Probabilistic Graphical Models **MVA Fall 2019**

Homework 1 — November 22

Instructor: Pierre Latouche *Student: Antoine Moulin, Marie Heurtevent*

1 Learning in discrete graphical models

Consider the following model: z and x are discrete variables taking respectively M and K different values with $p(z = m) = \pi_m$ and $p(x = k|z = m) = \theta_{mk}$.
Compute the maximum likelihood estimator for π and θ based on an i.i.d. sample of observations. Please provide your derivations and not just the final answer.

Solution.

Let $\{(x_i, z_i)\}_{i\in[\![1,n]\!]}$ be n i.i.d. samples. The likelihood ℓ of $\pi \in \mathbb{R}^M$ and $\theta \in \mathbb{R}^{M\times K}$ is given by:

$$\begin{aligned}\ell(\pi,\theta) &= \ell_{(x_1,z_1),\ldots,(x_n,z_n)}(\pi,\theta)\\ &= p\left((x_1,z_1),\ldots,(x_n,z_n)|\pi,\theta\right)\end{aligned}$$

By independence of the samples and by definition of the variables,

$$\begin{aligned}\ell(\pi,\theta) &= \prod_{i=1}^{n} p(x_i, z_i|\pi,\theta) && \text{(independence of the } (x_i,z_i)\text{)}\\ &= \prod_{i=1}^{n} p(x_i|z_i,\pi,\theta)\,p(z_i|\pi,\theta) && \text{(definition of conditional distribution)}\\ &= \prod_{i=1}^{n}\left(\prod_{m=1}^{M}\prod_{k=1}^{K}\theta_{mk}^{\mathbb{1}\{z_i=m,x_i=k\}}\right)\left(\prod_{m=1}^{M}\pi_m^{\mathbb{1}\{z_i=m\}}\right) && \text{(definition of the variables } x_i, z_i\text{)}\end{aligned}$$

Hence, the log-likelihood L is:

$$L(\pi,\theta) = \sum_{i=1}^{n}\sum_{m=1}^{M}\left(\log(\pi_m)\mathbb{1}_{\{z_i=m\}} + \sum_{k=1}^{K}\log(\theta_{mk})\mathbb{1}_{\{z_i=m,x_i=k\}}\right)$$

In order to get the maximum likelihood estimator for π and θ, one could maximize the log-likelihood. Hence, one has to solve the optimization problem:

$$\begin{aligned}&\min_{\pi,\theta} -L(\pi,\theta)\\ \text{s.t.}\quad &\sum_{1\le m\le M}\pi_m = 1\\ &\sum_{1\le k\le K}\theta_{mk} = 1 \text{ for } m = 1,\ldots,M\end{aligned}$$

The negative log-likelihood is strictly convex with respect to π and θ. Besides, the set of feasible points is bounded and non-empty. Hence, there exists a unique minimizer for this problem. By Slater's constraint qualification, strong duality holds. For $\lambda \in \mathbb{R}, \mu \in \mathbb{R}^M$, the Lagrangian $\mathcal{L}$ of this problem is given by the expression:

$$\mathcal{L}(\pi, \theta, \lambda, \mu) = -L(\pi, \theta) + \lambda \left(\sum_{m=1}^{M} \pi_m - 1 \right) + \sum_{m=1}^{M} \mu_m \left(\sum_{k=1}^{K} \theta_{mk} - 1 \right)$$

Let $m \in [\![1, M]\!], k \in [\![1, K]\!]$. One has:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \pi_m} = 0 \text{ iff } & -\frac{1}{\pi_m} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m\}} + \lambda = 0 \\ \text{iff } & \pi_m = \frac{1}{\lambda} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m\}} \end{aligned}$$

and,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_{mk}} = 0 \text{ iff } & -\frac{1}{\theta_{mk}} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m, x_i = k\}} + \mu_m = 0 \\ \text{iff } & \mu_m = \frac{1}{\theta_{mk}} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m, x_i = k\}} \end{aligned}$$

Besides, the constraints (π and the columns of θ sum to one) give us:

$$\begin{aligned} \sum_{m=1}^{M} \pi_m = 1 \text{ iff } & \frac{1}{\lambda} \sum_{m=1}^{M} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m\}} = 1 \\ \text{iff } & \lambda = \sum_{i=1}^{n} \sum_{m=1}^{M} \mathbb{1}_{\{z_i = m\}} \\ \text{iff } & \lambda = n \end{aligned}$$

and, for $m = 1, \ldots, M$,

$$\begin{aligned} \sum_{k=1}^{K} \theta_{mk} = 1 \text{ iff } & \frac{1}{\mu_m} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m, x_i = k\}} = 1 \\ \text{iff } & \mu_m = \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m\}} \sum_{k=1}^{K} \mathbb{1}_{\{x_i = k\}} \\ \text{iff } & \mu_m = \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m\}} \end{aligned}$$

Hence, the maximum likelihood estimators are

$$\boxed{\begin{aligned} \forall m \in [\![1, M]\!], \widehat{\pi}_m &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m\}} \\ \forall m \in [\![1, M]\!], \forall k \in [\![1, K]\!], \widehat{\theta}_{mk} &= \frac{1}{\sum_{i=1}^{n} \mathbb{1}_{\{z_i = m\}}} \sum_{i=1}^{n} \mathbb{1}_{\{z_i = m, x_i = k\}} \end{aligned}}$$

2 Linear classification

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). The goal of this exercise is to implement linear classification methods and to test them on the three data sets. The code should be written in R or Python. The source code should be handed in along with results. However, all the requested figures should be printed on paper or part of a pdf file which is turned in, with clear titles that indicate what the figures represent. Therefore, we recommend to write your code and report thanks to a Markdown file (in R or Python). The discussions may of course be handwritten.

