Probabilistic Graphical Models

Homework 1

Authors

Bastien Déchamps
bastien.dechamps@eleves.enpc.fr

and

Mathieu Orhan
mathieu.orhan@eleves.enpc.fr

Supervisors

Pierre Latouche

and

Nicolas Chopin

Abstract

We derive estimators and algorithms for different classification models : LDA, linear and logistic regression, and QDA. We compare them on three datasets and provide numerical values of learnt parameters and accuracies. All the methods are fully implemented in Python and code to reproduce the experiments will be available in December, 2019 at `https://github.com/mathieuorhan/PGM_HW1`.

Contents

1 Learning in discrete graphical models

Let $\mathcal{D} = \{(x_i, z_i)\}_{0 \leq i \leq N}$ be N i.i.d. observations of (x, z). The likelihood of the sample can be written:

$$\begin{aligned} L_{\mathcal{D}}(\pi, \theta) &= \prod_{i=1}^{N} p(x_i, z_i | \pi, \theta) \\ &= \prod_{i=1}^{N} p(z_i | \pi, \theta) p(x_i | z_i, \pi, \theta) \\ &= \prod_{i=1}^{N} \left[\left(\prod_{m=1}^{M} \pi_m^{\mathbb{1}_{z_i=m}} \right) \left(\prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{mk}^{\mathbb{1}_{z_i=m}\mathbb{1}_{x_i=k}} \right) \right] \end{aligned}$$

To compute the maximum likelihood estimator of θ and π, we need to solve the following optimization problem:

$$\begin{aligned} \underset{\theta,\pi}{\text{Minimize}} \quad & -\sum_{i=1}^{N} \log p(X = x_i, Z = z_i | \theta, \pi) \\ \text{s.t.} \quad & \sum_{m=1}^{M} \pi_m = 1, \\ & \forall m, 1 \leq m \leq M, \sum_{k=1}^{K} \theta_{mk} = 1. \end{aligned} \tag{1}$$

Let $\mathcal{L}$ be the Lagrangian of the problem (1) with multipliers λ_π and λ_{θ_m}. Let $\mathbb{1}_A$ denotes the indicator of A.

$$\begin{aligned} \mathcal{L}(\pi, \theta) = & -\sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M} \log(\theta_{mk}) \mathbb{1}_{x_i=k} \mathbb{1}_{z_i=m} \\ & + \sum_{i=1}^{N} \sum_{m=1}^{M} \log(\pi_m) \mathbb{1}_{z_i=m} \\ & + \lambda_\pi (\sum_{m=1}^{M} \pi_m - 1) \\ & + \sum_{m=1}^{M} \lambda_{\theta_m} \sum_{k=1}^{K} (\theta_{mk} - 1) \end{aligned} \tag{2}$$

Let's set the derivatives of $\mathcal{L}$ to 0 from (2):

$$\frac{\partial \mathcal{L}(\pi, \theta)}{\partial \pi_m} = 0 \implies -\frac{1}{\pi_m} \sum_{i=1}^{N} \mathbb{1}_{z_i=m} + \lambda_\pi = 0 \tag{3}$$

$$\frac{\partial \mathcal{L}(\pi, \theta)}{\partial \pi_m} = 0 \implies -\frac{1}{\theta_{mk}} \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \mathbb{1}_{x_i=k} + \lambda_{\theta_m} = 0 \tag{4}$$

From (3) and the first constraint of (1), we can compute λ_π :

$$\begin{aligned} \pi_m \lambda_\pi &= \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \\ \sum_{m=1}^{M} \pi_m \lambda_\pi &= \sum_{m=1}^{M} \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \\ \lambda_\pi &= \sum_{m=1}^{M} \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \\ \lambda_\pi &= N \end{aligned} \tag{5}$$

Similarly, we can compute the other multipliers λ_{θ_m} from (4) and (1).

$$\begin{aligned} \sum_{k=1}^{K} \theta_{mk} \lambda_{\theta_m} &= \sum_{k=1}^{K} \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \mathbb{1}_{x_i=k} \\ \lambda_{\theta_m} &= \sum_{k=1}^{K} \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \mathbb{1}_{x_i=k} \\ \lambda_{\theta_m} &= \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \end{aligned} \tag{6}$$

