MACHINE LEARNING FROM DATA
Fall 2018

Report: Lab Session 1 – MAP and Gaussian data
Classification criteria based on maximizing posterior probability

Names: Santagiustina Francesco, Simonetto Adriano
Group:

1. Instructions

- Answer the questions
- Save the report and upload the file

2. Questions

Q1: Find the eigenvalues of the matrix $\mathbf{C} = \frac{\sigma^2}{2}\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ as a function of the parameters ρ and σ^2
The two eigenvalues are:

$$\lambda_{1,2} = \frac{\sigma^2}{2}(1 \pm \rho)$$

Q2. Read the script `lab1_gauss3.m`, analyse the code, identify the most relevant parts. Describe briefly each part.

Parameter initialization: Set-up phase where the parameters as number of classes and features, number of samples, Energy and noise are initialized.

Dataset generation: Based on the previous parameters, the features are generated with their corresponding labels.

Histograms: Creation of the various histograms for each class-feature couple. Each of the histograms is compared to the shape of a Gaussian with same mean and variance.

Scatter plot: Generation of two different plots: the first one compares each couple of features, while the second is a 3-D plot where the three features are considered at the same time (each feature corresponds to one of the three axis).

Create a default (linear) discriminant analysis classifier: Here there is the creation of a linear classifier for the data and the display of the associated error on the training set.

Create a default quadratic discriminant analysis classifier: Same as in the previous section, only that this time the classifier is quadratic.

ROC and CONFUSION MATRIX: Plot of the ROC for both linear and quadratic classifiers and computation of their confusion matrices.

Quadratic Mahalanobis distance: Computation of the two Mahalanobis distances between the two classes for both the classifiers.

Q3. Create a table including error probabilities obtained by the linear classifier (LC) and error probabilities obtained by the quadratic classifier (QC), for each SNR value. Discuss the results.

	3 dB	0 dB	-3 dB	-10 dB
LC	0.0035	0.0039	0.124	0.274
QC	0.005	0.0385	0.123	0,2725

As it is possible to expect the probability of error increases as the SNR decreases. Moreover, the probability of error results really similar between linear and quadratic classifier. This is not surprising as this kind of data is best separated with a linear classifier, which means that a quadratic one is not going to be more helpful (but at the same time of course is not going to be worse).

Q4. Include in the report the confusion matrices obtained for SNR=-10db and SNR=-3dB and the two classifiers. Discuss the results.

For $SNR = -3dB$ we have:

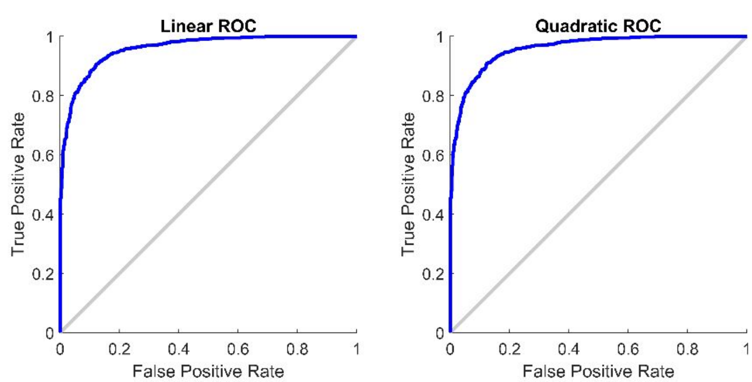
$$CM_l = \begin{bmatrix} 873 & 127 \\ 121 & 879 \end{bmatrix} CM_q = \begin{bmatrix} 873 & 127 \\ 119 & 881 \end{bmatrix}$$

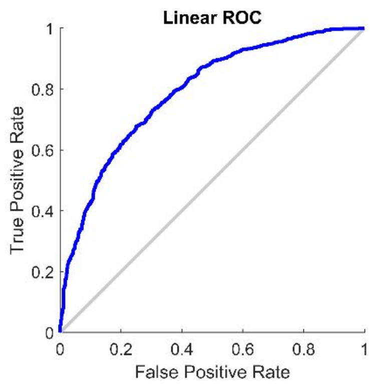
and for $SNR = -10dB$ instead:

