Grenoble INP - Ense3

AI and Autonomous systems

Lab 2 - M2-MARS

Discrimant Analysis

Author :
Souhaiel BEN SALEM

26 octobre 2021

Lab Objectives: The objective of this lab is to illustrate discriminant analysis (LDA + QDA) and naïve Bayes methods on both synthetic and computer vision (handwritten digits) datasets.

Introduction:

Discriminant analysis (DA) is a multivariate technique that divides a set of items, individuals, or entities into two or more groups or categories based on a set of specified variables relating to their qualities, types, or any other attributes. This method can also be used to determine which variables play a role in classification.
The objectives of DA are:

- Creating a discriminating function is the first step. This function (a mathematical equation) is used to distinguish between individuals in the population and assign them to one of several groups. A sequence of measurements (predictor variables) or observations on the individuals are used to create the function.
- To know whether significant differences exist among the groups, in terms of the predictor variables.
- Finding out which predictor variables contribute to most of the inter-group differences.
- Evaluating the accuracy of classification

1 Notebook 1: LDA binary classification for synthetic data

Exercice:

1. **Complete the code below (see FIXME tags) to compute the estimators of the mean vectors**
   To estimate the mean of sample class, we use the **numpy.mean()** which take two parameters generally: the array-like type containing numbers whose mean is desire and the axis along which the means are computed. In our, case we have two samples corresponding to either **Y=1** or **Y=2**. The first training sample (**Xtrain[Ytrain == 1]**) is a (22x2) numpy array whereas the second one (**Xtrain[Ytrain == 2]**) is a (78x2) numpy array (two dimentional data). To get the class mean vectors estimation, we must compute the mean along each column i.e we must set **axis=0**:

```
# Get class sizes
n1 = np.sum(Ytrain == 1)
n2 = np.sum(Ytrain == 2)
print((2*"number of class {} samples (training set) n{}={}\n").format(1, 1, n1, 2, 2, n2))

# estimate the class weight
pik = [n1/n, n2/n]
print((2 * "class weight pi{}={}\n").format(1, pik[0], 2, pik[1]))

# estimate the class mean vectors
mu1 = np.mean(Xtrain[Ytrain == 1],axis=0)  # FIXME: put the right expression here as a 2D vector
mu2 = np.mean(Xtrain[Ytrain == 2],axis=0)  # FIXME: put the right expression here as a 2D vector

print((2 * "class mean mu{}={}\n").format(1, mu1, 2, mu2))

# Compute pooled covariance (see LDA assumption) estimator
Xcentered = [Xtrain[Ytrain == 1]-mu1, Xtrain[Ytrain == 2]-mu2]

# @ for matrix multiplication
SigmaHat = (1./(n-2)) * (Xcentered[0].T @
                         Xcentered[0] + Xcentered[1].T @ Xcentered[1])

print("Pooled covariance estimator SigmaHat")
S_str = np.array2string(SigmaHat, precision=3, separator=', ')
print(' ' + S_str[1:-1])
print(Xtrain[Ytrain == 1].shape)
print(Xtrain[Ytrain == 2].shape)
```

Figure 1: mean vectors estimtion

2. **Complete the code below (see FIXME tags) to compute the decision boundary**
The linear term is given by:

$$L_{k,l} = \hat{\sum}^{-1} (\hat{\mu_k} - \hat{\mu_l})$$

We just need to implement this equality into our code and plug in our mean estimates and the inverse of the co-variance matrix:

```
# parameter of the LDA decision boudary: here a simple line

# Constant term (see course)
C = np.log(pik[0]/pik[1]) - 0.5 * mu1.T @ np.linalg.inv(SigmaHat) @ mu1 + \
    0.5*mu2.T @ np.linalg.inv(SigmaHat) @ mu2
print("Constant term C={}".format(C))

# Linear term (see course)
L =  np.linalg.inv(SigmaHat) @ (mu1-mu2) # FIXME: put the right expression here as a 2D vector
print("Linear term L={}".format(L))
```

```
Constant term C=-10.21618942328174
Linear term L=[-5.08154836  4.25066671]
```

Figure 2: Linear term estimation

After setting up the decision rule, here is the decision boundary obtained:

```
Text(0, 0.5, '$x_2$')
```