Injecting (5) (resp. (6)) in (3) (resp. (4)), we can finally express the maximum likelihood estimator.

$$\boxed{[\hat{\pi}_m]_{\text{MLE}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{z_i=m}}$$

$$\boxed{\left[\hat{\theta}_{mk}\right]_{\text{MLE}} = \frac{1}{N \left[\hat{\pi}_m\right]_{\text{MLE}}} \sum_{i=1}^{N} \mathbb{1}_{z_i=m} \mathbb{1}_{x_i=k}}$$

2 Linear Classification

2.1 Generative model (LDA)

Here we assume $y \sim \mathcal{B}(\pi)$ and $x|\{y=i\} \sim \mathcal{N}(\mu_i, \Sigma)$ for $i \in \{0, 1\}$.

(a) Let $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$ be a i.i.d. sample of (x, y). Its likelihood can be written:

$$\begin{aligned} L_{\mathcal{D}}(\pi, \mu_0, \mu_1, \Sigma) &= \prod_{i=1}^{N} p(x_i, y_i | \pi, \mu_0, \mu_1) \\ &= \prod_{i=1}^{N} p(y_i | \pi, \mu_0, \mu_1, \Sigma) p(x_i | y_i \pi, \mu_0, \mu_1, \Sigma) \\ &= \prod_{i=1}^{N} \pi^{y_i} (1-\pi)^{1-y_i} p_1(x_i)^{y_i} p_0(x_i)^{1-y_i} \end{aligned}$$

where $p_i(x) = \mathcal{N}(x | \mu_i, \Sigma) = \frac{1}{2\pi\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(x-\mu_i)\Sigma^{-1}(x-\mu_i)\right)$. By taking the logarithm of the likelihood, we have:

$$\begin{aligned} l_{\mathcal{D}}(\pi, \mu_0, \mu_1, \Sigma) = &\sum_{i=1}^{N} y_i \log \pi + (1-y_i) \log(1-\pi) + \frac{1}{2} \log \det \Sigma^{-1} \\ &- \frac{y_i}{2}(x_i - \mu_1)\Sigma^{-1}(x_i - \mu_1) \\ &- \frac{1-y_i}{2}(x_i - \mu_0)\Sigma^{-1}(x_i - \mu_0) + \text{cste} \end{aligned}$$

By differentiating $l_{\mathcal{D}}$, we get:

$$\begin{aligned} & \frac{\partial l_{\mathcal{D}}}{\partial \pi}(\pi, \mu_0, \mu_1, \Sigma) = \frac{1}{\pi} \sum_{i=1}^{N} y_i - \frac{1}{1-\pi} \sum_{i=1}^{N} (1-y_i) = 0 \\ \iff \quad & (1-\pi)\bar{y} - \pi(1-\bar{y}) = 0 \end{aligned}$$

and thus the MLE estimator for π is

$$\boxed{\hat{\pi}_{\text{MLE}} = \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i} \tag{7}$$

$$\begin{aligned} & \frac{\partial l_{\mathcal{D}}}{\partial \mu_1}(\pi, \mu_0, \mu_1, \Sigma) = \sum_{i=1}^{N} y_i \Sigma^{-1}(x_i - \mu_1) = 0 \\ \iff \quad & \sum_{i=1}^{N} y_i x_i = \left(\sum_{i=1}^{N} y_i \right) \mu_1 \end{aligned}$$

and thus, if we denote $N_1 = \sum_{i=1}^{N} y_i$, the MLE estimator for μ_1 is

$$\boxed{[\hat{\mu}_1]_{\text{MLE}} = \frac{1}{N_1} \sum_{i=1}^{N} y_i x_i} \tag{8}$$

By defining $N_0 = \sum_{i=1}^{N}(1-y_i)$, the same calculations give the the MLE estimator for μ_0

$$\boxed{[\hat{\mu}_0]_{\text{MLE}} = \frac{1}{N_0}\sum_{i=1}^{N}(1-y_i)x_i} \tag{9}$$