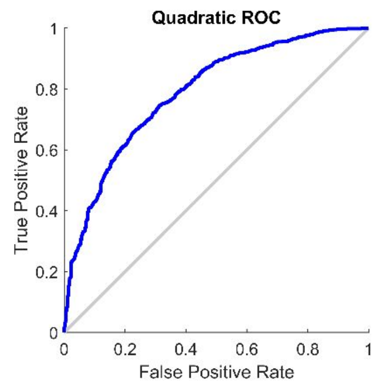
$$CM_l = \begin{bmatrix} 727 & 273 \\ 275 & 275 \end{bmatrix} CM_q = \begin{bmatrix} 719 & 281 \\ 264 & 736 \end{bmatrix}$$

From the confusion matrices we can see how a higher amount of noise increases the amount of class 0 elements mistaken for class 1 elements and vice versa. As we can expect from the symmetry of the problem, the amount of elements mistaken in one direction or the other is roughly the same. At last we can see how using a quadratic classifier instead of a linear one does not improve the performance as for the considered problem, the best way to distinguish between the two classes is linearly.

Q5. Include in the report the ROC curves obtained for SNR=-10db and SNR=-3dB and the two classifiers. Discuss the results.







The two pictures correspond respectively to the -3dB case and to the -10 dB case. In an ideal scenario the ROC would be composed by two vertical lines: one from (0,0) to (0,1) and the second from (0,1) to (1,1). The reason for that is the following: the best case scenario is the one without errors of any kind, which means not mistaking any element of the other class 0 to class 1 (False Positive Rate equal to 0), and correctly detecting all elements of class 1 (all the elements of the class are detected correctly), which is exactly what is represented by the two previously mentioned lines.
In practice however, the noise prevents such perfect results, and better results from the point of view of the True Positive Rate can only be achieved at the cost of a higher False Positive Rate.
Moreover the higher the noise, the further this negative tradeoff gets, as it is possible to see comparing the two sets of pictures: as we shift from -3dB to -10dB we can easily notice how the curve tends to drop over the bisector of the quadrant.
It is instead possible to notice that the performance does not change (as expected) between the two classifiers, for the same reasons explained before.

Q6. Compute the Mahalanobis distance between the two classes using `mahal.m`. Explain why the result differs depending on how this function is called (depending on the order of the parameters).

The Mahalanobis distance computed for SNR = -3dB is: d01 = 9.3875, d10 = 9.4582.
The reason behind this difference lies on the definition of the distance. The point is that for each of the classes the distance is defined based on the elements of the class, related to mean and covariance matrix of the other class. This means that if for example we kept the set of vectors of class 0 still but changed the set of class 1 maintaining the same mean and variance (e.g. one vector or N vectors all identical to the first one), the value of d01 would not change, while the other would vary..

QPSK and covariances of all classes identical but arbitrary (case 2)

Q7. Read the script `lab1_QPSK.m,` analyse the code, identify the most relevant parts.

Parameter initialization: Set-up phase where the parameters as number of classes and features, number of samples, Energy and noise are initialized.

Dataset generation: Based on the previous parameters, the features are generated with their corresponding labels.

Create a default (linear) discriminant analysis classifier: Here there is the creation of a linear classifier for the data and the display of the associated error on the training set.

Create a quadratic discriminant analysis classifier: Same as in the previous section, only that this time the classifier is quadratic.

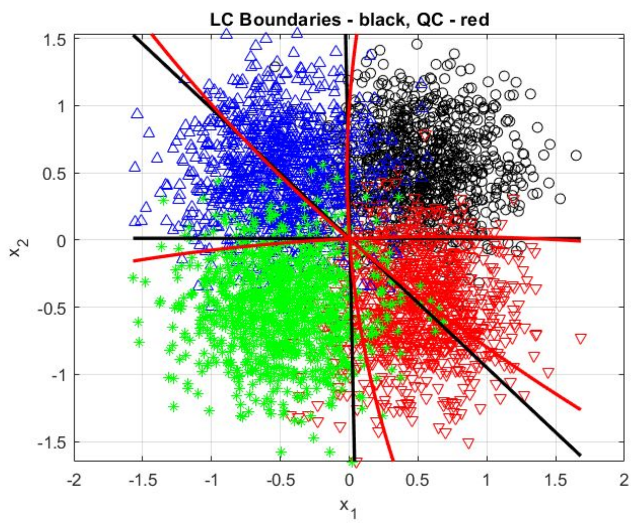
Scatter plot: Generation of the plot representing the data points on the features plane (representing the complex plane)

Plot the LINEAR classification boundaries for the class 1: the linear decision boundary between the first and each other class is plotted using the parameters extracted from the linear classifier.

Plot the QUADRATIC classification boundaries for the class 1: the quadratic decision boundary between the first and each other class is plotted using the parameters extracted from the linear classifier.

Confusion matrices: Computation of the confusion matrices for both linear and quadratic classifiers.

