HWK 1 - Probabilistic Graphical Models

Alejandro de la Concha (alexdavidhalo@gmail.com)

1 Learning in discrete graphical models

Consider the following model : z and x are discrete variables taking respectively M and K different values with $p(z = m) = \theta_m$ 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 succinctly your derivations and not just the final answer.

We define the dummy variable $\hat{Z}_1, ..., \hat{Z}_n$ such that $P(Z_i = m) = P(\hat{Z}_i^m = 1) = \pi_m$. $\hat{Z}_i \sim M(\pi_1, ..., \pi_M; 1)$ and the dummy variables $\hat{X}_1, ..., \hat{X}_n \in \mathbb{R}^K$ such that $\{X_j = K\} = \{\hat{X}_j^K = 1\}$ and $\hat{X}_j^K | \hat{Z}_i^m = 1 \sim M(\theta_{m1}, ..., \theta m K; 1)$

We can express the likelihood as:

$$L(\pi, \theta) = \prod_{n=1}^{N} P(\pi, \theta) = \prod_{n=1}^{N} \prod_{m=1}^{M} \pi_m^{\hat{Z}_n^m} \prod_{k=1}^{K} \theta_{m,k}^{\hat{Z}_n^m \hat{X}_n^K}$$

Then:

$$l(\pi, \theta) = \sum_{n=1}^{N} \sum_{m=1}^{M} (\hat{Z}_n^k ln(\pi_m) + \sum_{k=1}^{K} \hat{Z}_n^m \hat{x}_n^k ln(\theta_{mk}))$$

We can find the maximum likelihood estimator by solving the following optimization problem.

$$\begin{aligned} \max_{\pi,\theta} \quad & \sum_{n=1}^{N} \sum_{m=1}^{M} (\hat{Z}_n^k ln(\pi_m) + \sum_{k=1}^{K} \hat{Z}_n^m \hat{X}_n^k ln(\theta_{mk})) \\ \text{subject to} \quad & \\ & \sum_{m=1}^{M} \pi_m = 1; \quad \sum_{k=1}^{K} \theta_{mk} = 1 \quad m = 1, ..., M \end{aligned} \tag{1}$$

Problem 1 is a constrained concave optimization problem so it has a solution and as the Slater conditions can be verified, strong duality holds.

The Lagrangian of the problem is:

$$L(\pi,\theta,\lambda_\pi,\lambda_{\theta_1},...,\lambda_{\theta_m}) = \sum_{n=1}^{N}\sum_{m=1}^{M}(\hat{Z}_n^k ln(\pi_m)+\sum_{k=1}^{K}\hat{Z}_n^m\hat{X}_n^k ln(\theta_{mk}))+\lambda_\pi(\sum_{m=1}^{M}\pi_m-1)+\sum_{m=1}^{M}\lambda_{\theta_m}(\sum_{k=1}^{K}\theta_{mk}-1)$$

The gradient equations are:

$$\frac{\partial L}{\partial \pi_k} = \sum_{n=1}^{N}\frac{\hat{Z}_n^k}{\pi_k} + \lambda_\pi$$

$$\frac{\partial L}{\partial \theta_{m,k}} = \sum_{n=1}^{N}\frac{\hat{Z}_n^m\hat{X}_n^k}{\theta_{mk}} + \lambda_{\theta_m}$$

Then:

$$\hat{\pi}_k = \frac{N_k}{N} \quad \hat{\theta}_{mk} = \frac{N_{mk}}{N_m}$$

Where N_k is the number of observations such that $Z = k$, N_{mk} is the number of observations such that $Z = m$ and $X = k$, and N is the number of observations.

2 Linear classification

The files classificationA.train, classificationB.train and classificationC. train 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.

2.1 LDA

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

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

(a) Derive the form of the maximum likelihood estimator for this model. Indication : the model was presented in class but not the MLE computations. Compare $p(y = 1|x)$ with the form of logistic regression.

