An exploration of linear classification

By : Thibault Lahire (thibault.lahire@student.isae-supaero.fr)

The files trainA, trainB and trainC contain samples of data (x_n, y_n) where $x_n \in \mathbb{R}^2$ and $y_n \in \{0, 1\}$ (each line of each file contains the 2 components of x_n then y_n). Our goal is to implement linear classification methods and to test them on the three data sets.

1 Generative model (LDA)

Given the class variable, the data are assumed to be Gaussian with different means for different classes but with the same covariance matrix.

$$y \sim B(\pi),\ x|y = i \sim N(\mu_i, \Sigma).$$

a) We first derive the form of the maximum likelihood estimator for this model.

We consider for this exercise that we observed the sample : $S = \{(x_1, y_1), ..., (x_n, y_n)\}$.
We first focus on the sample $(y_1, ..., y_n)$ to find the MLE $\widehat{\pi}$ of π.
For $y \sim B(\pi)$, we have the probability distribution :

$$p(y|\pi) = \pi^y (1-\pi)^{1-y}$$

We derive the log-likelihood :

$$L_{(y_i)_i}(\pi) = \sum_{i=1}^{n} \log(\pi^{y_i}(1-\pi)^{1-y_i}) = \sum_{i=1}^{n} y_i \log(\pi) + (1-y_i)\log(1-\pi)$$

$$L_{(y_i)_i}(\pi) = (\log(\pi) - \log(1-\pi)) \sum_{i=1}^{n} y_i + n \log(1-\pi)$$

We maximize the log-likelihood to have the expression of the MLE :

$$0 = \frac{dL_{(y_i)_i}}{d\pi}(\pi) = \left(\frac{1}{\pi} + \frac{1}{1-\pi}\right) \sum_{i=1}^{n} y_i - \frac{n}{1-\pi}$$

Hence, we obtain :

$$n = \left(\frac{1-\pi}{\pi} + 1\right) \sum_{i=1}^{n} y_i \qquad \longrightarrow \pi = \frac{1}{n} \sum_{i=1}^{n} y_i$$

We derive the MLE of π :

$$\widehat{\pi} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Now, we want to find the MLEs $\widehat{\mu}_0$, $\widehat{\mu}_1$ and $\widehat{\Sigma}$.

We know the expression of the probability distribution :

$$p(x|y,\mu_0,\mu_1,\Sigma) = \sum_{k=0}^{1} \mathbf{1}\{y=k\} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)\right\}$$

We derive the log-likelihood :

$$\begin{aligned}
L_{(x_i)_i,(y_i)_i}(\mu_0,\mu_1,\Sigma) &= \sum_{i=1}^{n} \log\left(\sum_{k=0}^{1} \mathbf{1}\{y=k\} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)\right\}\right) \\
&= \sum_{\substack{i=1 \\ y_i=0}}^{n} \left\{-\frac{1}{2}\log((2\pi)^d|\Sigma|) - \frac{1}{2}(x_i-\mu_0)^\top\Sigma^{-1}(x_i-\mu_0)\right\} \\
&\quad + \sum_{\substack{i=1 \\ y_i=1}}^{n} \left\{-\frac{1}{2}\log((2\pi)^d|\Sigma|) - \frac{1}{2}(x_i-\mu_1)^\top\Sigma^{-1}(x_i-\mu_1)\right\} \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{\substack{i=1 \\ y_i=0}}^{n}(x_i-\mu_0)^\top\Sigma^{-1}(x_i-\mu_0) - \frac{1}{2}\sum_{\substack{i=1 \\ y_i=1}}^{n}(x_i-\mu_1)^\top\Sigma^{-1}(x_i-\mu_1) \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{\substack{i=1 \\ y_i=0}}^{n}\left\{x_i^\top\Sigma^{-1}x_i - 2\mu_0^\top\Sigma^{-1}x_i + \mu_0^\top\Sigma^{-1}\mu_0\right\} \\
&\quad - \frac{1}{2}\sum_{\substack{i=1 \\ y_i=1}}^{n}\left\{x_i^\top\Sigma^{-1}x_i - 2\mu_1^\top\Sigma^{-1}x_i + \mu_1^\top\Sigma^{-1}\mu_1\right\}
\end{aligned}$$

We now set gradients of L to zero to find the MLE estimators. If we note n_0 (respectively n_1) the number of examples in the first class (respectively second class), then we have :

$$0 = \nabla_{\mu_0} L_{(x_i)_i,(y_i)_i}(\mu_0,\mu_1,\Sigma) = \sum_{\substack{i=1 \\ y_i=0}}^{n}\left\{\Sigma^{-1}x_i - \Sigma^{-1}\mu_0\right\} \qquad \longrightarrow \mu_0 = \frac{1}{\sum_{\substack{i=1 \\ y_i=0}}^{n} 1}\sum_{\substack{i=1 \\ y_i=0}}^{n} x_i = \frac{1}{n_0}\sum_{\substack{i=1 \\ y_i=0}}^{n} x_i$$