Q8. Include the scatter plot, decision boundary, confusion matrices and error probabilities obtained using the linear classifier (LC) and the quadratic classifier (QC) for $\rho = 0$. Compare the metrics for the two classifiers and discuss the results.



Using SNR = 3 dB we get:

error_Linear = 0.147

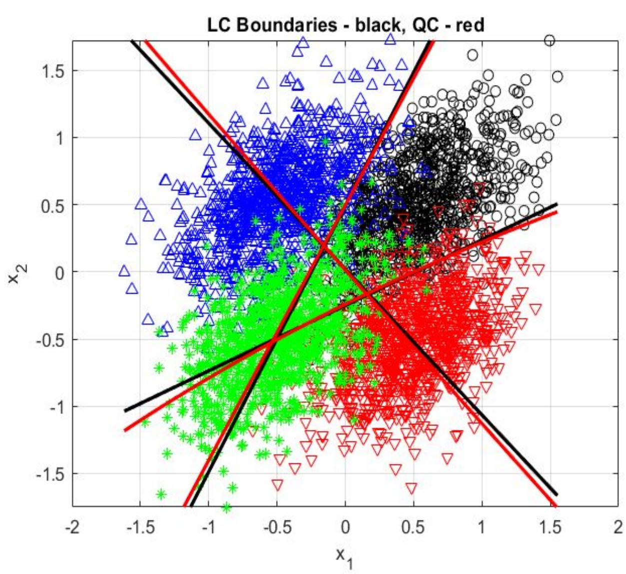
error_Quadratic = 0.146

which are very similar. Indeed the quadratic can't take a big advantage as the optimal decision regions are almost correspondent to the four quadrants of the plane, which are defined by linear boundaries.

The confusion matrices are :

$$CM_l = \begin{bmatrix} 860 & 69 & 63 & 8 \\ 78 & 841 & 6 & 75 \\ 66 & 7 & 850 & 77 \\ 5 & 69 & 65 & 861 \end{bmatrix} \quad CM_q = \begin{bmatrix} 861 & 69 & 64 & 6 \\ 75 & 843 & 6 & 76 \\ 69 & 7 & 850 & 74 \\ 5 & 66 & 67 & 862 \end{bmatrix}$$

Q9. Repeat the previous analysis (Q8) for $\rho = 0{,}5$. Compare the metrics for the two classifiers and discuss the results.



Using SNR = 3 dB we get:

error_Linear = 0.11575

error_Quadratic = 0.1155

which are again very similar. Even if the decision region are now less symmetric the quadratic boundaries keep to stick almost totally to the linear boundaries, thus resulting in equal classifications for most of the cases and thus similar errors.

The confusion matrices are are also similar :

$$CM_l = \begin{bmatrix} 869 & 44 & 42 & 45 \\ 48 & 910 & 0 & 42 \\ 48 & 0 & 895 & 57 \\ 52 & 44 & 41 & 863 \end{bmatrix} \quad CM_q = \begin{bmatrix} 870 & 41 & 43 & 46 \\ 47 & 910 & 0 & 43 \\ 47 & 0 & 897 & 56 \\ 53 & 45 & 41 & 862 \end{bmatrix}$$

Q10. Compare and discuss the results obtained in Q8 and Q9

We can see that by adding a correlation between the components the distributions now follow an elliptic shape. This affect the error probability of both the linear and the quadratic classifier but not in a drastic way. Confusion matrices are also similar, but we can point out the following difference. In the $\rho = 0$ case all terms in the antidiagonal are very low but not zero. This because they correspond to misclassifications between classes which are opposed with respect to the origin of the feature plane (like black and green, or red and blue clusters) and not adjacent. By introduce a covariance term ($\rho = 0.5$) we obtain a larger variance along the antidiagonal direction for black and green clusters making them overlap around the origin, this result in larger counts for the class1/class2 confusion coefficients. At the same time the clusters of class 2 and 3 are shrunk along the antidiagonal direction so they no longer overlap around the origin, thus resulting in null class2/class3 confusion coefficients.

QPSK and different covariance matrices (case 3)

Q11. Include the error probabilities obtained using the linear classifier (LC) and the quadratic classifier (QC) for SNR = +5 dB and +10 dB. Compare the metrics for the two classifiers and discuss the results.