2.1 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.

$$y \sim \text{Bernoulli}(\pi), x|y = i \sim \text{Normal}(\mu_i, \Sigma)$$

a Derive the form of the maximum likelihood estimator for this model.

b What is the form of the conditional distribution $p(y = 1|x)$? Compare with the form of logistic regression.

c Implement the MLE for this model and apply it to the data. Represent graphically the data as a point cloud in $\mathbb{R}^2$ and the line defined by the equation

$$p(y = 1|x) = 0.5$$

Solution.

a Let $\{(x_k, y_k)\}_{k \in [\![1,n]\!]}$ be n i.i.d. samples. We note that Σ has to be positive definite, otherwise the density would not be defined. As in the first part, using the independence of the samples and the definition of the variables, one can show that the likelihood of $\theta = (\pi, \mu_0, \mu_1, \Sigma)$ is given by:

$$\begin{aligned}
\ell(\theta) &= p((x_1, y_1), \dots, (x_n, y_n)|\theta) \\
&= \prod_{k=1}^{n} p(x_k, y_k|\theta) \\
&= \prod_{k=1}^{n} p(x_k|y_k, \theta) p(y_k|\theta) \\
&= \prod_{k=1}^{n} \left(\frac{1}{(2\Pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (x_k - \mu_{y_k})^T \Sigma^{-1} (x_k - \mu_{y_k}) \right] \right) \left(\pi^{y_k} (1-\pi)^{1-y_k} \right)
\end{aligned}$$

where $d = 2$ is the dimension of the x_k's and Π the universal constant (not to be confused with the parameter π). Hence, the log-likelihood L is given by:

$$L(\theta) = -\frac{nd}{2} \log 2\Pi - \frac{n}{2} \log |\Sigma| + \sum_{k=1}^{n} \left\{ -\frac{1}{2} (x_k - \mu_{y_k})^T \Sigma^{-1} (x_k - \mu_{y_k}) + y_k \log(\pi) + (1 - y_k) \log(1 - \pi) \right\}$$

L is differentiable. Let's compute the gradient with respect to π, μ_0, μ_1 and Σ. We have:

$$\begin{aligned}\frac{\partial L}{\partial \pi} &= \frac{1}{\pi}\sum_{k=1}^{n} y_k - \frac{1}{1-\pi}\sum_{k=1}^{n} 1-y_k \\ &= \frac{1}{\pi}\sum_{k=1}^{n} y_k - \frac{1}{1-\pi}\left(n - \sum_{k=1}^{n} y_k\right)\end{aligned}$$

Hence,

$$\begin{aligned}\frac{\partial L}{\partial \pi} = 0 \text{ iff } \frac{1}{\pi}\sum_{k=1}^{n} y_k &= \frac{1}{1-\pi}\left(n - \sum_{k=1}^{n} y_k\right) \\ \text{iff } \pi &= \frac{1}{n}\sum_{k=1}^{n} y_k \\ \text{iff } \pi &= \frac{n_1}{n} \triangleq \widehat{\pi}\end{aligned}$$

where $n_1 = \sum_{k=1}^{n} y_k = \sum_{k=1}^{n} \mathbb{1}_{\{y_k=1\}}$ (we also define $n_0 = \sum_{k=1}^{n} \mathbb{1}_{\{y_k=0\}} = n - n_1$). The MLE estimator for π is the proportion of examples from the class 1 in the dataset. Now, the gradient with respect to μ_i, where $i \in \{0,1\}$, is

$$\frac{\partial L}{\partial \mu_i} = \Sigma^{-1}\sum_{k=1}^{n}(x_k - \mu_i)\,\mathbb{1}_{\{y_k=i\}}$$

It is zero when

$$\begin{aligned}\frac{\partial L}{\partial \mu_i} = 0 \text{ iff } &\sum_{k=1}^{n}(x_k - \mu_i)\,\mathbb{1}_{\{y_k=i\}} = 0 \\ \text{iff } \mu_i &= \frac{1}{n_i}\sum_{k=1}^{n} x_k \mathbb{1}_{\{y_k=i\}} \triangleq \widehat{\mu}_i\end{aligned}$$

Hence, the MLE estimator for μ_i is the empirical mean relatively to the class i. The gradient of L with respect to Σ is equal to the gradient of $g : \Sigma \mapsto -\frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{k=1}^{n}(x_k - \widehat{\mu}_{y_k})^T \Sigma^{-1}(x_k - \widehat{\mu}_{y_k})$. Note that we replace the μ_i by their estimator $\widehat{\mu}_i$. As the second term is a sum of real numbers, one can write:

$$\begin{aligned}g(\Sigma) &= -\frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{k=1}^{n}\mathrm{Tr}\left[(x_k - \widehat{\mu}_{y_k})^T \Sigma^{-1}(x_k - \widehat{\mu}_{y_k})\right] \\ &= -\frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{k=1}^{n}\mathrm{Tr}\left[\Sigma^{-1}(x_k - \widehat{\mu}_{y_k})(x_k - \widehat{\mu}_{y_k})^T\right] \\ &= -\frac{n}{2}\left(\log|\Sigma| + \mathrm{Tr}\left(\Sigma^{-1}\tilde{\Sigma}\right)\right)\end{aligned}$$

where $\tilde{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k - \widehat{\mu}_{y_k})(x_k - \widehat{\mu}_{y_k})^T$ is the empirical covariance matrix.

Derivative of the first term of g. Let's compute the gradient of the function $X \in \mathcal{S}_{++}^{n}(\mathbb{R}) \longmapsto \log|X|$. For $X, H \in \mathcal{S}_{++}^{n}(\mathbb{R})$ such that H is small (e.g. its Frobenius norm is small), one has:

$$\begin{aligned}\log\det(X+H) &= \log\det\left[X^{1/2}\left(I + X^{-1/2}HX^{-1/2}\right)X^{1/2}\right] \\ &= \log\det X + \log\det\left(I + X^{-1/2}HX^{-1/2}\right) \\ &= \log\det X + \sum_{i=1}^{n}\log(1+\lambda_i)\end{aligned}$$

where λ_i is the i-th eigenvalue of $X^{-1/2}HX^{-1/2}$. H is small, so are its eigenvalues. Hence, one can make the first-order approximation $\log(1+\lambda_i) = \lambda_i + \underset{\lambda_i \to 0}{o}(\lambda_i)$, which gives:

$$\begin{aligned}\log\det(X+H) &= \log\det X + \operatorname{Tr}\left(X^{-1/2}HX^{-1/2}\right) + \underset{||H||\to 0}{o}(H)\\ &= \log\det X + \operatorname{Tr}\left(X^{-1}H\right) + \underset{||H||\to 0}{o}(H)\end{aligned}$$

Because, X^{-1} is symmetric, the second term corresponds to the Frobenius inner product between X^{-1} and H. It is possible to identify the gradient of this function with respect to X: it is equal to X^{-1}.