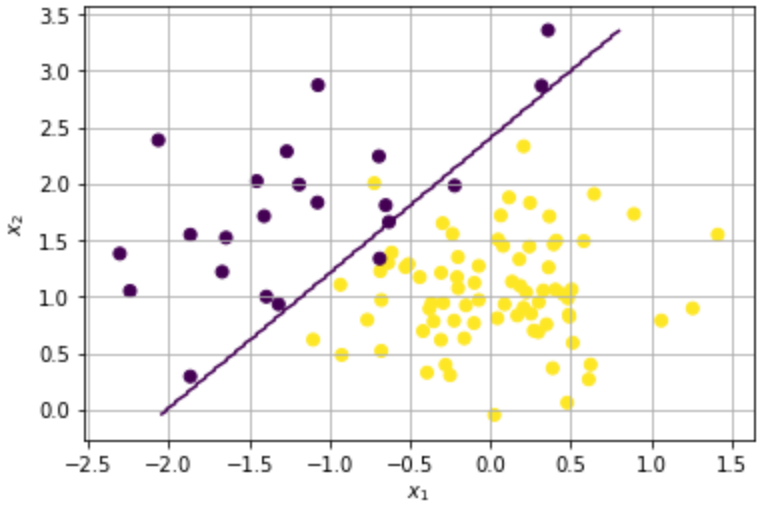

Figure 3: Data separation with the projected discriminant function

3. **Compare with built-in methods from sklearn**
We got almost the same result as those obtained by using sklearn's built-in methods
$\Rightarrow$ *the same decision boundary*

4. **check LDA decision boundary/classification rule an compute performances**
   the LDA model fits the data well as both the training misclassification error rate and test misclassification error rate are low. This is expected since we are working with a small dataset and two classes that are clearly linearly separable since the distance between mean of each sample is sufficient enough.

```
LDA training misclassification error rate = 0.030000000000000027
LDA test misclassification error rate = 0.05500000000000005

Text(0, 0.5, '$x_2$')
```

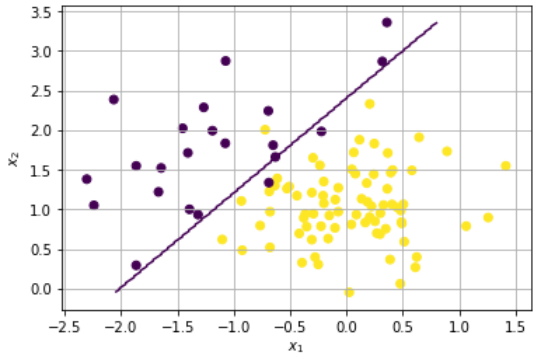

Figure 4: error rates and decision boundary

5. **QDA**:
   We also notice that the QDA performs quiet well on this dataset. This is expected since the complexity of the QDA is higher than that of LDA. In fact, LDA tends to be more reliable for small datasets where reducing the variance is crucial.

```
QDA training misclassification error rate = 0.050000000000000044
QDA test misclassification error rate = 0.050000000000000044

Text(0, 0.5, '$x_2$')
```

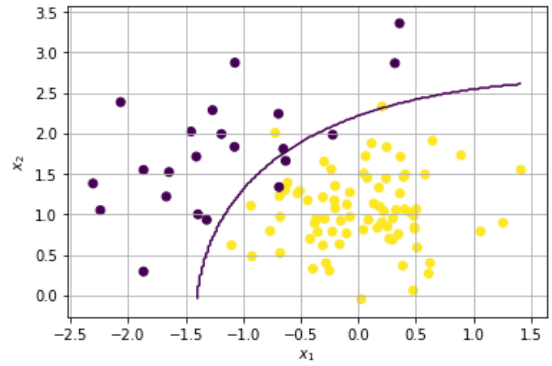

Figure 5: QDA error rates and decision boundary

6. **Why the QDA boundary is no longer a straight line?**
   The QDA assumes that the covariance for each class is different from the other (unlike the LDA model). This assumption leads to the introduction of a quadratic term in the discriminant function which is why the boundary is no longer a straight line but a parabolic-like function.