We do the same for μ_1, and we finally obtain :

$$\widehat{\mu}_0 = \frac{1}{n_0}\sum_{\substack{i=1 \\ y_i=0}}^{n} x_i \qquad \textit{and} \qquad \widehat{\mu}_1 = \frac{1}{n_1}\sum_{\substack{i=1 \\ y_i=1}}^{n} x_i$$

We now want to find the MLE of Σ. We need to set the gradient of $L_{(x_i)_i,(y_i)_i}$ with respect to Σ to zero. Be careful, Σ is a matrix... First, we re-write the log-likelihood :

$$L_{(x_i)_i,(y_i)_i}(\mu_0,\mu_1,\Sigma) = -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}(x_i-\mu_0)^\top\Sigma^{-1}(x_i-\mu_0) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}(x_i-\mu_1)^\top\Sigma^{-1}(x_i-\mu_1)$$

$$= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}Tr((x_i-\mu_0)^\top\Sigma^{-1}(x_i-\mu_0)) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}Tr((x_i-\mu_1)^\top\Sigma^{-1}(x_i-\mu_1))$$

$$= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}Tr(\Sigma^{-1}(x_i-\mu_0)(x_i-\mu_0)^\top) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}Tr(\Sigma^{-1}(x_i-\mu_1)(x_i-\mu_1)^\top)$$

$$= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}Tr\left(\Sigma^{-1}\widetilde{\Sigma}_0^{(i)}\right) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}Tr\left(\Sigma^{-1}\widetilde{\Sigma}_1^{(i)}\right)$$

$$= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}h(\lambda) + f(\lambda)$$

where we introduced $\lambda = \Sigma^{-1}$. For $A \in S_n^{++}(\mathbb{R})$, we have defined $h(A) = \log\det(A)$ and :

$$f(\lambda) = -\frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}Tr\left(\lambda^\top\widetilde{\Sigma}_0^{(i)}\right) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}Tr\left(\lambda^\top\widetilde{\Sigma}_1^{(i)}\right)$$

Given the Frobenius inner product $\langle\cdot,\cdot\rangle_F$ between matrices, with $\langle A,B\rangle_F = Tr(A^\top B)$, we recall that for a differentiable real valued matrix function $g : \mathbb{R}^{p\times d} \longrightarrow \mathbb{R}$, the gradient of g at A is defined as the matrix $\nabla g(A)$ such that for all $H \in \mathbb{R}^{p\times d}$, we have :

$$g(A+H) = g(A) + \langle\nabla g(A), H\rangle_F + o(||H||_F)$$

A work on functions h and f (that has been done during "les cours de pré-rentrée du MVA") gives us that :

$$\nabla_\lambda h(\lambda) = \lambda^{-1} \qquad and \qquad \nabla_\lambda f(\lambda) = -\frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}\widetilde{\Sigma}_0^{(i)} - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}\widetilde{\Sigma}_1^{(i)}$$

Therefore :

$$\nabla_\lambda L_{(x_i)_i,(y_i)_i}(\mu_0,\mu_1,\lambda) = \frac{n}{2}\lambda^{-1} - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}\widetilde{\Sigma}_0^{(i)} - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}\widetilde{\Sigma}_1^{(i)}$$

Setting this gradient to zero gives us the MLE of Σ :

$$\widehat{\Sigma} = \frac{1}{n}\left(\sum_{\substack{i=1\\y_i=0}}^{n}\widehat{\widetilde{\Sigma}_0^{(i)}} + \sum_{\substack{i=1\\y_i=1}}^{n}\widehat{\widetilde{\Sigma}_1^{(i)}}\right)$$

where $\widehat{\widetilde{\Sigma}_0^{(i)}}$ and $\widehat{\widetilde{\Sigma}_1^{(i)}}$ are estimators built from the previous estimators $\widehat{\mu}_0$ and $\widehat{\mu}_1$:

$$\widehat{\widetilde{\Sigma}_0^{(i)}} = (x_i-\widehat{\mu}_0)(x_i-\widehat{\mu}_0)^\top \qquad and \qquad \widehat{\widetilde{\Sigma}_1^{(i)}} = (x_i-\widehat{\mu}_1)(x_i-\widehat{\mu}_1)^\top$$

Remark : Note that, in the computation of $\widehat{\Sigma}$, the $\widehat{\tilde{\Sigma}_1^{(i)}}$ are in proportion $\widehat{\pi}$ and the $\widehat{\tilde{\Sigma}_0^{(i)}}$ are in proportion $1-\widehat{\pi}$. This will be used in the code.

b) We now study the form of the conditional distribution $p(y=1|x)$ and compare with logistic regression.