To compute the MLE for Σ, let's differentiate the log-likelihood $l_{\mathcal{D}}$ with respect to the precision matrix $\Lambda \doteq \Sigma^{-1}$. The function $f : A \to \log\det A$ is differentiable on $\mathcal{S}_2^{++}$, and:

$$\forall A \in \mathcal{S}_2^{++}, \quad \frac{\partial f}{\partial A}(A) = A^{-1} \tag{10}$$

Thus, we have:

$$\begin{aligned}\frac{\partial l_{\mathcal{D}}}{\partial \Lambda}(\pi, \mu_0, \mu_1, \Sigma) &= \sum_{i=1}^{N}\left[\frac{1}{2}\Lambda^{-1} - \frac{y_i}{2}(x_i-\mu_1)(x_i-\mu_1)^\top - \frac{1-y_i}{2}(x_i-\mu_0)(x_i-\mu_0)^\top\right] \\ &= \frac{N}{2}\Sigma - \frac{N_1}{2}\tilde{\Sigma}_1 - \frac{N_0}{2}\tilde{\Sigma}_0\end{aligned}$$

where $\tilde{\Sigma}_0 = \frac{1}{N_0}\sum_{i=1}^{N}(1-y_i)(x_i-\mu_0)(x_i-\mu_0)^\top$ and $\tilde{\Sigma}_1 = \frac{1}{N_1}\sum_{i=1}^{N} y_i(x_i-\mu_1)(x_i-\mu_1)^\top$.

Then, by setting $\frac{\partial l_{\mathcal{D}}}{\partial \Lambda}$ to 0, we have the following MLE estimator for Σ:

$$\boxed{\hat{\Sigma}_{\text{MLE}} = \frac{N_0}{N}\tilde{\Sigma}_0 + \frac{N_1}{N}\tilde{\Sigma}_1} \tag{11}$$

(b) The Bayes formula gives us:

$$\begin{aligned}p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0) + p(x|y=1)p(y=1)} \\ &= \frac{\pi p_1(x)}{(1-\pi)p_0(x) + \pi p_1(x)} \\ &= \frac{1}{1 + \frac{1-\pi}{\pi}\frac{p_0(x)}{p_1(x)}}\end{aligned}$$

To retrieve the logistic regression form for $p(y=1|x)$, let's find (ω, b) such that:

$$\exp(-(\omega^\top x + b)) = \frac{1-\pi}{\pi}\frac{p_0(x)}{p_1(x)}$$

To find them, we compute:

$$\begin{aligned}-\log\left(\frac{1-\pi}{\pi}\frac{p_0(x)}{p_1(x)}\right) &= \frac{1}{2}(x-\mu_0)\Sigma^{-1}(x-\mu_0) - \frac{1}{2}(x-\mu_1)\Sigma^{-1}(x-\mu_1) \\ &= (\mu_1-\mu_0)^\top\Sigma^{-1}x + \frac{1}{2}\mu_0\Sigma^{-1}\mu_0 - \frac{1}{2}\mu_1\Sigma^{-1}\mu_1 - \log\left(\frac{1-\pi}{\pi}\right) \\ &= \omega^\top x + b\end{aligned}$$

where $\omega = \Sigma^{-1}(\mu_1 - \mu_0)$ and $b = \frac{1}{2}\mu_0^\top \Sigma^{-1}\mu_0 - \frac{1}{2}\mu_1^\top \Sigma^{-1}\mu_1 - \log\left(\frac{1-\pi}{\pi}\right)$.

Finally, we can rewrite:

$$\boxed{p(y=1|x) = \frac{1}{1+\exp\left(-(\omega^\top x + b)\right)} = \sigma(\omega^\top x + b)} \tag{12}$$

We apply the LDA on datasets A, B and C, and plot the results in appendix A. Numerical values of learnt parameters are available in appendix B, table 2.

