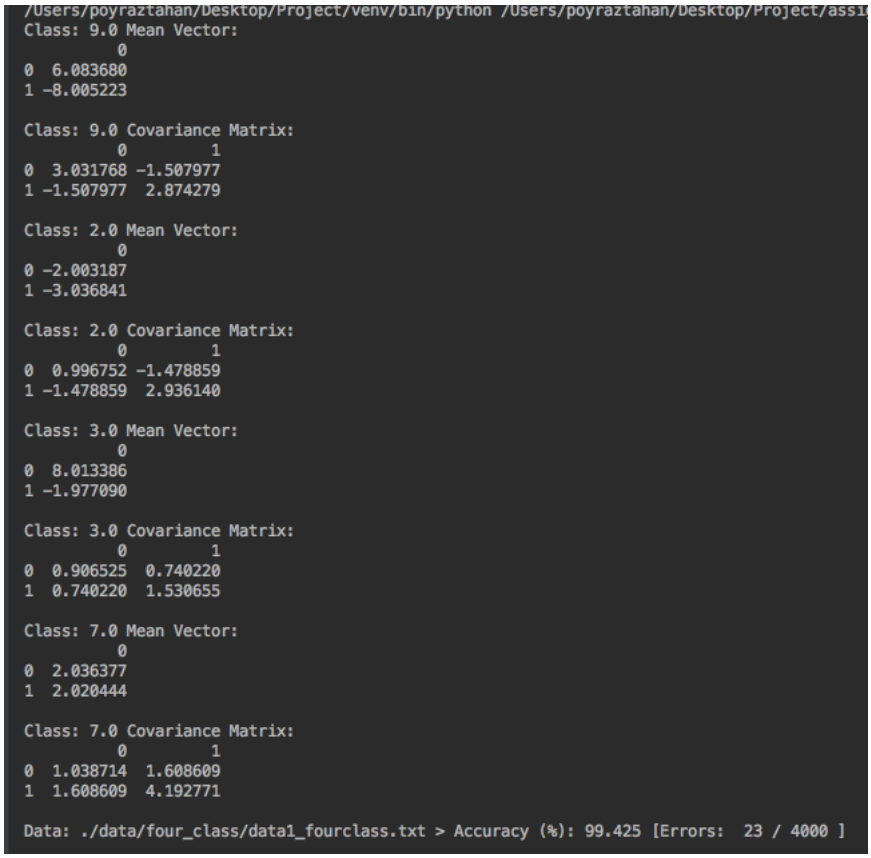

Figure 18: data set 1 - 4 class output part 1

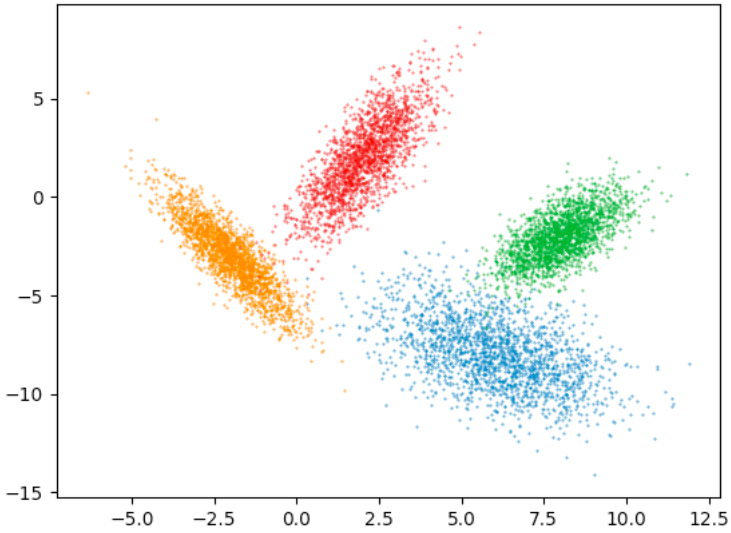

Figure 19: data set 1 - 4 class plain view

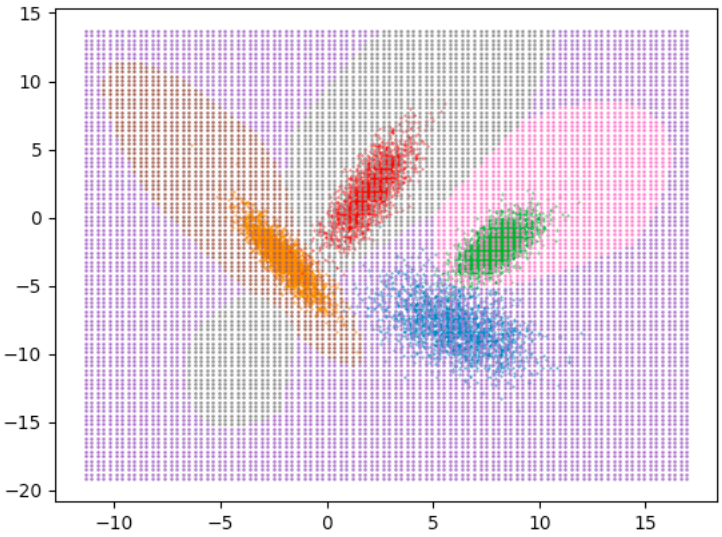

Figure 20: data set 1 - 4 class shaded