<u>Derivative of the second term of g.</u> Let's compute the gradient of $X \in \mathcal{S}_{++}^n(\mathbb{R}) \longmapsto \operatorname{Tr}\left(X^{-1}\tilde{\Sigma}\right)$. For $X, H \in \mathcal{S}_{++}^n(\mathbb{R})$ such that H is small (for $I + X^{-1}H$ to be invertible, it is supposed small enough so that $\left|\left|X^{-1}H\right|\right| < 1$ holds), one has:

$$\operatorname{Tr}\left[(X+H)^{-1}\tilde{\Sigma}\right] = \operatorname{Tr}\left[\left(I + X^{-1}H\right)^{-1}X^{-1}\tilde{\Sigma}\right]$$

Because $\left|\left|X^{-1}H\right|\right| < 1$ (and $||\cdot||$ a sub-multiplicative norm), one has:

$$\left(I + X^{-1}H\right)^{-1} = \sum_{k=0}^{+\infty}(-1)^k\left(X^{-1}H\right)^k$$

which gives:

$$\begin{aligned}\operatorname{Tr}\left[(X+H)^{-1}\tilde{\Sigma}\right] &= \underbrace{\operatorname{Tr}\left(X^{-1}\tilde{\Sigma}\right)}_{k=0} + \underbrace{\operatorname{Tr}\left[-X^{-1}HX^{-1}\tilde{\Sigma}\right]}_{k=1} + \underbrace{\underset{||H||\to 0}{o}(H)}_{k>2}\\ &= \operatorname{Tr}\left(X^{-1}\tilde{\Sigma}\right) + \operatorname{Tr}\left[\left(-X^{-1}\tilde{\Sigma}X^{-1}\right)H\right] + \underset{||H||\to 0}{o}(H)\end{aligned}$$

Once again, it is possible to identify the gradient of this function with respect to X, which is equal to $-X^{-1}\tilde{\Sigma}X^{-1}$ (because the matrices are symmetric).

Hence, the gradient of the log-likelihood with respect to Σ is:

$$\begin{aligned}\frac{\partial L}{\partial \Sigma} &= \frac{\partial g}{\partial \Sigma}\\ &= -\frac{n}{2}\left(\Sigma^{-1} - \Sigma^{-1}\tilde{\Sigma}\Sigma^{-1}\right)\end{aligned}$$

and we have

$$\begin{aligned}\frac{\partial L}{\partial \Sigma} = 0 &\text{ iff } \Sigma^{-1} - \Sigma^{-1}\tilde{\Sigma}\Sigma^{-1} = 0\\ &\text{ iff } \Sigma = \tilde{\Sigma} \triangleq \widehat{\Sigma}\end{aligned}$$

This shows that the $\left(\widehat{\pi}, \widehat{\mu}_0, \widehat{\mu}_1, \widehat{\Sigma}\right)$ obtained is the only stationary point of the log-likelihood. To check that it corresponds to a maximum, one can compute the Hessian matrix. To sum up, the maximum likelihood estimators are given by:

$$\boxed{\begin{aligned}\widehat{\pi} &= \frac{n_1}{n}\\ \widehat{\mu}_i &= \frac{1}{n_i}\sum_{k=1}^{n} x_k \mathbb{1}_{\{y_k = i\}} && \text{for } i \in \{0,1\}\\ \widehat{\Sigma} &= \frac{1}{n}\sum_{k=1}^{n}\left(x_k - \widehat{\mu}_{y_k}\right)\left(x_k - \widehat{\mu}_{y_k}\right)^T\end{aligned}}$$

b One has:

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

where $w=\Sigma^{-1}\left(\mu_1-\mu_0\right)$ because Σ^{-1} is symmetric and $b=-\frac{1}{2}\left(\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0\right)+\ln\left(\frac{\pi}{1-\pi}\right)$. The first line comes from the Bayes' theorem. The second line just comes from the definition of the distributions and some simplifications. Hence, denoting σ the sigmoid function,

$$\boxed{\begin{aligned} p\left(y=1|x\right)=\sigma\left(w^Tx+b\right) \text{ where } w &= \Sigma^{-1}\left(\mu_1-\mu_0\right) \\ b &= -\frac{1}{2}\left(\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0\right)+\ln\left(\frac{\pi}{1-\pi}\right)\end{aligned}}$$

c The MLE for this model are implemented in the Python class `LDA`, in the file `lda.py`. In the following table are the numerical values of the parameters learnt for the LDA model.

Parameter	Dataset A	Dataset B	Dataset C
w_0	2.019	1.703	0.302
w_1	-6.378	-3.043	-2.868
b	31.951	7.886	20.574

Besides, in figure 2.1 are shown the data as well as the line defined by $p\left(y=1|x\right)=0.5$.

2.2 Logistic regression

Implement logistic regression for an affine function $f(x)=w^Tx+b$ (do not forget the constant term).

a Give the numerical values of the parameters learnt.

b Represent graphically the data as a cloud point in $\mathbb{R}^2$ as well as the line defined by the equation

$$p\left(y=1|x\right)=0.5$$

Solution.

For the logistic regression, we will assume: $y|x\sim\mathcal{B}(\theta)$, a Bernoulli law with parameter $\theta=\sigma(w^Tx)$.

In this case, the offset b is incorporated in the weight vector w with: $\begin{cases} w := \begin{pmatrix} b \\ | \\ w \\ | \end{pmatrix} \\ X := \begin{pmatrix} 1 & — x_1^T — \\ \vdots & \vdots \\ 1 & — x_n^T — \end{pmatrix}\end{cases}$ where $\{(x_i,y_i)\}_{i\in[\![1,n]\!]}\subset\mathbb{R}^d\times\{0,1\}$ are n i.i.d. samples.