	5 dB	10 dB
LC	0.06325	0.003
QC	0.059	0.003

We can see that the quadratic classifier performs better than the linear one only in the 5 dB SNR case and both are equal in the 10 dB SNR case. This can be explained by the fact that even if now the quadratic classifier really adapts its boundaries to the covariance matrices of each class, this is only needed when the different ellipses are close to each other that is for low SNRs. At 10 dB the different clusters are sufficiently distant to effectively separate them by simple linear boundaries.

Q12. Complete the table with the theoretical eigenvalues obtained in Q1, and the eigenvalues computed using the class covariance matrices with the function

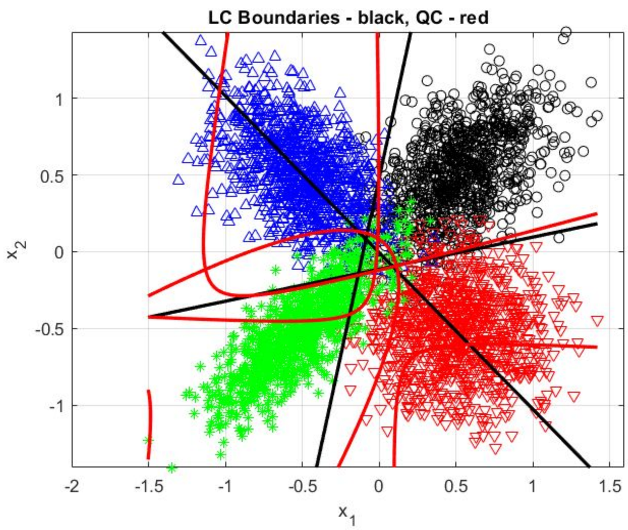
```
Autoval=eig(squeeze(M_covar(:,:,c)))
```

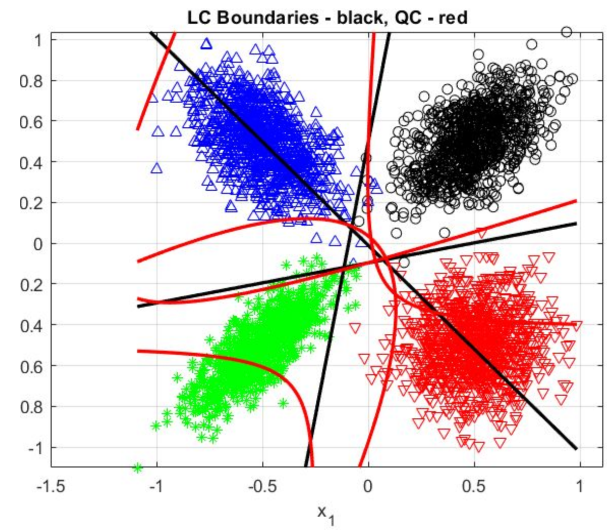
	Class 1	Class 2	Class 3	Class 4
Theoretical eigenval	1/80 3/80	1/40 1/40	1/80 3/80	1.8/40 0.2/40
`eig` eigenvalue	0.0125 0.0375	0.0250 0.0250	0.0125 0.0375	0.0050 0.0450

These results correspond to a SNR = 10 dB = 10 which implies a variance $\frac{\sigma^2}{2} = \frac{1}{2}\frac{E}{SNR} = \frac{1}{4}\frac{1}{SNR} = \frac{1}{40}$.

Q13. Include scatter plots for the linear and quadratic classifiers using SNR= +5 dB and SNR= +10 dB. Relate the shape of the clusters with the eigenvalues of the covariance matrices.

The shape of clusters is elliptic and each eigenvalue is proportional to the length of the axis whose direction is determined by the corresponding eigenvector. So, class 2 (in red) has for instance a circular shape, given that its two eigenvalues are equal while class 4 (in green) has the more stretched shape as the higher eigenvalue is nine times the smaller one. Clusters associated to class 1 (in black) and class 3 (in blue) have the same shape with only different orientation, because their eigenvalues are equal but eigenvectors have different directions. Given the constant Energy of the symbols a higher SNR correspond to lower variance and thus more condensed clusters.





Q14. Include error probabilities, scatter plots and decision boundaries. Compare the performance of the classifier and justify the results.

Using new_M_covar(1)= 30*M_covar(1) we get :

error_Linear = 0.129

error_Quadratic = 0.05125

this time there is a significative improvement of the quadratic classifier over the linear one. The scatter plots and decision boundaries allow us to understand why easely : points of class 1 are so spread out that we find many of them beyond the clusters of the other classes and only quadratic boundaries are able to enclose the clusters of other classes. In this way, points who are closer to the mean of another class but too far away from it to probably come from their distribution given its limited variance will be (often correctly) classified as instances of class 1.