We recall that :

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

and we know that

$$p(x|y=k) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)\right)$$

Hence,

$$\begin{aligned}
p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1)+p(x|y=0)p(y=0)} \\
&= \frac{1}{1+\frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}\exp\left\{-\frac{1}{2}(x-\mu_0)^\top\Sigma^{-1}(x-\mu_0)+\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1)\right\}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}\exp\left\{-x^\top\Sigma^{-1}\mu_0-\frac{1}{2}\mu_0^\top\Sigma^{-1}\mu_0+x^\top\Sigma^{-1}\mu_1+\frac{1}{2}\mu_1^\top\Sigma^{-1}\mu_1\right\}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}\exp\left\{-(\mu_0-\mu_1)^\top\Sigma^{-1}x-\frac{1}{2}(\mu_0^\top\Sigma^{-1}\mu_0-\mu_1^\top\Sigma^{-1}\mu_1)\right\}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}\exp\left\{-(\Sigma^{-1}(\mu_0-\mu_1))^\top x-\frac{1}{2}(\mu_0^\top\Sigma^{-1}\mu_0-\mu_1^\top\Sigma^{-1}\mu_1)\right\}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}\exp\left\{-(w^\top x+b)\right\}}
\end{aligned}$$

with $w=\Sigma^{-1}(\mu_0-\mu_1)$ and $b=\frac{1}{2}(\mu_0^\top\Sigma^{-1}\mu_0-\mu_1^\top\Sigma^{-1}\mu_1)$

The key point of logistic regression is the following equation : $p(y=1|x)=\sigma(w^\top x)$, where $\sigma(z)=1/(1+\exp(-z))$. If we use the trick to include the bias b in the scalar product (explained in class), we have an equality between the logistic regression and the LDA when $\pi=0.5$.

c) We now represent graphically the data as a point cloud in $\mathbb{R}^2$ and the line defined by the equation $p(y=1|x)=0.5$ thanks to our implementation.

For the training dataset A, we have :

$$\widehat{\pi}_A = 0.48 \quad ; \quad \widehat{\mu}_{0_A} = [10.73248858, \quad 10.93983367]^\top \quad ; \quad \widehat{\mu}_{1_A} = [11.03264581, \quad 5.99294053]^\top$$

$$\widehat{\Sigma}_A = \begin{pmatrix} 0.58821974 & 0.13912842 \\ 0.13912842 & 0.81959919 \end{pmatrix}$$

For the training dataset B, we have :

$$\widehat{\pi}_B = 0.55 \quad ; \quad \widehat{\mu}_{0_B} = [10.58256756, \quad 11.17169818]^\top \quad ; \quad \widehat{\mu}_{1_B} = [11.24757662, \quad 6.095283]^\top$$

$$\widehat{\Sigma}_B = \begin{pmatrix} 1.64391088 & 0.70139847 \\ 0.70139847 & 2.0605845 \end{pmatrix}$$

For the training dataset C, we have :

$$\widehat{\pi}_C = 0.417 \quad ; \quad \widehat{\mu}_{0_C} = [10.6192273, \quad 10.83868653]^\top \quad ; \quad \widehat{\mu}_{1_C} = [11.18463199, \quad 6.04249315]^\top$$

$$\widehat{\Sigma}_C = \begin{pmatrix} 1.27823018 & -0.06243809 \\ -0.06243809 & 1.66584186 \end{pmatrix}$$

Here are the numerical values of the learnt parameters :

Dataset	Vector w	Bias b
A	$[2.01894473, \quad -6.37846623]^\top$	32.0312625
B	$[1.7029769, \quad -3.0432533]^\top$	7.68578304
C	$[0.30224914, \quad -2.86781214]^\top$	20.9109271

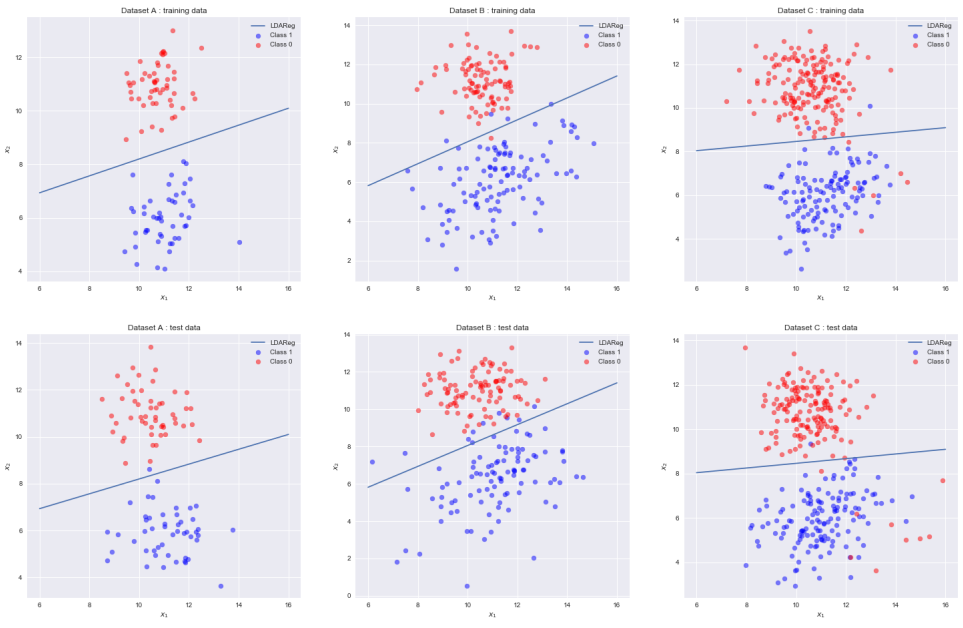
If we normalize along the vertical axis so that the cartesian equation defining the straight line is easier to write, we obtain :