7. **Which model (LDA or QDA) should be preferred on this small dataset?**
   For small datasets, LDA is should be preferred over QDA mainly because QDA might overfit for small datasets when the number of sample points are less than the number of parameters whereas LDA is less flexible and has substantially lower number of parameters.
   QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix is clearly untenable.

2 Notebook 2: Handwritten digits recognition

Exercise I.1:

1. **Can you explain why sklearn algorithm raises a warning when we train the QDA and LDA models (and not in the QDA+NB and LDA+NB case)?**
   LDA and QDA, like regression techniques involves computing a matrix inversion ($\hat{\sum}^{-1}$) , which is inaccurate if the determinant is close to 0 (i.e. two or more variables are almost a linear combination of each other which is expected from this specific dataset) which is why sklearn is raising the warning "Variables are collinear".
   Naive Bayes assumes one feature is not correlated with any other feature. Having multicollinearity i.e. two or more features carrying the same information will not affect Naive bayes because it assumes presence of one feature is independent of presence or absence of any other feature, which is based on bayes theorm of conditional probability. In other words, we don't have a matrix inversion computation when use use NB + QDA or NB + LDA and therefore sklearn does not raise a warning in this case.

2. **How to justify that the linear discriminant analysis under the Naïve Bayes assumption seems the most appropriate among all the methods of discriminant analysis?**
   LDA, when used under Naive Bayes assumption is the most appropriate among all the methods of discriminant analysis because, for starters, LDA has less parameters than QDA which makes it more efficient and less likely to overfit especially for small datasets (QDA does not generalise very well). In addition, if ww use LDA under the NB assumption, the number of parameters becomes even smaller ($numberof\,parameters = K \times p + p + (k+1)$

   - $K \times p$: means
   - p : number of parameters of the covariance matrix which is diagonal
   - $(k+1)$: prior probabilities

Exercise I.2:

1. **Do these synthetic examples seem realistic to you?**
   The synthetic data is not easily visually interpretable and does not seem realistic.
2. **What is the interest of such a model?** This type of model (generative models) can be used to generate new data from the parameters learnt from the training set.The model learns estimates for the conditional probabilities with which each value of each feature variable occurs given a class label (category). These conditional probabilities can be denoted $P(xi|y)$. The data generated may not be realistic but it can be used for dataset augmenting for example if the NB assumptions are met.
3. **Comparing with QDA based synthetic examples on the cell below, what can we conclude (remember that QDA obtains here catastrophic generalization performances on test data)?**
   The data generated using QDA is more realistic even though its performance on test data was poor. The conclusion is that QDA was a subject of overfitting during the training process since its error rate = 0 for training data.

Exercise II.1:

1. **Explain what happens in the limiting case γ=0 or γ=1 ? And which methods correspond to these particular cases of regularized discriminant analysis?**
   - When γ=0 : we use diagonal entries equal to those of $\hat{\sum}$ (Naïve Bayes empirical estimate) : RDA reduces to NB.
   - When γ=1 : we only use the empirical pooled covariance matrix : RDA reduces to LDA.
2. **What are a good choices here for the γ values?**
   Any value of γ in the range [0.2,0.6] is a good value because these values minimize misclassification rate.
3. **In practice, when there is no test set, which common procedure can we use to estimate the optimal value of γ? (note: we will see below a performant and cost-efficient alternative)**
   To estimate the optimal value of γ when there is no test set we can use **cross validation**.
4. **Compare with the performance/computational cost obtained for a k-NN classifier**
   K-NN is more computational heavy than RDA as it requires to stock all the training data and calculate the distances while RDA reduces the dimention of a given problem.

Exercise II.2:

1. **Is this in good agreement with the estimates of the optimal regularized LDA performance derived previously?**
   The results given by the Auto regularized LDA mirror those obtained earlier as we got almost the same miclassification rate for both training and testing (mcr for training : $0.008 \approx 0$ and mcr for testing:0.188 is close to the mcr=0.172 found earlier).

2. **What are the benefits of this 'automatic' method compare to cross-validation?**
   The automatic method is easier to implement and easier to compute. Cross validation results depend on the data segmentation. CV also needs alot of computation time and resources.