2.2 Logistic regression

In this section, we derive and implement logistic regression for a affine function $f(x) = w^T x + b$ using the Newton method to find the maximum likelihood estimator l. We have a set of observations supposed i.i.d. $(x_i, y_i)_{1 \leq i \leq N}$.

$$p(y=1|x,w,b) = \sigma(f(x))$$

Let's write the log-likelihood we would like to minimize.

$$l(y|w,b) = \sum_{i=1}^{N} y_i \log(\sigma(f(x_i))) + (1-y_i)\log(1-\sigma(f(x_i)))$$

As it is well-known, there is no closed-form solution to find w and b. We thus introduce a quantity to *minimize*, the so-called cross-entropy $E(w,b)$:

$$E(w,b) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\sigma(f(x_i))) + (1-y_i)\log(1-\sigma(f(x_i)))$$

We introduce $\tilde{x} = (x_1, x_2, 1)^T$, and $\theta = (w_1, w_2, b)$ to simplify. We now compute the gradient ∇E_θ and Hessian H_θ of $E(\theta)$. With slight abuse of notation, we write $f(\tilde{x}) = f(x) = \theta^T \tilde{x}$.

$$\nabla_\theta E(\theta) = \frac{1}{N}\sum_{i=1}^{N} \left[\sigma(f(\tilde{x_i})) - y_i\right]\tilde{x_i}$$

$$H_\theta(\theta) = \frac{1}{N}\sum_{i=1}^{N} \left[\sigma(f(\tilde{x_i}))(1-\sigma(f(\tilde{x_i})))\tilde{x_i}\tilde{x_i}^T\right]$$

To solve iteratively the logistic regression, we can use the Newton method, summarized in algorithm (1). The Newton method converge in few iterations on the 3 datasets (10 for $\epsilon = 10^{-3}$ on dataset A, see figure 1). With a better precision, the Hessian becomes non-singular, as there is no regularizer in this simple implementation. The hyperplane separates the 3 test sets well, as we see on figure 3 in appendix A.

Algorithm 1 Logistic regression

Require: ϵ, precision
Require: θ, initial guess
  converged $\leftarrow 0$
  while not converged **do**
    $\theta \leftarrow \theta - H_\theta^{-1} \nabla_\theta E$
    if $E(\theta) < \epsilon$ **then**
      converged $\leftarrow 1$
    end if
  end while
  return θ

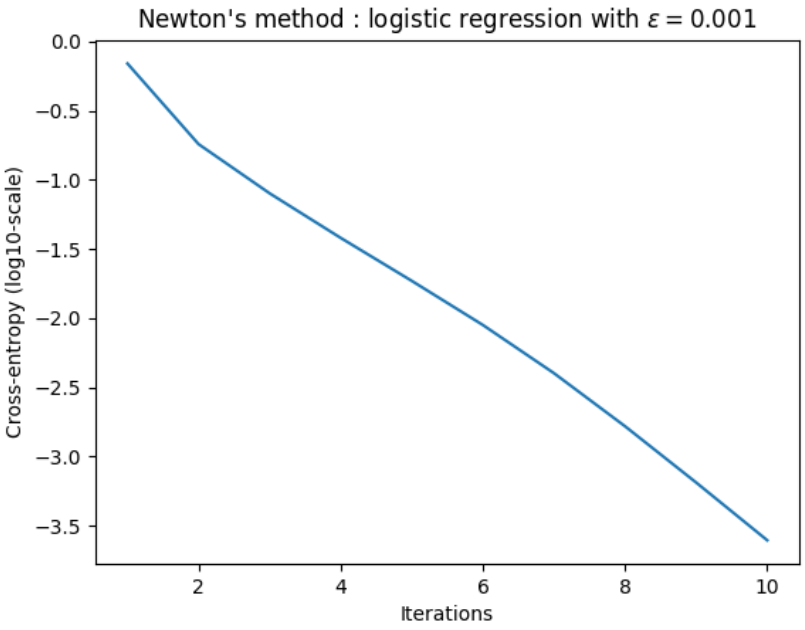

Figure 1: Number of iterations for Newton's method on train set A

2.3 Linear regression

Let's recall the normal equation that solves linear regression for an affine function $f(x) = w^T x + b$. We introduce $\tilde{x}$ similarly as in the last section, and $X = (\tilde{x}_1, ..., \tilde{x}_N)^T$, $Y = (y_1, ..., y_N)^T$.

$$\hat{\theta} = (X^T X)^{-1}(X^T y)$$

A visualisation of the learnt hyperplane is displayed figure 4 on appendix A. Numerical values of learnt parameters are available in appendix B, table 2.