$$P(y, x) = \pi^y(1-\pi)^{1-y}N(y\mu_1 + (1-y)\mu_0, \Sigma)$$

Let's call $\Lambda = \Sigma^{-1}$, then:

$$
\begin{aligned}
L(\pi, \mu_0, \mu_1, \Lambda) &= \prod_{n=1}^{N} \pi^{y_n}(1-\pi)^{1-y_n}(\frac{1}{(2\pi)^{\frac{d}{2}}|\Lambda|^{\frac{-1}{2}}})exp(-\frac{(x_n - \mu_1 y_n - (1-y_n)\mu_0)^T \Lambda (x_n - \mu_1 y_n - (1-y_n)\mu_0)}{2}) \\
l(\pi, \mu_0, \mu_1, \Lambda) &= \sum_{n=1}^{N}(y_n ln(\pi) + (1-y_n)ln(1-\pi) - \frac{d}{2}ln(2\pi) + \frac{1}{2}log(|\Lambda|) \\
&\quad - \frac{(x_n - \mu_1 y_n - (1-y_n)\mu_0)^T \Lambda (x_n - \mu_1 y_n - (1-y_n)\mu_0)}{2})
\end{aligned}
$$

In order to get the MLE , we obtain the gradient equations:

$$
\begin{aligned}
\frac{\partial L}{\partial \pi} &= \sum_{n=1}^{N} \frac{y_n}{\pi} - \sum_{n=1}^{N} \frac{1-y_n}{1-\pi} \\
\frac{\partial L}{\partial \mu_0} &= -\sum_{n=1}^{N}(1-y_n)\Sigma^{-1}(x_n - \mu_0) \\
\frac{\partial L}{\partial \mu_1} &= -\sum_{n=1}^{N} y_n \Sigma^{-1}(x_n - \mu_1) \\
\Delta_\Lambda L &= -\frac{N}{2}\Lambda^{-1} - \frac{1}{2}\sum_{n=1}^{N}(x_n - \mu_1 y_n - (1-y_n)\mu_0)(x_n - \mu_1 y_n - (1-y_n)\mu_0)^T
\end{aligned}
$$

Then the MLE are:

$$
\begin{aligned}
\hat{\pi} &= \frac{N_1}{N} \\
\hat{\mu}_0 &= \sum_{n=1}^{N} \frac{(1-y_n)x_n}{N} \\
\hat{\mu}_1 &= \sum_{n=1}^{N} \frac{y_n x_n}{N} \\
\hat{\Sigma} &= \frac{\sum_{n=1}^{N}(x_n - \mu_1 y_n - (1-y_n)\mu_0)(x_n - \mu_1 y_n - (1-y_n)\mu_0)^T}{N} \\
&= \frac{\sum_{n=1}^{N} y_n(x_n - \mu_1)(x_n - \mu_1)^T + (1-y_n)(x_n - \mu_0)(x_n - \mu_0)^T}{N}
\end{aligned}
$$

(b) 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:

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

$$
\begin{aligned}
P(y=1|x) &= \frac{\pi N(\mu_1,\Sigma)}{(1-\pi)N(\mu_0,\Sigma)+\pi N(\mu_1,\Sigma)} \\
&= \frac{\frac{\pi}{1-\pi}\frac{N(\mu_1,\Sigma)}{N(\mu_0,\Sigma)}}{1+\frac{\pi}{1-\pi}\frac{N(\mu_1,\Sigma)}{N(\mu_0,\Sigma)}} \\
\frac{\pi}{1-\pi}\frac{N(\mu_1,\Sigma)}{N(\mu_0,\Sigma)} &= exp(-\frac{(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)}{2}+\frac{(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)}{2}) \\
&= exp(x^T\Sigma^{-1}(\mu_1-\mu_0)-\frac{1}{2}(\mu_1-\mu_0)\Sigma^{-1}(\mu_1+\mu_0)+log(\frac{\pi}{1-\pi}))
\end{aligned}
$$

The if we call $\beta_0 = \frac{1}{2}(\mu_1-\mu_0)\Sigma^{-1}(\mu_1+\mu_0)+log(\frac{\pi}{1-\pi})$ and $\beta = \Sigma^{-1}(\mu_1-\mu_0)$.

This means that :

$$P(y=1|x) = \sigma(\beta_0 + X^T\beta)$$

In conclusion, both the logistic regression and LDA estimates $P(y=1|x)$ by using the logistic function conjugated with an affine function in the covariates.