Dataset	Vector w	Bias b
A	$[-0.31652511, \quad 1.]^\top$	−5.02178131
B	$[-0.55959092, \quad 1.]^\top$	−2.52551539
C	$[-0.10539363, \quad 1.]^\top$	−7.29159586

We have to remark that :

$$p(y=1|x) = 0.5 \iff \frac{1}{\frac{1-\pi}{\pi}\exp(-(w^\top x + b))} = \frac{1}{2} \iff w^\top x + b = -\log\frac{\pi}{1-\pi}$$

Since we now have the equation of the separating hyperplane, we can plot it :



2 Logistic regression

We implement logistic regression for an affine function $f(x) = w^\top x + b$.

First, we explain briefly what logistic regression is about. We will consider that the bias b has already been included in the scalar product $w^\top x$ thanks to the trick explained in class. The key assumption of logistic regression for binary classification is :

$$\log \frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=0|X=x)} = w^\top x$$

which implies that $p(y=1|x) = \sigma(w^\top x)$, where $\sigma(z) = 1/(1+\exp(-z))$ as written above.
Let $\eta := \sigma(w^\top x)$. By assumption, $Y|X=x \sim B(\eta)$.
We derive the probability distribution :

$$p(y|x,\eta) = \eta^y(1-\eta)^{1-y} = p(y|x,w) = \sigma(w^\top x)^y \sigma(-w^\top x)^{1-y}$$

We derive the log-likelihood for the sample S :

$$L(w) = \sum_{i=1}^{n} y_i \log \sigma(w^\top x_i) + (1-y_i)\log \sigma(-w^\top x) = \sum_{i=1}^{n} y_i \log \eta_i + (1-y_i)\log(1-\eta_i)$$

$$L(w) = \sum_{i=1}^{n} y_i \log \frac{\eta_i}{1-\eta_i} + \log(1-\eta_i) = \sum_{i=1}^{n} y_i w^\top x_i + \log \sigma(-w^\top x_i)$$

$$\nabla_w L(w) = X^\top(\mathbf{y} - \zeta) \qquad and \qquad \nabla_w^2 L(w) = -X^\top Diag(\eta_i(1-\eta_i))X$$

where $\eta_i = \sigma(w^\top x_i)$, and $\zeta = [\eta_1, ..., \eta_n]^\top$.
Note that this function is differentiable and concave. However, setting the gradient of L to zero will not give us a simple analytical expression. We have to compute an optimization algorithm for this function. For such a problem, Newton will perfectly work and ensures a quadratic convergence. We recall Newton's algorithm :

Initialization : w_0, ϵ
Repeat
- Compute Newton step : $d = -\nabla_w^2 L(w)^{-1} \nabla_w L(w)$
- Compute decrement : $\lambda^2 = \nabla_w L(w)^\top \nabla_w^2 L(w)^{-1} \nabla_w L(w)$
- Stopping criterion : **quit** if $\lambda^2/2 < \epsilon$
- Line search : Choose step size t by backtracking line-search (not mandatory).
- Update : $w_{t+1} = w_t + td$

Numerical values of the learnt parameters :

Dataset	Vector w	Bias b
A	$[0.22329753, \quad -0.70546545]^\top$	3.53383096
B	$[0.3303269, \quad -0.59030068]^\top$	1.52999937
C	$[0.06701844, \quad -0.63588697]^\top$	4.56060824

If we normalize along the vertical axis so that the cartesian equation defining the straight line is easier to write, we obtain :

