Machine Learning Exercises 11

This set of exercises are about Bayes classifier for generative models. LDA and QDA are both generative models that are based on Gaussian distributions for the features conditionally on the class. Of course, unless we know and use the exact distribution of the data, the classifier will be only an approximation to Bayes classifier.

Useful information

Estimated variance matrices in LDA and QDA with multiple features. In lectures and in ISLwR we have seen the (bias-corrected) mle for LDA and QDA with a single feature. Here are the corresponding results for multiple features.

The covariance matrices needed for QDA with multiple features are obtained for each class k as

$$\hat{\sigma}^2 = \frac{1}{n_k - 1} \sum_{i:y_i=k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

Here, x_i is the feature vector for observation i, and n_k is the number of observations in class k.

The shared covariance matrix for LDA is estimated as

$$\hat{\sigma}^2 = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i:y_i=k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

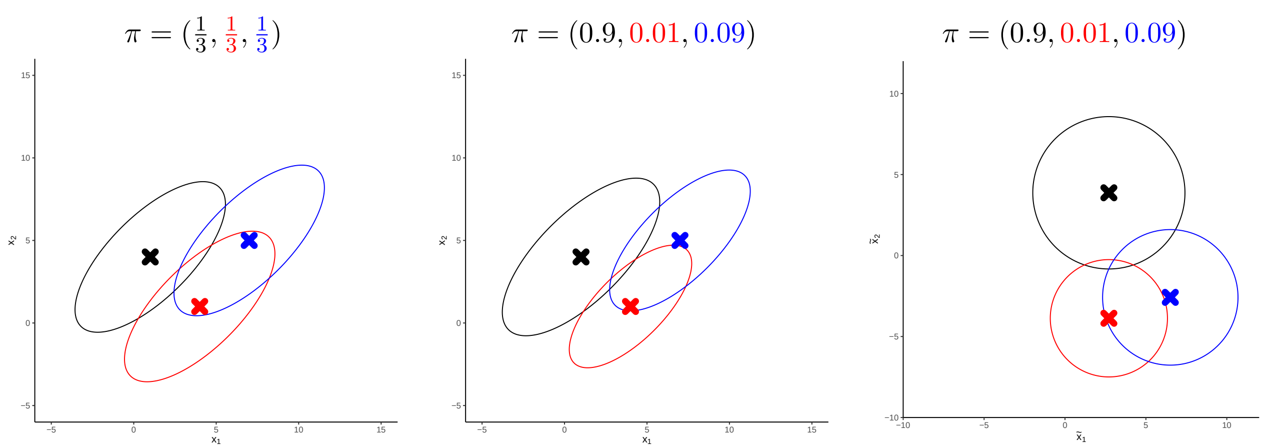
Multivariate normal distribution in Python. A multivariate random variable is implemented as `multivariate_normal` from `scipy.stats`. The pdf for the multivariate normal distribution can be found as method `pdf`, and random samples can be drawn using method `rvs`.

LDA decision regions

Let us warm up by sketching decision boundaries for LDA as we did it in lectures.

Exercise 1. Below you see three examples of contour curves for a specific value of the three discriminant functions in a three-class classification with LDA,

$$g_k(\boldsymbol{x}) = 2 \log \pi_k - (\boldsymbol{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k), \; k = 1, 2, 3.$$



For each example, sketch the decision boundaries and decision regions for the LDA classifier as follows:

a) First sketch the decision boundary between each pair of classes.

b) Then decide the winning colour in each of the six resulting regions

In the third example, how would boundaries look if the priors were instead equal?

LDA – shared covariance matrix for all classes

Exercise 2. Imagine that you have a binary classification problem, and that you know that data comes from a model, where class probabilities are $\mathrm{P}(Y = \text{black}) = 0.4$ and $\mathrm{P}(Y = \text{red}) = 0.6$ and class conditionals are multivariate normal with parameters

$$\boldsymbol{\mu}_{\text{black}} = (2, 1), \boldsymbol{\mu}_{\text{red}} = (4, 2), \text{ and a shared covariance matrix } \Sigma = \begin{bmatrix} 3 & -1 \\ -1 & 2 \end{bmatrix}$$

(a) Explain how to derive Bayes classifier for this model via two discriminant functions, where classification is to the class with highest discriminant function.