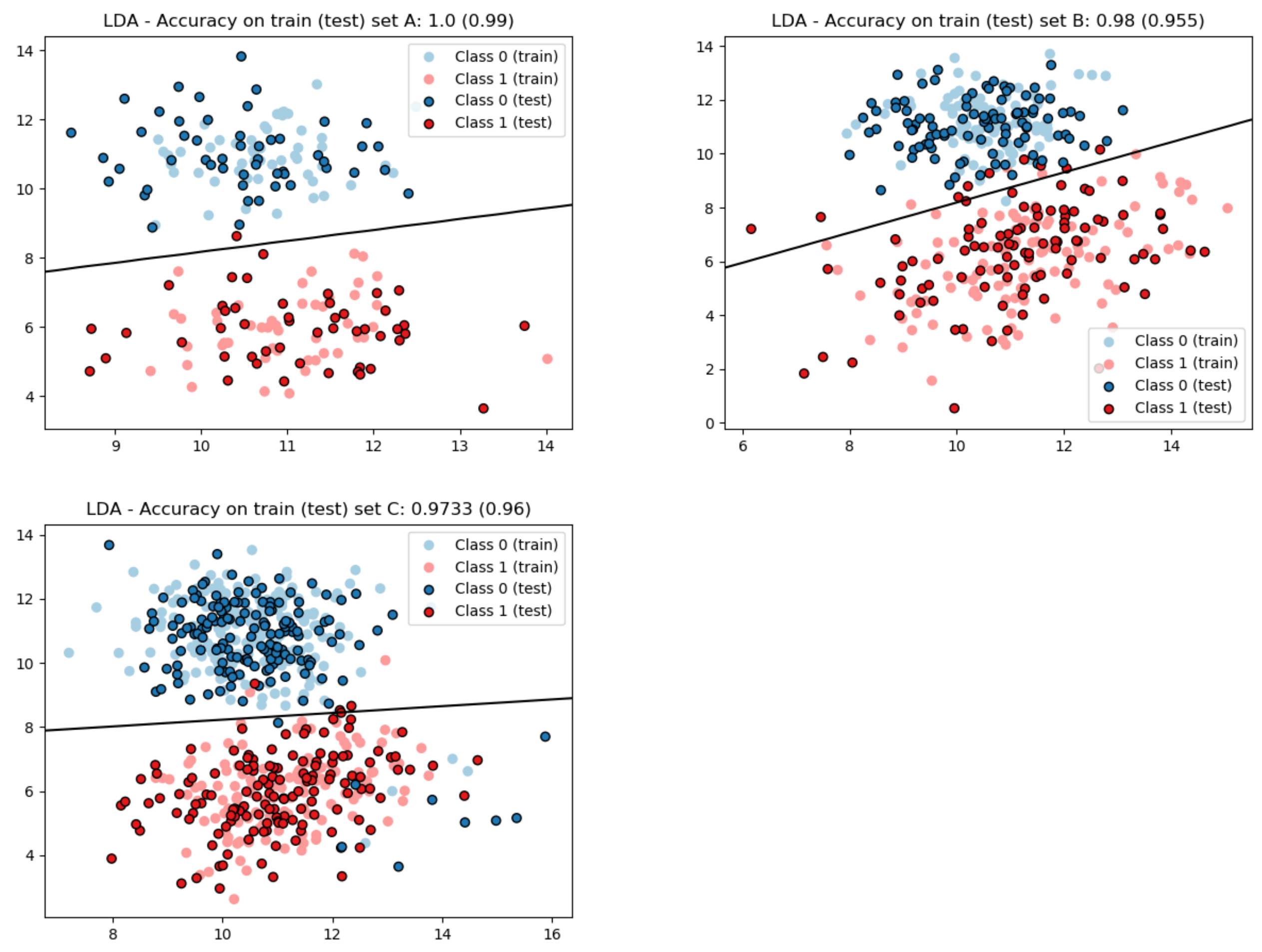

Figure 2.1: The lines are defined by the equation $p(y = 1|x) = 0.5$ for the LDA Model.

With this in mind, we find:

$$\begin{aligned} p\,(y|x) &= \theta^y (1-\theta)^{1-y} \\ &= \sigma\left(w^T x\right)^y \left(1 - \sigma\left(w^T x\right)\right)^{1-y} \end{aligned}$$

As such, the log-likelihood is given by:

$$\begin{aligned} L\,(w) &= \sum_{k=1}^{n} y_k \log \sigma(w^T x_k) + (1-y_k)\log(1-\sigma(w^T x_k)) \\ &= \sum_{k=1}^{n} y_k \log \sigma(w^T x_k) + (1-y_k)\log \sigma(-w^T x_k) \end{aligned}$$

We can calculate the gradient since $w \mapsto w^T x$ is linear and $z \mapsto \log \sigma(z) = -\log(1+e^{-z})$ is concave:

$$\begin{aligned} \nabla_w L\,(w) &= \sum_{k=1}^{n} y_k \frac{\nabla_w(\sigma(w^T x_k))}{\sigma(w^T x_k)} + (1-y_k)\frac{\nabla_w(\sigma(-w^T x_k))}{\sigma(-w^T x_k)} \\ &= \sum_{k=1}^{n} y_k x_k \frac{\nabla_w(\sigma)(w^T x_k)}{\sigma(w^T x_k)} - (1-y_k) x_k \frac{\nabla_w(\sigma)(-w^T x_k)}{\sigma(-w^T x_k)} \end{aligned}$$

As σ' is given by $\sigma' = \sigma\,(1-\sigma)$,

$$\begin{aligned}
\nabla_w L(w) &= \sum_{k=1}^{n} y_k x_k \frac{\sigma(w^T x_k)(1-\sigma(w^T x_k))}{\sigma(w^T x_k)} - (1-y_k) x_k \frac{\sigma(-w^T x_k)(1-\sigma(-w^T x_k))}{\sigma(-w^T x_k)} \\
&= \sum_{k=1}^{n} y_k x_k (1-\sigma(w^T x_k)) - (1-y_k) x_k \sigma(w^T x_k) \\
&= \sum_{k=1}^{n} x_k \left[y_k - y_k \sigma(w^T x_k)) - \sigma(w^T x_k) + y_k \sigma(w^T x_k) \right] \\
&= \sum_{k=1}^{n} x_k \left[y_k - \sigma(w^T x_k) \right]
\end{aligned}$$