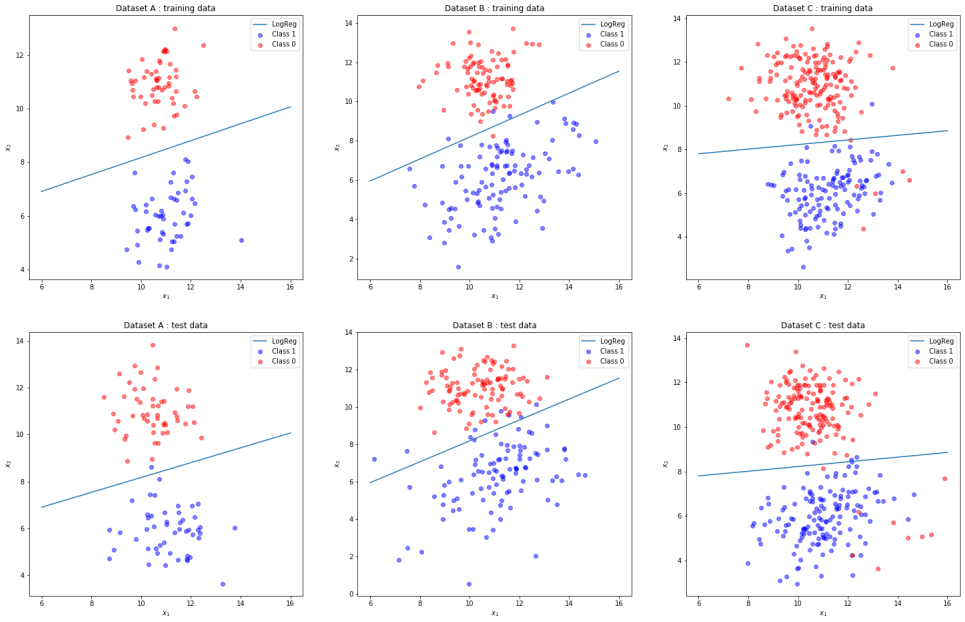
Dataset	Vector w	Bias b
A	$[-0.31652511, \quad 1.]^\top$	-5.00921901
B	$[-0.55959092, \quad 1.]^\top$	-2.59189836
C	$[-0.10539363, \quad 1.]^\top$	-7.1720423

It is interesting to compare the normalized weights we obtained for the logistic regression with the one we obtained with the LDA method. We recall that, when $\pi = 0.5$, the logistic regression and the LDA method gives the same decision frontier.

For the three datasets we have, π is never equal to 0.5, but for dataset A, $\widehat{\pi}_A = 0.48$, for dataset B, $\widehat{\pi}_B = 0.55$ and for dataset C, $\widehat{\pi}_C = 0.417$. In other words, the results of logistic regression and the LDA method are supposed to be closer for dataset A than for dataset B and C. The normalized weights obtained illustrates this theoretical point. Indeed, w and b for dataset A resulting from logistic regression and LDA are closer than w and b for dataset B or C resulting from logistic regression and LDA.

We now represent graphically the data as a cloud point in $\mathbb{R}^2$ as well as the line defined by the equation $p(y = 1|x) = 0.5$.

Since $p(y = 1|x) = 0.5$, we have $\log \frac{0.5}{1-0.5} = w^\top x$, in other words, we have to represent in the figures below the hyperplane $w^\top x = 0$ (or equivalently $w^\top x + b = 0$).



3 Linear regression

We consider class y as a real valued variable taking the values 0 and 1 only. We implement linear regression (for an affine function $f(x) = w^\top x + b$) by solving the

normal equations.

Solving the linear regression problem means solving the normal equation :

$$X^\top X w = X^\top y$$

where we consider that the bias b has already been included in the matrix X and the vector w thanks to the trick explained in class. Hopefully, for the given datasets, the square matrix $X^\top X$ is non-singular. Hence the solution of the linear regression problem is :

$$w = (X^\top X)^{-1} X^\top y$$

Numerical values of the learnt parameters :

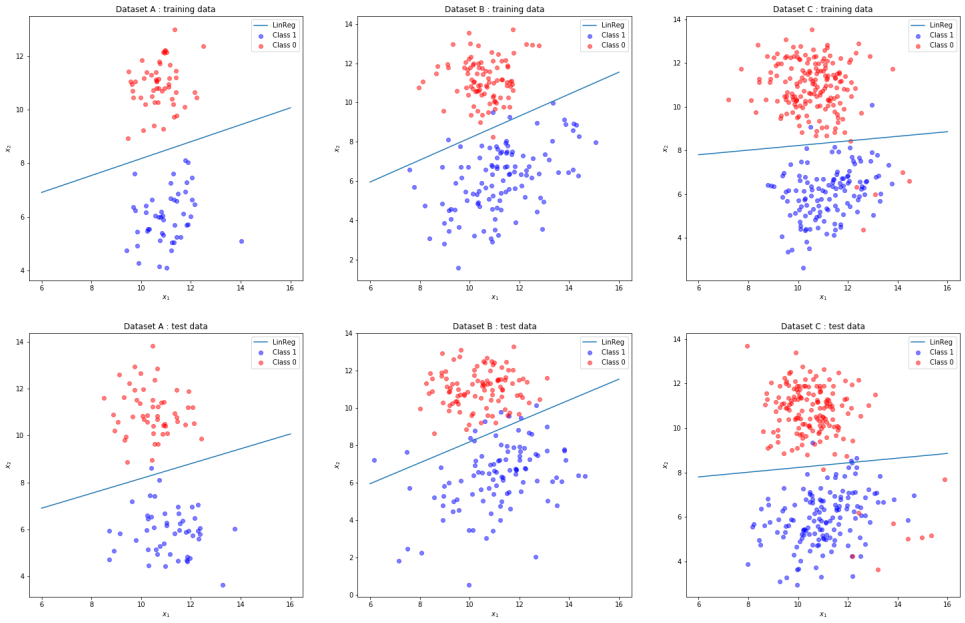
Dataset	Vector w	Bias b
A	$[0.05582438, \quad -0.17636636]^\top$	1.38345774
B	$[0.08258172, \quad -0.14757517]^\top$	0.88249984
C	$[0.01675461, \quad -0.15897174]^\top$	1.64015206