You do not have to derive the linear discriminant functions as we have done in lectures, but rather you can use directly the pdf for the multivariate normal distribution in the two discriminant functions.

(b) Classify a new data point with $X_1 = 3$ and $X_2 = 1$. Compute the posterior probability for each class.

(c) Create a plot that shows the decision regions for Bayes classifier.

(d) Explain how you would simulate from the model.

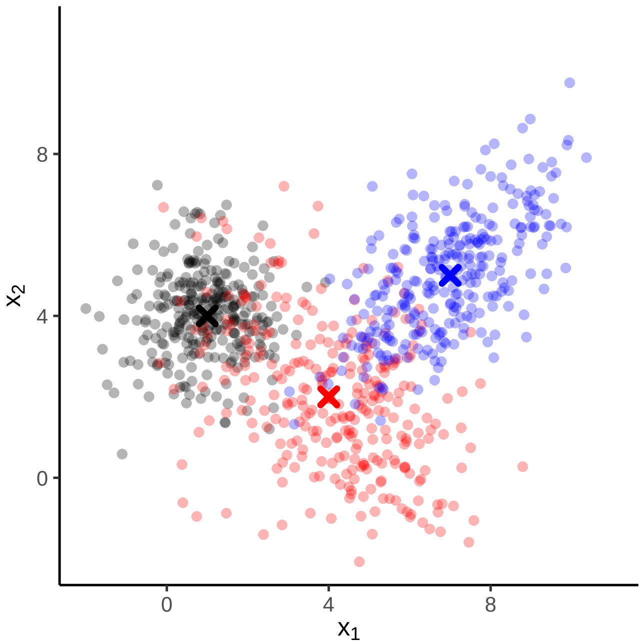
(e) Simulate a training set and use it to estimate the model parameters.

(f) Visualise the decision regions for the LDA classifier trained on the training set and compare to the decision regions for the Bayes classifier.

(g) Simulate a test set of 1000 observations and use it to compute both the Bayes error (the minimum possible error rate, as obtained with Bayes classifier) and the misclassification error for the LDA classifier trained on data.

QDA – one covariance matrix per class

Exercise 3. Revisit the two datasets from Exercise set 7, Ex1-training.csv and Ex1-test.csv. There are two features (x_1 and x_2) and three classes (Black = 1, Red = 2, Blue = 3). The training data is seen in Figure 1.



Class conditionals, $p(x|y)$:

$$p(x|\text{black}) = \text{MVN}\left(\begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}1 & 0.1\\0.1 & 1\end{bmatrix}\right)$$

$$p(x|\text{red}) = \text{MVN}\left(\begin{bmatrix}4\\2\end{bmatrix}, \begin{bmatrix}3 & -1.3\\-1.3 & 3\end{bmatrix}\right)$$

$$p(x|\text{blue}) = \text{MVN}\left(\begin{bmatrix}7\\5\end{bmatrix}, \begin{bmatrix}2 & 1.2\\1.2 & 2\end{bmatrix}\right)$$

Class probabilities:

$$\pi_{\text{black}} = 0.9$$
$$\pi_{\text{red}} = 0.01$$
$$\pi_{\text{blue}} = 0.09$$

Figure 1: Training data with two features and three classes (Black = 1, Red = 2, Blue = 3).

a) What is the expectation of feature X_2 when class is black?

b) What is the variance of feature X_1 when the class is red?

c) What is the covariance between features X_1 and X_2 when class is red? Their correlation?

d) Train and visualise the LDA classifier.

e) Compute the estimated posterior distribution of classes, $\widehat{\mathrm{P}}(Y = y|X = (3, 3.5))$, for a new observation $(x_1, x_2) = (3, 3.5)$.

f) Compute the test error. Is it better or worse than the KNN test error you got in Exercise set 7?

g) Looking at Figure 1, what can you say about the suitability of LDA? Discuss the relation to the test errors in f).

h) What would be an argument for using QDA instead of LDA?

i) Train the QDA classifier and visualise its decision regions.

j) Make two sets of contour plots: one for the three estimated Gaussian distributions as obtained by LDA, and one for those obtained by QDA. Comment again on which model seems more suitable for these data.

k) Compare the test error for QDA and LDA and comment.