2.2 QDA

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.

$$y \sim Bernoulli(\pi)///\text{and} x|y=i \sim Normale(\mu_i,\pi_i)$$

Implement the maximum likelihood estimator and apply it to the data. (a) Derive the form of the maximum likelihood estimator for this model.

$$P(y,x) = \pi^y(1-\pi)^{1-y}N(\mu_0,\Sigma_0)^{(1-y)}N(\mu_1,\Sigma_1)^y$$

$$
\begin{aligned}
L(\pi,\mu_0,\mu_1,\Lambda) &= \prod_{n=1}^{N}\pi^{y_n}(1-\pi)^{1-y_n}(\frac{y_n}{(2\pi)^{\frac{d}{2}}|\Lambda_1|^{\frac{-1}{2}}})(\frac{1-y_n}{(2\pi)^{\frac{d}{2}}|\Lambda_0|^{\frac{-1}{2}}}) \\
&\quad exp(-\frac{y_n(x_n-\mu_1)^T\Lambda(x_n-\mu_1)+(1-y_n)(x_n-\mu_0)^T\Lambda(x_n-\mu_0)}{2}) \\
l(\pi,\mu_0,\mu_1,\Lambda) &= \sum_{n=1}^{N}(y_n ln(\pi)+(1-y_n)ln(1-\pi)-\frac{d}{2}ln(2\pi)+\frac{y_n}{2}log(|\Lambda_1|)+\frac{1-y_n}{2}log(|\Lambda_0|) \\
&\quad -\frac{y_n(x_n-\mu_1)^T\Lambda_1(x_n-\mu_1)}{2}-\frac{(1-y_n)(x_n-\mu_0)^T\Lambda_0(x_n-\mu_0)}{2})
\end{aligned}
$$

For $\hat{\pi}$,$\hat{\mu}_1$ and $\hat{\mu}_0$ we kept the same estimators as in the LDA. For $\hat{\Lambda}_0$ and $L\hat{ambda}_1$.

$$\begin{aligned}
\Delta_{\Lambda_0} l &= \Delta_{\Lambda_0}(-\sum_{n=1}^{N} \frac{(1-y_n)}{2} log\Lambda_0 - \frac{1}{2}trace(\Lambda_0 \sum_{n=1}^{N}(1-y_n)(x_n-\mu_0)(x_n-\mu_0)T) \\
&= -\sum_{n=1}^{N} \frac{1-y_n}{2}{\Lambda_0}^{-1} - \frac{1}{2}\sum_{n=1}^{N}(1-y_n)(x_n-\mu_0)(x_n-\mu_0)^T \\
\Delta_{\Lambda_1} l &= -\sum_{n=1}^{N} \frac{y_n}{2}{\Lambda_1}^{-1} - \frac{1}{2}\sum_{n=1}^{N}(y_n)(x_n-\mu_1)(x_n-\mu_1)^T
\end{aligned}$$

Then the MLE estimators are:

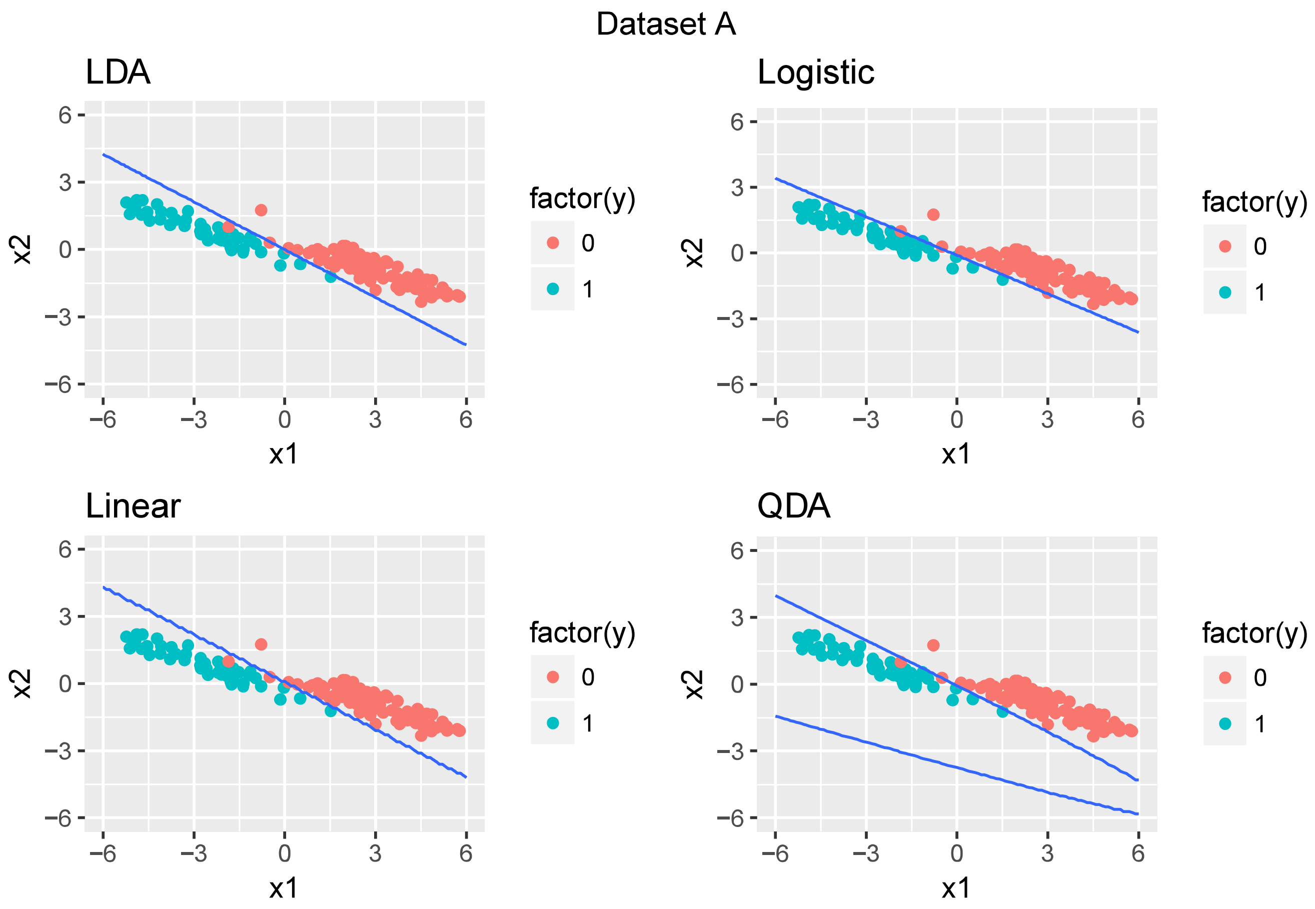
$$\begin{aligned}
\hat{\Sigma}_1 &= \frac{\sum_{n=1}^{N} y_n(x_n-\mu_1)(x_n-\mu_1)^T}{\sum_{n=1}^{N} y_n} \\
\hat{\Sigma}_0 &= \frac{\sum_{n=1}^{N}(1-y_n)(x_n-\mu_0)(x_n-\mu_0)^T}{\sum_{n=1}^{N}(1-y_n)}
\end{aligned}$$

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

$$\begin{aligned}
P(y=1|x) &= \frac{\pi}{1-\pi}\frac{N(\mu_1,\Sigma_1)}{N(\mu_0,\Sigma_0)} \\
&= exp(-\frac{(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}{2} + \frac{(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)}{2} + log(\frac{\pi}{1-\pi}) - \frac{1}{2}log(\frac{|\Sigma_1|}{|\Sigma_0|}) \\
&= exp(\frac{X^T(\Sigma_0^{-1}-\Sigma_1^{-1})X}{2} + X^T(\Sigma^{-1}\mu_1 - \Sigma_0^{-1}\mu_0) + \frac{\mu_0^T\Sigma_0^{-1}\mu_0}{2} \\
&- \frac{\mu_1^T\Sigma_1^{-1}\mu_1}{2} + log(\frac{\pi}{1-\pi}) - \frac{log(|\Sigma_1|)}{2} + \frac{log(|\Sigma_0|)}{2}
\end{aligned}$$

If we make $P = (\Sigma_0^{-1} - \Sigma_1^{-1}$, $\beta = \Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0$ and $\beta_0 = \frac{\mu_0^T\Sigma_0^{-1}\mu_0}{2} - \frac{\mu_1^T\Sigma_1^{-1}\mu_1}{2} + log(\frac{\pi}{1-\pi}) - \frac{log(|\Sigma_1|)}{2} + \frac{log(|\Sigma_0|)}{2}$