If we normalize along the vertical axis so that the cartesian equation defining the straight line is easier to write, we obtain :

Dataset	Vector w	Bias b
A	$[-0.31652511, \quad 1.]^\top$	-7.84422673
B	$[-0.55959092, \quad 1.]^\top$	-5.9800022
C	$[-0.10539363, \quad 1.]^\top$	-10.31725536

We now represent graphically the data as a point cloud in $\mathbb{R}^2$ as well as the line defined by the equation $p(y = 1|x) = 0.5$.

Once w (and b) has been found, it is important to see the function f defined by $f(x) = w^\top x + b$ as a function whose thresholding gives us a decision. Indeed, $f(x) = 0.5$ i.e. $w^\top x + b = 0.5$ has to be seen as a separating hyperplane. If a new covariate x_{new} is such that $f(x_{new}) > 0.5$, then x_{new} belongs to the first class, otherwise it belongs to the second class. Hence, in the figures below, the line is the separating hyperplane defined by the equation $w^\top x + b = 0.5$.



4 Application

Data in the files testA, testB and testC are respectively drawn from the same distribution as the data in the files trainA, trainB and trainC. We test the different models learnt from the corresponding training data on these test data.

We compute for each model the misclassification error (i.e. the fraction of the data misclassified) on the training data and compute it as well on the test data.

Set →	Train A	Test A	Train B	Test B	Train C	Test C
LDA	0.0	0.01	0.02	0.045	0.0267	0.04
Logistic	0.0	0.01	0.02	0.045	0.0267	0.04
Linear	0.0	0.01	0.02	0.045	0.0267	0.04

The first remarkable thing when seeing the result is to note that the misclassification error is larger on the test sets, and smaller on the training sets. This is explained by the fact that the estimators / methods have been built on the training sets only. When they see the test sets, they apply the same decision rule as for the training sets, without adaptation. Hence it is normal to see a larger error on the test sets.

On these datasets, the LDA method, the logistic regression and the linear regression provides similar results. This can be explained by the fact that the datasets we are using are very simple. In a more general setting, this is not supposed to happen : these 3 methods are not equivalent.

Note : For dataset A, the points are clearly linearly separable during the training, so all the methods work perfectly.

5 QDA model

We finally relax the assumption that the covariance matrices for the two classes are the same. So, given the class label the data are assumed to be Gaussian with means and covariance matrices which are a priori different.

$$y \sim B(\pi),\ x|y=i \sim N(\mu_i, \Sigma_i).$$

We implement the maximum likelihood estimator and apply it to the data.

Now, we want to find the MLEs $\widehat{\mu}_0$, $\widehat{\mu}_1$, $\widehat{\Sigma}_0$, and $\widehat{\Sigma}_1$.

We know the expression of the probability distribution :

$$p(x|y,\mu_0,\mu_1,\Sigma_0,\Sigma_1) = \sum_{k=0}^{1} \mathbf{1}\{y=k\} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\right\}$$

We derive the log-likelihood :