Exercise II.3:

1. **Apply (see below) regularized LDA to these larger datasets and compare the performances with the previous one. What can you conclude?**
   When we increased the training and testing datasets, we see that the mcr of Auto regularized LDA is even lower for both the training and testing sets. This proves that Auto regularized LDA is robust and effective for large datasets.
2. **Compare now the optimal values for the regularization parameter γ. How to explain this?**
   For this experiment which involves a large dataset, the shrinkage coefficient γ is almost 0. This means that the algorithm emphasises the Naive Bayes approach more than the LDA approach because NB is more more suitable for multi-class classification problems.

Exercise II.4:

1. **What are the the most common confusions between classes?**
   The most common confusions are between 4 & 9 and 5 & 3.

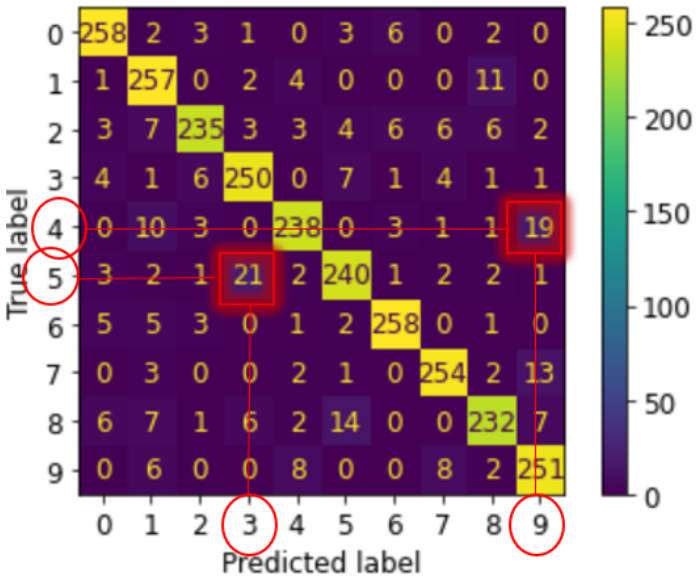

Figure 6: common confusions

3 Notebook 3: Linear and Quadratic Discriminant Analysis with covariance ellipsoid

Exercise:

1. **Which model (LDA or QDA) should be preferred in the first example (first row)?**
   For data with fixed covariance, LDA should be preferred over QDA because LDA assumes fixed covariance by default whereas QDA assumes different covariance between the classes. QDA would give a good result however the computation would be more efficient using LDA.
2. **What kind of functions define the decision boudary for QDA in the second example (second row)? Is this boudary in agreement with the data?**
   In the second row, the decision boundary for QDA is decided by discriminative quadratic functions, hence the parabolic separation boundaries. These boundaries separate the data in an understandable manner (better separation than LDA).

4 Notebook 4: Normal and Shrinkage Linear Discriminant Analysis for classification

Exercise:

1. **why the performance of standard (without regulariaztion/shrinkage) LDA decreases with the number of features?**
   Increasing the dimension of the dataset and adding noise can lead to the algorithm being less robust especially if the number of classes exceeds the number of effective parameters (or even approaches the number of effective parameters). **Moreover, when we have high dimensional data, we need enormous training datasets in order to learn all the features (curse of dimentionality).**
2. **why the regularization/shrinkage allows us to mitigate the curse of dimensionality?**
   The regularization/shrinkage technique helps improving the generalization performance of the classifier by shrinking the predictors that are not informative of the response to 0. Basically, the Auto regularized LDA focus only on samples with high covariance which is more data-representative . RDA limits the separate covariance of QDA towards the common covariance of LDA. This improves the estimates the covariance matrix in situations where the number of predictors is larger than the number of samples in the training data leading to improvement in the model accuracy.

Conclusion:

Discriminant Analysis finds a set of prediction equations based on independent variables that are used to classify individuals into groups. There are two possible objectives in a discriminant analysis: finding a predictive equation for classifying new individuals or interpreting the predictive equation to better understand the relationships that may exist among the variables. During this Lab we examined the LDA and QDA models, their advantages and disadvantages , and their implementation under the Naive Bayes assumptions.