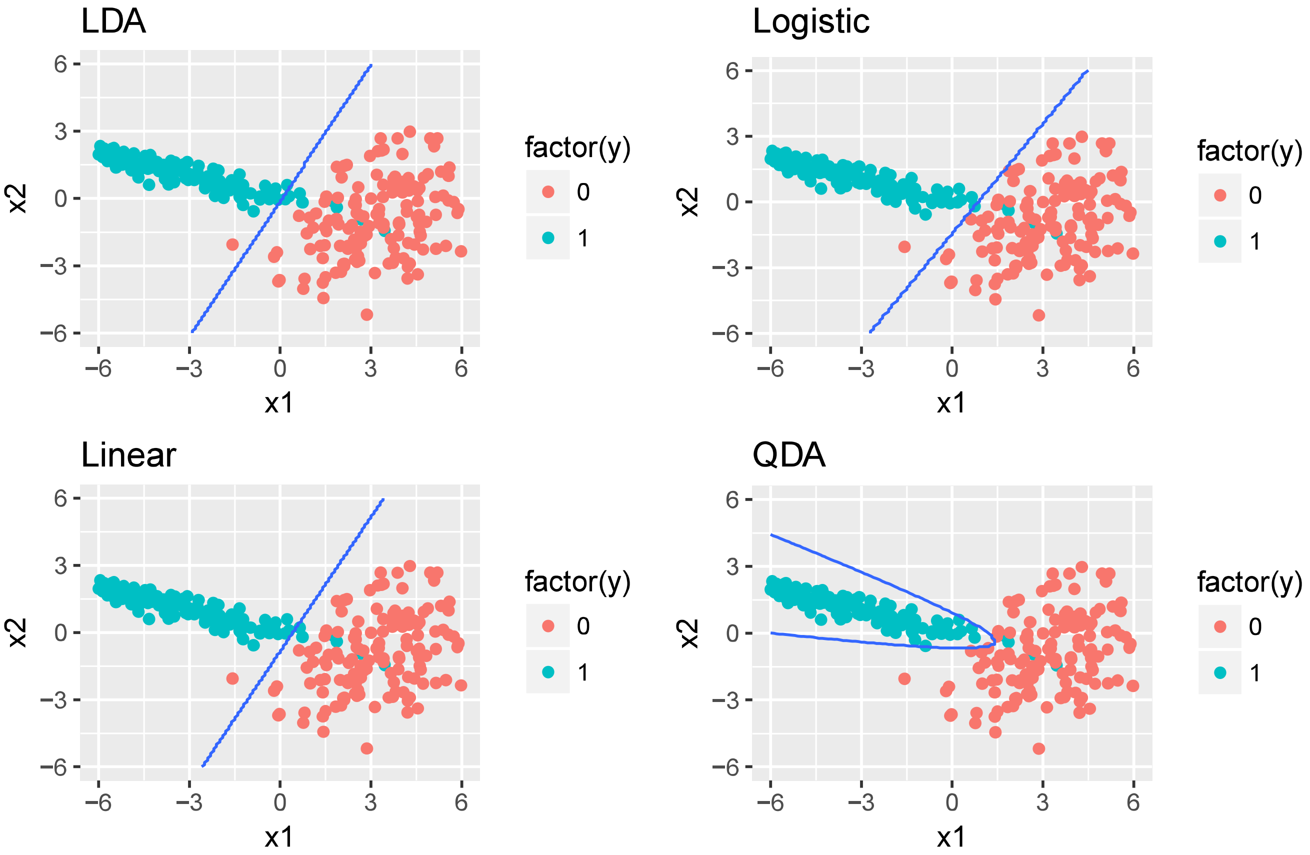
So QDA also has a connection with the logistic regression, but, QDA estimate $P(Y=1|X)$ with a quadratic function.



	train	test
LDA	1.33	2
LINEAR	1.33	2.13
LOGISTIC	0	3.53
QDA	0.67	2

In the train set A, the classes are separable so all algortihms performed well. The plot also suggests that both classes have the same covariance matrix, which could explain why LDA made the most accurate predictions. By definition of the objective function, Newton's algorithm for logistic regression won't converge in a finite number of steps when the classes are separable since the proabiblities of belonging to one of the classes is 0 or 1, then the optimum of the logistic function is attained when the norm of the parameters goes to infinity. I stopped the Newton's algorithm until I got 0 error in the train set. We can see from the train and test error that this model overfits a little. QDA also showed a good performance, but it has the problem of having too many parameters compared with the other algorithms.