2.4 Application

Table 1 summarizes our results with all the methods, including the QDA model which is described next section. Best test accuracy is highlighted on dataset A, B and C. As expected, misclassification is consistently higher on the test set by a small margin. All methods perform well on dataset A which is linearly separable. QDA and logistic regression perform well on dataset B, but LDA and linear regression outperform them on dataset C. Dataset C is still looking linear but has lots of outliers. Our conclusion is that linear regression and LDA are more robust to outliers and have less capacity than QDA.

		LDA	Log. reg.	Lin. reg.	QDA
Dataset A	train	1.000	1.000	1.000	1.000
	test	**0.990**	**0.990**	**0.990**	**0.990**
Dataset B	train	0.980	0.990	0.980	0.985
	test	0.955	0.965	0.955	**0.970**
Dataset C	train	0.973	0.970	**0.973**	0.970
	test	0.960	0.953	0.960	0.950

Table 1: Performance (accuracy) of all methods

2.5 QDA model

Let $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$ a i.i.d. sample of (x, y). The likelihood of $\mathcal{D}$ has the same form as the LDA model except that now $p_i(x) = \mathcal{N}(x|\mu_i, \Sigma_i)$. The MLE estimators for π, μ_0 and μ_1 remain unchanged but now:

$$\left[\hat{\Sigma}_0\right]_{\text{MLE}} = \tilde{\Sigma}_0 \quad \text{and} \quad \left[\hat{\Sigma}_1\right]_{\text{MLE}} = \tilde{\Sigma}_1 \tag{13}$$

where $\tilde{\Sigma}_0$ and $\tilde{\Sigma}_1$ are defined in subsection 2.1. By using Bayes formula, the probability $p(y = 1|x)$ takes the form:

$$p(y = 1|x) = \sigma\left(x^\top A x + \omega^T x + b\right) \tag{14}$$

where

$$\begin{cases} A &= \frac{1}{2}\left(\Sigma_0^{-1} - \Sigma_1^{-1}\right) \\ \omega &= \Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0 \\ b &= \frac{1}{2}\log\det\left(\Sigma_0\Sigma_1^{-1}\right) - \log\left(\frac{1-\pi}{\pi}\right) + \frac{1}{2}\mu_0^\top\Sigma_0^{-1}\mu_0 - \frac{1}{2}\mu_1^\top\Sigma_1^{-1}\mu_1 \end{cases}$$

Thus, the equation $p(y = 1|x) = 0.5$ is equivalent to $x^\top A x + \omega^T x + b = 0$, which is an ellipse.

We apply QDA on datasets A, B and C and plotted the results in 5 in appendix A. The learnt parameters we found are:

$$\pi^A = 0.48, \quad \pi^B = 0.55, \quad \pi^C = 0.42$$

$$\mu_0^A = \begin{bmatrix} 10.73 \\ 10.9 \end{bmatrix}, \quad \mu_0^B = \begin{bmatrix} 10.58 \\ 11.17 \end{bmatrix}, \quad \mu_0^C = \begin{bmatrix} 10.62 \\ 10.84 \end{bmatrix}$$

$$\mu_1^A = \begin{bmatrix} 11.03 \\ 5.99 \end{bmatrix}, \quad \mu_1^B = \begin{bmatrix} 11.25 \\ 6.10 \end{bmatrix}, \quad \mu_1^C = \begin{bmatrix} 11.18 \\ 6.04 \end{bmatrix}$$