$$\begin{aligned}
L_{(x_i)_i,(y_i)_i}(\mu_0,\mu_1,\Sigma_0,\Sigma_1) &= \sum_{i=1}^{n} \log\left(\sum_{k=0}^{1} \mathbf{1}\{y=k\} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\right\}\right) \\
&= \sum_{\substack{i=1\\y_i=0}}^{n} \left\{-\frac{1}{2}\log((2\pi)^d|\Sigma_0|) - \frac{1}{2}(x_i-\mu_0)^\top\Sigma_0^{-1}(x_i-\mu_0)\right\} \\
&\quad + \sum_{\substack{i=1\\y_i=1}}^{n} \left\{-\frac{1}{2}\log((2\pi)^d|\Sigma_1|) - \frac{1}{2}(x_i-\mu_1)^\top\Sigma_1^{-1}(x_i-\mu_1)\right\} \\
&= -\frac{n_0 d}{2}\log(2\pi) - \frac{n_0}{2}\log(|\Sigma_0|) - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n}(x_i-\mu_0)^\top\Sigma_0^{-1}(x_i-\mu_0) \\
&\quad - \frac{n_1 d}{2}\log(2\pi) - \frac{n_1}{2}\log(|\Sigma_1|) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n}(x_i-\mu_1)^\top\Sigma_1^{-1}(x_i-\mu_1) \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n_0}{2}\log(|\Sigma_0|) - \frac{n_1}{2}\log(|\Sigma_1|) \\
&\quad - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n} Tr\left(\Sigma_0^{-1}(x_i-\mu_0)(x_i-\mu_0)^\top\right) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n} Tr\left(\Sigma_1^{-1}(x_i-\mu_1)(x_i-\mu_1)^\top\right) \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n_0}{2}\log(|\Sigma_0|) - \frac{n_1}{2}\log(|\Sigma_1|) - \frac{1}{2}\sum_{\substack{i=1\\y_i=0}}^{n} Tr\left(\Sigma_0^{-1}\widetilde{\Sigma}_0^{(i)}\right) - \frac{1}{2}\sum_{\substack{i=1\\y_i=1}}^{n} Tr\left(\Sigma_1^{-1}\widetilde{\Sigma}_1^{(i)}\right)
\end{aligned}$$

We recognize a form very close to the one we studied a few lines above. Therefore, we derive :

$$\widehat{\mu}_0 = \frac{1}{n_0}\sum_{\substack{i=1\\y_i=0}}^{n} x_i \qquad and \qquad \widehat{\mu}_1 = \frac{1}{n_1}\sum_{\substack{i=1\\y_i=1}}^{n} x_i$$

Moreover, if we introduce $\lambda_0 = \Sigma_0^{-1}$, $\lambda_1 = \Sigma_1^{-1}$, and if we write the log-likelihood $L_{(x_i)_i,(y_i)_i}(\mu_0,\mu_1,\lambda_0,\lambda_1)$, setting the gradients to zero gives us :

$$0 = \nabla_{\lambda_1} L = \frac{n_1}{2}\lambda^{-1} - \frac{1}{2}\sum_{\substack{i=1 \\ y_i=0}}^{n} \overset{\sim}{\Sigma}_1^{(i)} \qquad \longrightarrow \Sigma_1 = \frac{1}{n_1}\sum_{\substack{i=1 \\ y_i=0}}^{n} \overset{\sim}{\Sigma}_1^{(i)}$$

We set to zero the gradient of the log-likelihood with respect to λ_0, and we obtain a similar result. Finally, we conclude :

$$\widehat{\Sigma}_0 = \frac{1}{n_0}\sum_{\substack{i=1 \\ y_i=0}}^{n} \widehat{\overset{\sim}{\Sigma}_0^{(i)}} \qquad with \qquad \widehat{\overset{\sim}{\Sigma}_0^{(i)}} = (x_i - \widehat{\mu}_0)(x_i - \widehat{\mu}_0)^\top$$

$$\widehat{\Sigma}_1 = \frac{1}{n_1}\sum_{\substack{i=1 \\ y_i=1}}^{n} \widehat{\overset{\sim}{\Sigma}_1^{(i)}} \qquad with \qquad \widehat{\overset{\sim}{\Sigma}_1^{(i)}} = (x_i - \widehat{\mu}_1)(x_i - \widehat{\mu}_1)\top$$

Here is what we obtained :

For the training dataset A, we have :

$$\widehat{\pi}_A = 0.48 \quad ; \qquad \widehat{\mu}_{0_A} = [10.73248858, \quad 10.93983367]^\top \quad ; \qquad \widehat{\mu}_{1_A} = [11.03264581, \quad 5.99294053]^\top$$

$$\widehat{\Sigma}_{0_A} = \begin{pmatrix} 0.46464757 & 0.0989291 \\ 0.0989291 & 0.71324415 \end{pmatrix}$$

$$\widehat{\Sigma}_{1_A} = \begin{pmatrix} 0.7220896 & 0.18267769 \\ 0.18267769 & 0.93481714 \end{pmatrix}$$

For the training dataset B, we have :

$$\widehat{\pi}_B = 0.55 \quad ; \qquad \widehat{\mu}_{0_B} = [10.58256756, \quad 11.17169818]^\top \quad ; \qquad \widehat{\mu}_{1_B} = [11.24757662, \quad 6.095283]^\top$$

$$\widehat{\Sigma}_{0_B} = \begin{pmatrix} 0.76164392 & 0.05352417 \\ 0.05352417 & 1.10741986 \end{pmatrix}$$

$$\widehat{\Sigma}_{1_B} = \begin{pmatrix} 2.36576566 & 1.23147744 \\ 1.23147744 & 2.84044647 \end{pmatrix}$$