Dataset B


LDA

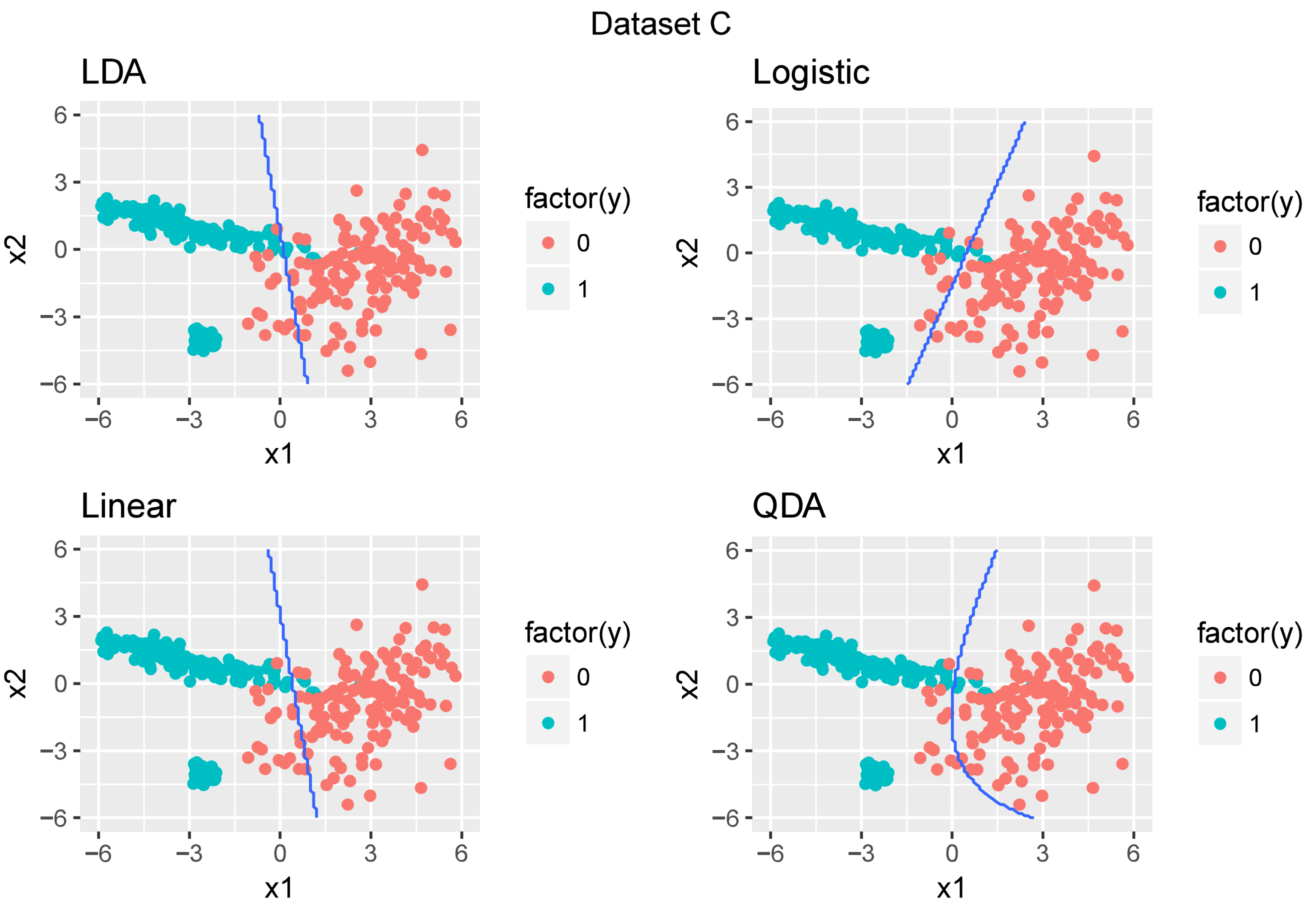
Logistic

Linear

QDA

	train	test
LDA	3	4.15
LINEAR	3.33	4.2
LOGISTIC	2	4.3
QDA	1.33	2

The train set seems to satisfy the QDA assumptions: two clases that show different means and levels of dispertion. Probably, that is the reason why QDA performs the best in this case. For the linear models, results were not satisfactory. QDA also suggests that we might get better results in the regressions by using a quadratric transformation of the covariates.



	train	test
LDA	5.5	4.23
LINEAR	6.25	4.6
LOGISTIC	4	2.27
QDA	5.25	3.83

The dataset C shows a similar pattern that dataset B with the exception of an isolated cluster of points. This feature made that all models, except for logistic regression, had a worse performance. Both QDA and LDA models rely on the assumption of normality and in linear regression we model y as a continuos variable, this mismatch between the hypothesis and the dataset can be the cause of their inferiority compared to logistic regression. It is interesting to note how the test error is smaller than the train error, which is counterintuitive, but it is a sign models didn't overfit