$$\Sigma_0^A = \begin{bmatrix} 0.46 & 0.10 \\ 0.10 & 0.71 \end{bmatrix}, \quad \Sigma_0^B = \begin{bmatrix} 0.76 & 0.05 \\ 0.05 & 1.11 \end{bmatrix}, \quad \Sigma_0^C = \begin{bmatrix} 1.26 & -0.43 \\ -0.43 & 1.83 \end{bmatrix}$$

$$\Sigma_1^A = \begin{bmatrix} 0.72 & 0.18 \\ 0.18 & 0.93 \end{bmatrix}, \quad \Sigma_1^B = \begin{bmatrix} 2.37 & 1.23 \\ 1.23 & 2.84 \end{bmatrix}, \quad \Sigma_1^C = \begin{bmatrix} 1.27 & 0.46 \\ 0.46 & 1.44 \end{bmatrix}$$

A Boundaries Visualizations

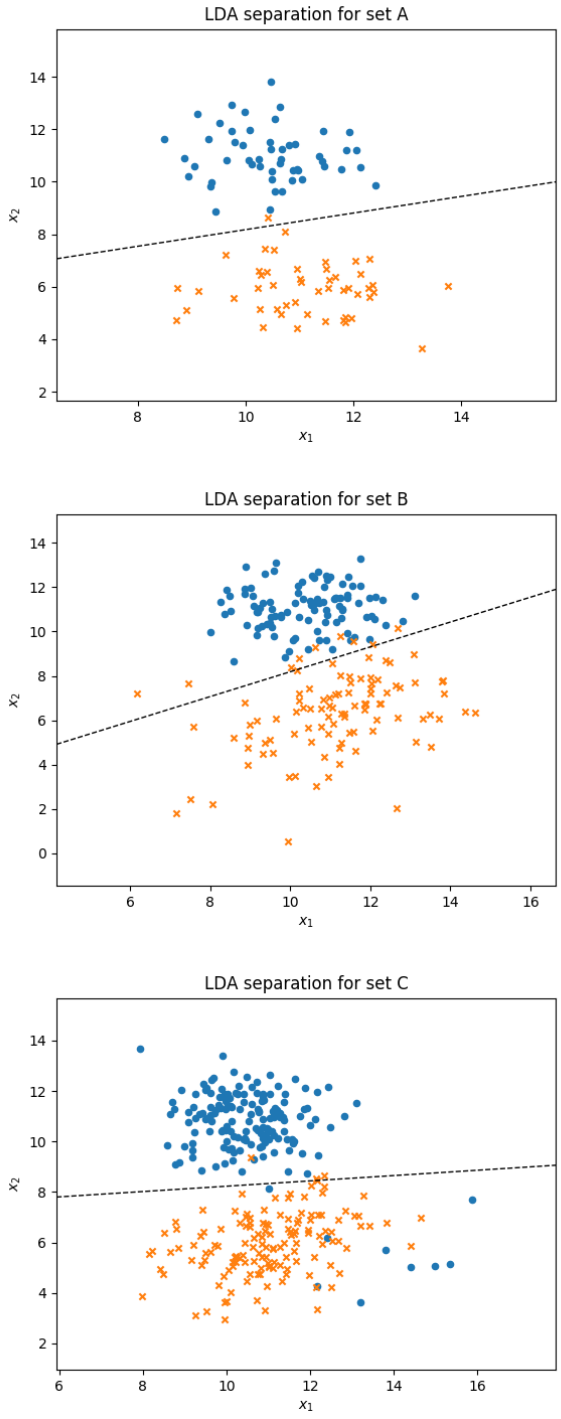

Figure 2: LDA separation for datasets A, B and C

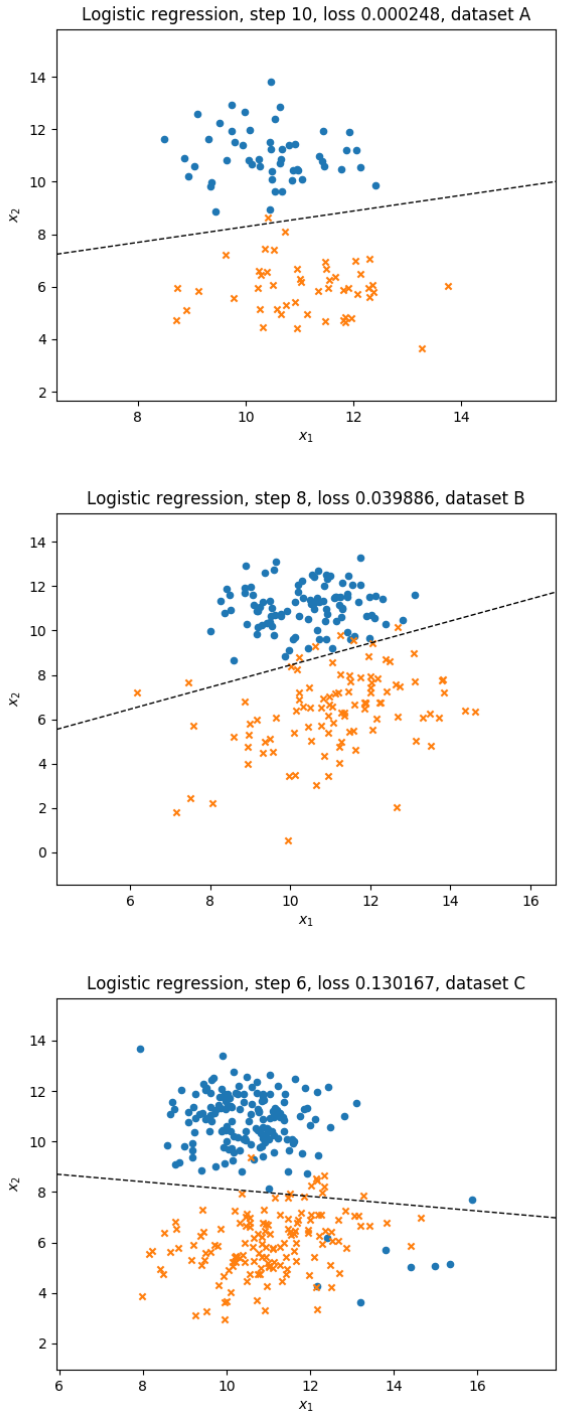

Figure 3: Logistic regression separation for datasets A, B and C

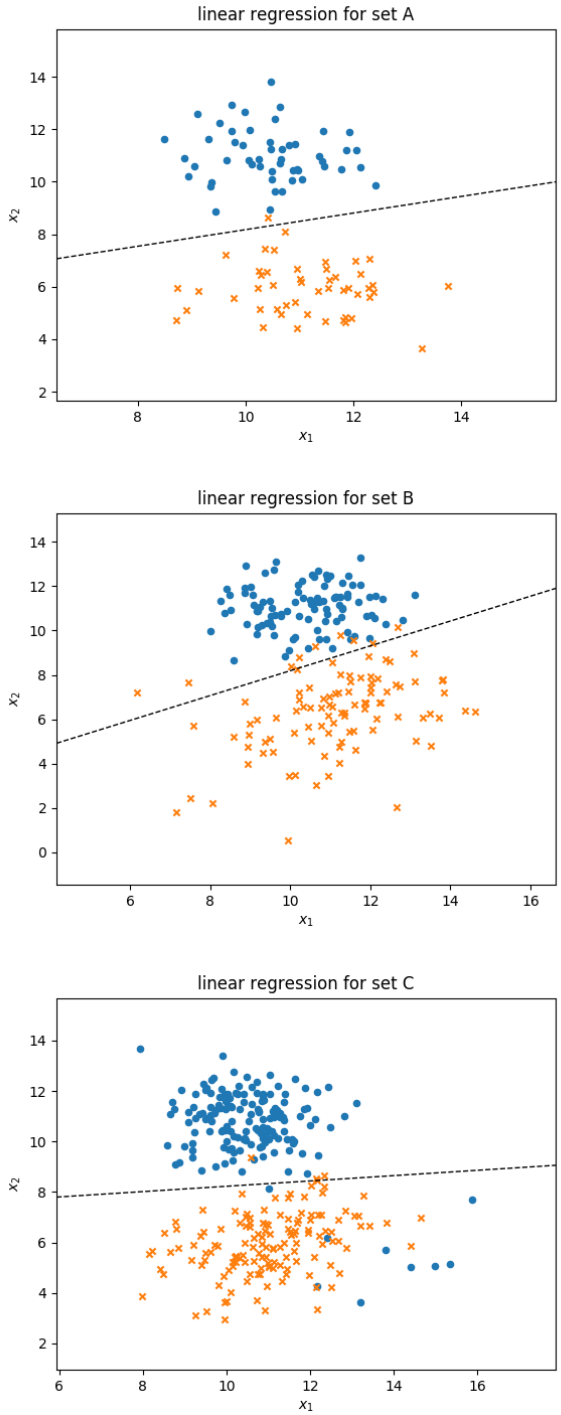

Figure 4: Linear regression separation for datasets A, B and C