The equation $\nabla_w L(w) = 0 \iff \sum_{k=1}^{n} x_k \left[y_k - \sigma(w^T x_k) \right] = 0$ is nonlinear, so we can use the Newton-Raphson method to find a good root approximation. Therefore, we need to compute the Hessian.

$$\begin{aligned}
\operatorname{Hess} L(w) &= \sum_{k=1}^{n} -x_k \sigma(w^T x_k) \sigma(-w^T x_k) x_k^T \\
&= -X^T \operatorname{Diag}\left(\sigma(w^T x_k) \sigma(-w^T x_k) \right) X
\end{aligned}$$

where The Newton-Raphson method then consists in updating w as follows:

$$w_{t+1} = w_t - \operatorname{Hess} L(w_t)^{-1} \nabla_w L(w_t)$$

a In the following table are the numerical values of the parameters learnt for the Logistic Regression model.

Parameter	Dataset A	Dataset B	Dataset C
w_0	16.999	1.842	-0.277
w_1	-66.605	-3.714	-1.914
b	378.214	13.430	18.807
Number of iterations	200	11	9
Convergence	False	True	True

b This model is implemented in the Python class `LogisticRegression`, in the file `log_reg.py`. In the figure 2.2 are shown the data as well as the line defined by $p(y=1|x) = 0.5$.

2.3 Linear regression

Consider class y as a real valued variable taking the values 0 and 1 only. Implement linear regression (for an affine function $f(x) = w^T x + b$) by solving the normal equations.

a Provide the numerical values of the learnt parameters.

b Represent graphically the data as a point cloud in $\mathbb{R}^2$ as well as the line defined by the equation

$$p(y=1|x) = 0.5$$

Solution.

Given a dataset $\{(x_i, y_i)\}_{i \in [\![1,n]\!]} \subset \mathbb{R}^d \times \{0,1\}$, a linear regression can be obtained solving an ordinary least squares (OLS) problem:

$$(w,b) \in \operatorname*{arg\,min}_{w \in \mathbb{R}^d, b \in \mathbb{R}} \sum_{i=1}^{n} \left(y_i - \left(w^T x_i + b \right) \right)^2$$

Note that this is not an equality as the solution is, *a priori*, not unique.

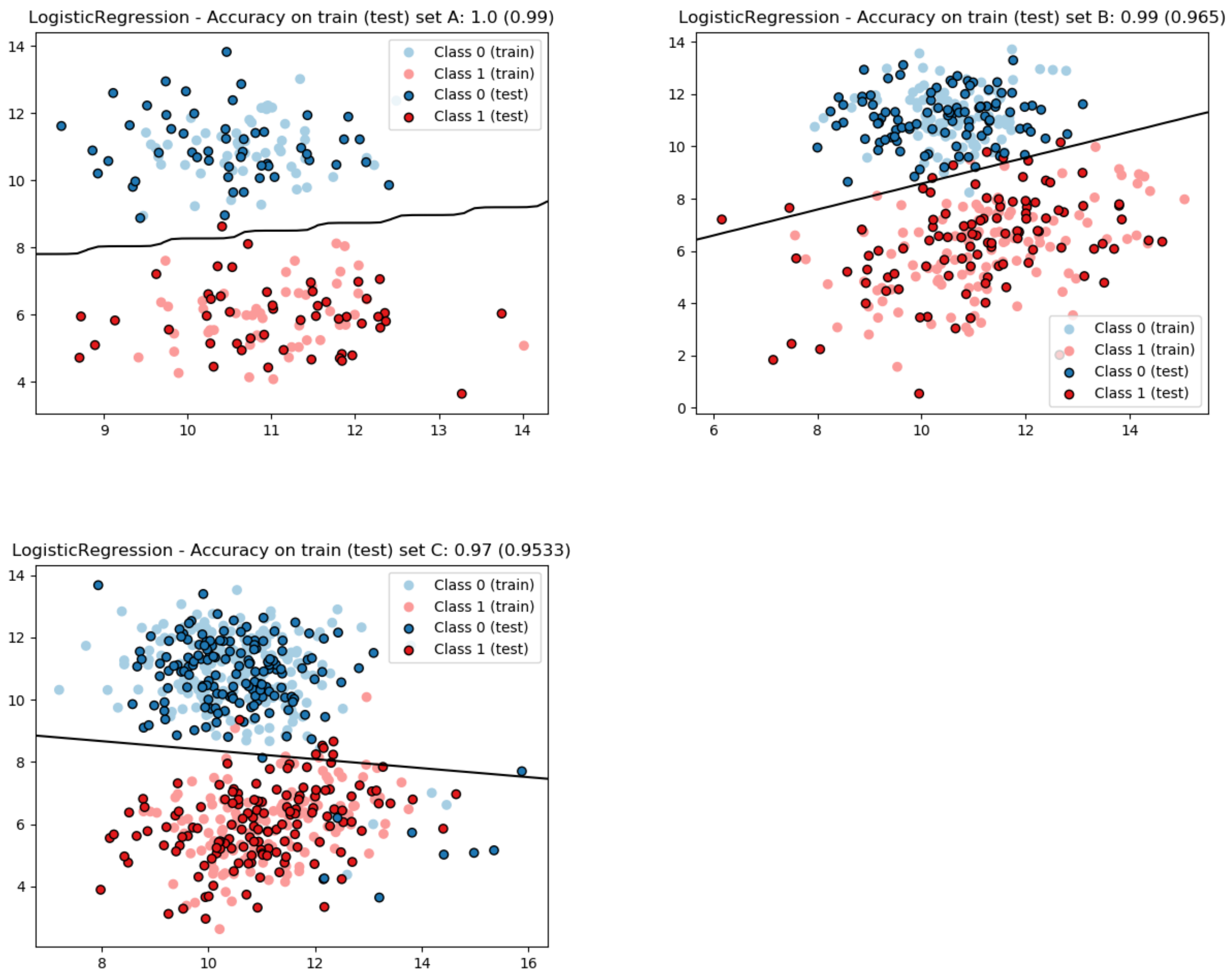

Figure 2.2: The lines are defined by the equation $p(y=1|x)=0.5$ for the Logistic Regression model.