For the training dataset C, we have :

$$\widehat{\pi}_C = 0.416 \quad ; \qquad \widehat{\mu}_{0_C} = [10.6192273, \quad 10.83868653]^\top \quad ; \qquad \widehat{\mu}_{1_C} = [11.18463199, \quad 6.04249315]^\top$$

$$\widehat{\Sigma}_{0_C} = \begin{pmatrix} 1.28576019 & -0.43356756 \\ -0.43356756 & 1.82602425 \end{pmatrix}$$

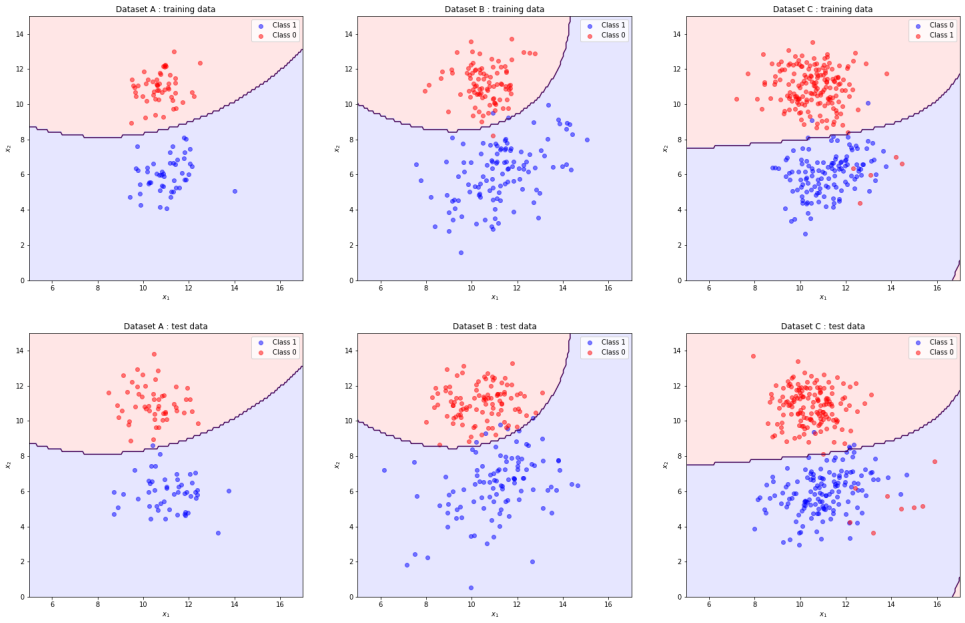
$$\widehat{\Sigma}_{1_C} = \begin{pmatrix} 1.26768817 & 0.45714317 \\ 0.45714317 & 1.44158652 \end{pmatrix}$$

We represent graphically the data as well as the conic defined by $p(y = 1|x) = 0.5$.

Recall that :

$$
\begin{aligned}
p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1)+p(x|y=0)p(y=0)} \\
&= \frac{1}{1+\frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}\frac{|\Sigma_1|^{\frac{1}{2}}}{|\Sigma_0|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu_0)^\top\Sigma_0^{-1}(x-\mu_0)+\frac{1}{2}(x-\mu_1)^\top\Sigma_1^{-1}(x-\mu_1)\right\}}
\end{aligned}
$$

Thanks to the estimators $\widehat{\pi}$, $\widehat{\mu}_0$, $\widehat{\mu}_1$, $\widehat{\Sigma}_0$ and $\widehat{\Sigma}_1$ that we have built, we can compute $p(y=1|x)$ in each point x of the considered domain. The equalization of this expression with 0.5 gives us a frontier. We have :



Set →	Train A	Test A	Train B	Test B	Train C	Test C
LDA	0.0	0.01	0.02	0.045	0.0267	0.04
Logistic	0.0	0.01	0.02	0.045	0.0267	0.04
Linear	0.0	0.01	0.02	0.045	0.0267	0.04
QDA	0.0	0.01	0.015	0.025	0.0267	0.0433

On dataset A, since the points are clearly linearly separable, all the methods work. We always have the same results.

However, on dataset B, the QDA model performs best. Even if it has a misclassification error comparable to the other methods on the training set, it clearly generalizes better than the others when working on the test set.

On dataset C, the QDA method has a slight tendency to overfit, compared to the other methods.

The excellent performance of QDA on the sets B can be explained by the geometry. Indeed, a curved line classifies much better than a straight line.

This is also the case for the training of dataset C. However, the test phase does not provide better results than a straight line because some red points are found "behind" the blue points.

We note that the form of the red conic is such that the bottom right corner is considered as red. That's why we think that if we had more data during the training phase, especially more data "behind" the blue points, the red area would have fit better...