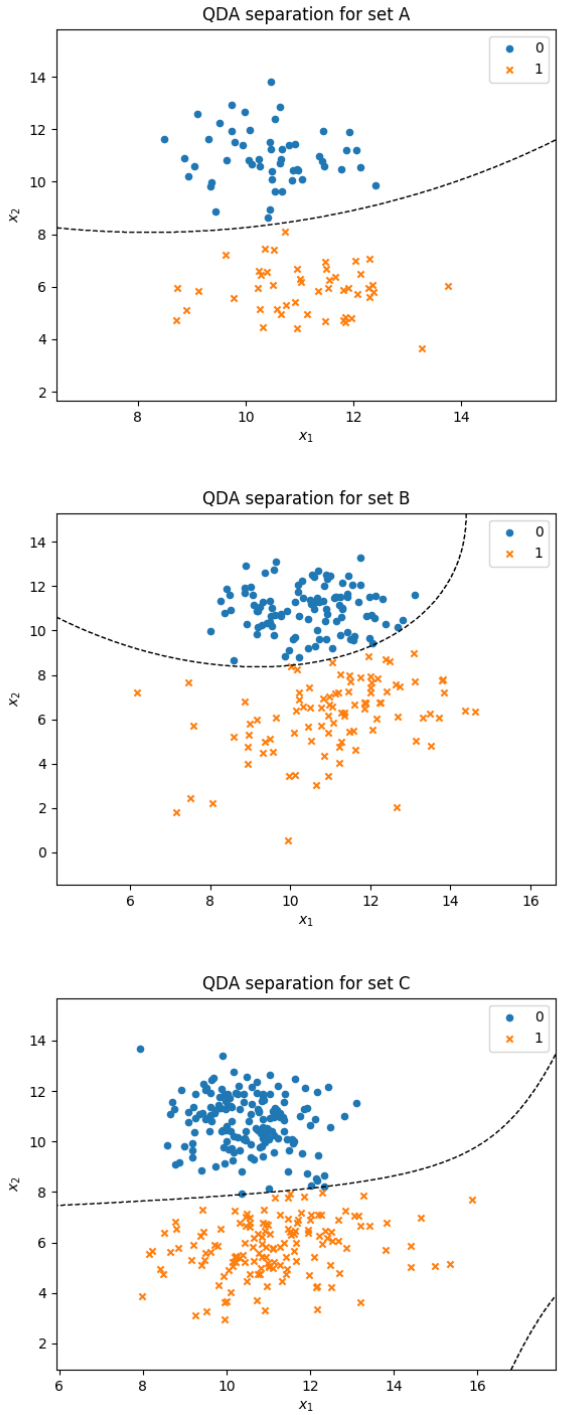

Figure 5: QDA separation for datasets A, B and C

B Learnt parameter values

		LogReg	LinReg
Dataset A	w_1	2.66	0.06
	w_2	-8.90	-0.18
	b	47.67	1.38
Dataset B	w_1	1.84	0.08
	w_2	-3.71	-0.15
	b	13.41	0.88
Dataset C	w_1	-0.28	0.02
	w_2	-1.91	-0.16
	b	18.80	1.64

Table 2: Reported values learnt by logistic regression and linear regression.