Using matrices, it is equivalent to solve:

$$\theta \in \underset{\theta\in\mathbb{R}^{d+1}}{\arg\min} ||Y - X\theta||_2^2$$

where $\theta = (b, w_1, \dots, w_d)^T$, Y is the n-dimensional vector formed by the y_i's, $X \in \mathbb{R}^{n\times(d+1)}$ the matrix defined by

$$X = \begin{pmatrix} 1 & — x_1^T — \\ \vdots & \vdots \\ 1 & — x_n^T — \end{pmatrix}$$

The uniqueness of the solution depends on $\mathrm{Ker}\,(X)$ (if $\mathrm{Ker}\,(X) = \{0\}$ it is unique, otherwise the set of solutions is given by $\widehat{\theta} + \mathrm{Ker}\,(X)$ where $\widehat{\theta}$ is a particular solution). Solving the so-called normal equation, $X^T\left(Y - \widehat{Y}\right) = 0$ where $\widehat{Y} = X\widehat{\theta}$ and $\widehat{\theta}$ is a particular solution, one can show that a solution is given by:

$$\widehat{\theta} = \left(X^TX\right)^+ X^TY$$

where $\left(X^TX\right)^+$ is the Moore-Penrose inverse of X^TX (which always exists), that corresponds to the usual inverse when X^TX is invertible.

a In the following table are the numerical values of the parameters learnt for the Linear Regression model.

Parameter	Dataset A	Dataset B	Dataset C
w_0	0.056	0.083	0.017
w_1	-0.176	-0.148	-0.159
b	1.383	0.882	1.640

b The MLE for this model are implemented in the Python class `LinearRegression`, in the file `lin_reg.py`. In the figure 2.3 are shown the data as well as the line defined by $p\,(y = 1|x) = 0.5$.

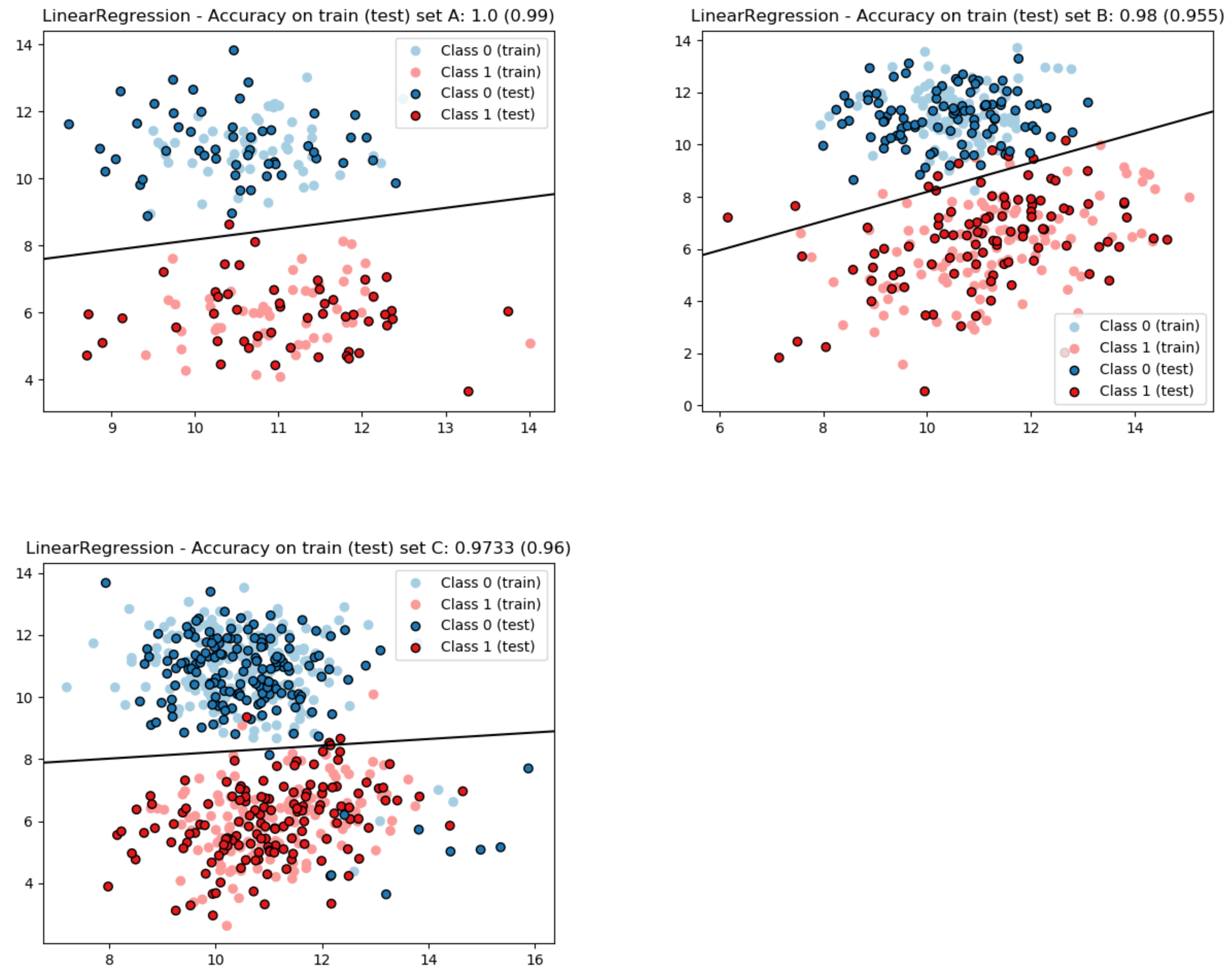

Figure 2.3: The lines are defined by the equation $p(y = 1|x) = 0.5$ for the Linear Regression model.

2.4 Application

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

a 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.

b Compare the performances of the different methods on the three datasets. Is the misclassification error larger, smaller, or similar on the training and test data? Why? Which methods yield very similar/dissimilar results? Which methods yield the best results on the different datasets? Provide an interpretation.

Solution.

a Below are shown the misclassification errors on each dataset, for each model:

LDA

Misclassif. error	Dataset A	Dataset B	Dataset C
Train set	0	0.02	0.027
Test set	0.01	0.045	0.04

Logistic Regression

Misclassif. error	Dataset A	Dataset B	Dataset C
Train set	0	0.01	0.03
Test set	0.01	0.035	0.047

Linear Regression

Misclassif. error	Dataset A	Dataset B	Dataset C
Train set	0	0.02	0.027
Test set	0.01	0.045	0.04

b
- The misclassification error is always larger on the test set. On the dataset A, it is quite similar. It is due to the fact that the data is linearly separable and the points closest to the border are all in the training set except for one (hence the one misclassfied point). Datasets B and C, however, are a bit more complex and the errors on the training and testing sets tend to be quite different.
- LDA and Linear Regression yield extremely similar results in all three cases. The Gaussian model represents datasets A and B accurately, so the LDA model does well on both. The logistic regression does better in practice for both datasets, but the decision boundaries yielded by the LDA model seem more logical.

2.5 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.

$$y \sim \text{Bernoulli}\,(\pi)\,, x|y = i \sim \text{Normal}\,(\mu_i, \Sigma_i)$$

Implement the maximum likelihood estimator and apply it to the data.

a Provide the numerical values of the learnt parameters.

b Represent graphically the data as well as the conic defined by

$$p\,(y = 1|x) = 0.5$$

c Compute the misclassification error for QDA for both train and test data.

d Comment the results as previously.

Solution.

For the QDA model, one can proceed similarly as with the LDA model. Let's highlight the changes due to the covariance matrices. Defining $\theta = (\pi, \mu_0, \mu_1, \Sigma_0, \Sigma_1)$, The log-likelihood is now given by:

$$\begin{aligned} L\,(\theta) = &-\frac{nd}{2}\log 2\Pi - \frac{n_0}{2}\log|\Sigma_0| - \frac{n_1}{2}\log|\Sigma_1| \\ &+ \sum_{k=1}^{n}\left\{-\frac{1}{2}\left(x_k - \mu_{y_k}\right)^T \Sigma_{y_k}^{-1}\left(x_k - \mu_{y_k}\right) + y_k \log(\pi) + (1-y_k)\log(1-\pi)\right\}\end{aligned}$$

<u>MLE for π.</u> It is exactly the same since the gradient does not depend on the covariance matrix.

<u>MLE for μ_i, $i \in \{0,1\}$.</u> The gradient with respect to μ_i becomes:

$$\frac{\partial L}{\partial \mu_i} = \Sigma_i^{-1} \sum_{k=1}^{n} (x_k - \mu_i)\, \mathbb{1}_{\{y_k = i\}}$$

which leads to the same result than previously.

<u>MLE for Σ_i, $i \in \{0,1\}$.</u> Let $g_i : \Sigma \mapsto -\frac{n_i}{2} \log |\Sigma| - \frac{1}{2} \sum_{k:y_k=i} (x_k - \widehat{\mu}_i)^T \Sigma^{-1} (x_k - \widehat{\mu}_i)$. The gradient of L with respect to Σ_i is:

$$\begin{aligned} \frac{\partial L}{\partial \Sigma_i} &= \frac{\partial g_i}{\partial \Sigma} (\Sigma_i) \\ &= -\frac{n_i}{2} \left(\Sigma_i^{-1} - \Sigma_i^{-1} \tilde{\Sigma}_i \Sigma_i^{-1} \right) \end{aligned}$$

where $\tilde{\Sigma}_i = \frac{1}{n_i} \sum_{k:y_k=i} (x_k - \widehat{\mu}_i)(x_k - \widehat{\mu}_i)^T$ and we have

$$\frac{\partial L}{\partial \Sigma_i} = 0 \text{ iff } \Sigma_i = \tilde{\Sigma}_i \triangleq \widehat{\Sigma}_i$$

Hence, the MLE estimators for the QDA model are given by:

$$\boxed{\begin{aligned} \widehat{\pi} &= \frac{1}{n} \sum_{k=1}^{n} y_k & \\ \widehat{\mu}_i &= \frac{1}{n_i} \sum_{k=1}^{n} x_k \mathbb{1}_{\{y_k=i\}} & \text{for } i \in \{0,1\} \\ \widehat{\Sigma}_i &= \frac{1}{n_i} \sum_{k=1}^{n} (x_k - \widehat{\mu}_i)(x_k - \widehat{\mu}_i)^T \mathbb{1}_{\{y_k=i\}} & \text{for } i \in \{0,1\} \end{aligned}}$$

Besides,

$$\begin{aligned} p(y=1|x) &= \frac{p(x|y=1)\,p(y=1)}{p(x|y=0)p(y=0) + p(x|y=1)p(y=1)} \\ &= \frac{\exp\left[-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)\right] \pi}{\exp\left[-\frac{1}{2}(x-\mu_0)^T \Sigma_0^{-1} (x-\mu_0)\right](1-\pi) + \exp\left[-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)\right]\pi} \\ &= \frac{1}{\left(\frac{1}{\pi} - 1\right) \exp\left[\frac{1}{2}\left\{(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1) - (x-\mu_0)^T \Sigma_0^{-1} (x-\mu_0)\right\}\right] + 1} \\ &= \frac{1}{\exp\left[\frac{1}{2}\left\{x^T\left(\Sigma_1^{-1} - \Sigma_0^{-1}\right)x + 2\mu_0^T \Sigma_0^{-1} x - 2\mu_1^T \Sigma_1^{-1} x + \mu_1^T \Sigma_1^{-1} \mu_1 - \mu_0^T \Sigma_0^{-1} \mu_0\right\} + \ln\left(\frac{1}{\pi} - 1\right)\right] + 1} \\ &= \frac{1}{\exp\left[\frac{1}{2}x^T\left(\Sigma_1^{-1} - \Sigma_0^{-1}\right)x + \left(\mu_0^T \Sigma_0^{-1} - \mu_1^T \Sigma_1^{-1}\right)x + \frac{1}{2}\left(\mu_1^T \Sigma_1^{-1} \mu_1 - \mu_0^T \Sigma_0^{-1} \mu_0\right) + \ln\left(\frac{1}{\pi} - 1\right)\right] + 1} \\ &= \frac{1}{1 + \exp\left(-(-\frac{1}{2}x^T\left(\Sigma_1^{-1} - \Sigma_0^{-1}\right)x + w^T x + b)\right)} \\ &= \sigma\left(-\frac{1}{2}x^T\left(\Sigma_1^{-1} - \Sigma_0^{-1}\right)x + w^T x + b\right) \end{aligned}$$

where $w = \Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0$ because Σ_1^{-1} and Σ_0^{-1} are symmetric and $b = -\frac{1}{2}\left(\mu_1^T \Sigma_1^{-1} \mu_1 - \mu_0^T \Sigma_0^{-1} \mu_0\right) + \ln\left(\frac{\pi}{1-\pi}\right)$.

a In the following table are the numerical values of the parameters learnt for the QDA model.

Parameter	Dataset A	Dataset B	Dataset C
w_0	-6.069	-8.533	-2.898
w_1	-8.900	-9.339	-7.010
b	87.927	96.234	54.484

b The MLE for this model are implemented in the Python class `QDA`, in the file `qda.py`. In the figure 2.4 are shown the data as well as the conic defined by $p(y=1|x)=0.5$.

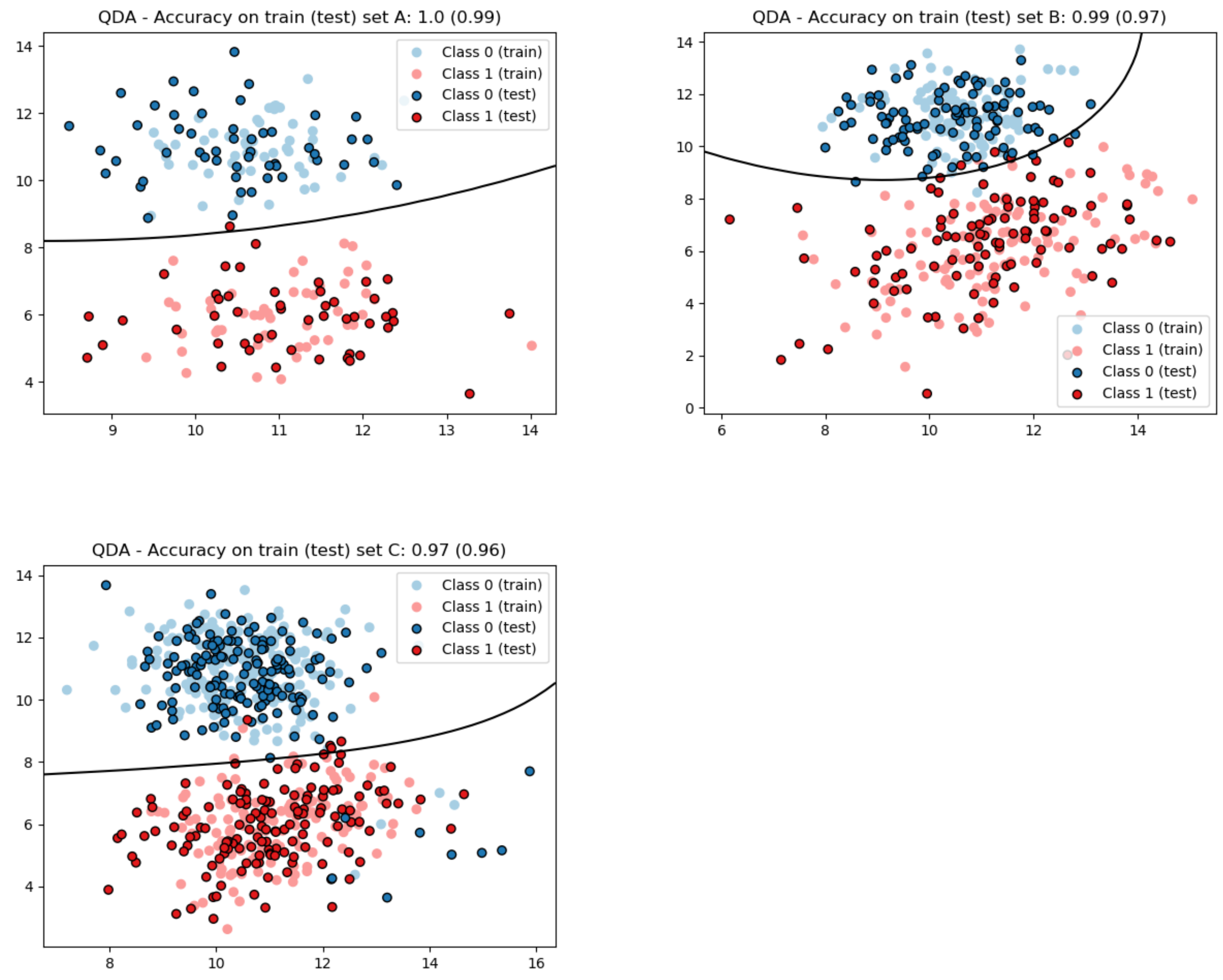

Figure 2.4: The conics are defined by the equation $p(y=1|x)=0.5$ for the QDA Model.

c Here are the results obtained with the QDA model:

Misclassif. error	Dataset A	Dataset B	Dataset C
Train set	0	0.01	0.03
Test set	0.01	0.03	0.04

d We see that in most of the cases, the QDA performs better on the datasets B and C. As it seems that the datasets B and C are generated from two Gaussians with different means and different covariance matrices (with noisy examples for the dataset C), QDA model is well adapted. For the dataset A, it is not useful, as the data are linearly separable.