1. Learning in discrete graphical models

Let $(\mathbf{x}_i, \mathbf{z}_i)_{1,\ldots,n}$ be an i.i.d sample of observations. $p(z)$ can be written as $p(\mathbf{z}_i|\boldsymbol{\pi}) = \prod_{m=1}^{M} \pi_m^{z_{im}}$ and $p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\Theta}) = \prod_{m,k} (\theta_{mk})^{z_{im}x_{ik}}$ The log-likelihood of the observations as a function of $\boldsymbol{\pi}$ and $\boldsymbol{\Theta} = (\theta_{mk})$:

$$\ell(\boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{i=1}^{n} \log p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{i=1}^{n} \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\Theta}) + \sum_{i=1}^{n} \log p(\mathbf{z}_i|\boldsymbol{\pi})$$

The maximum likelihood estimator of the right term is given by $\hat{\boldsymbol{\pi}} = (\frac{n_1}{n}, \frac{n_0}{n}, ..., \frac{n_M}{n})$ (multinomial distribution and the two terms depend on different parameters). For the left term, with $\mathcal{A}_m = \{i|z_{im} = 1\}$ the indexes of the pairs with $z = m$, $n_m = |\mathcal{A}_m|$ and n_{mk} the number of pairs $(x = k, z = m)$:

$$\sum \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\Theta}) = \sum_i \sum_{k,m} z_{im} x_{ik} \log \theta_{mk} = \sum_{m=1}^{M} \left(\sum_{i \in \mathcal{A}_m} \sum_{k=1}^{K} x_{ik} \log \theta_{mk} \right) = \sum_{m=1}^{M} \sum_{k=1}^{K} n_{mk} \log \theta_{mk}$$

By analogy with the multinomial distribution maximum likelihood estimator, each term in the sum over m is maximized by $\hat{\boldsymbol{\theta}}_m = \left(\frac{n_{mk}}{|\mathcal{A}_m|}\right)_k$, hence $\boxed{\hat{\boldsymbol{\Theta}} = \left(\frac{n_{mk}}{n_m}\right)_{km}}$ and $\boxed{\hat{\boldsymbol{\pi}} = \left(\frac{n_m}{n}\right)_m}$

2. Linear classification - Generative model (LDA) (2.1 (a))

Let $(\mathbf{x}_i, y_i)_{1,\ldots,n}$ be an i.i.d sample of observations.

$$p(\mathbf{x}_i, y_i|\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\mathbf{x}_i|y_i) \cdot p(y_i) = \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})^{y_i} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma})^{1-y_i} \cdot \pi^{y_i}(1-\pi)^{1-y_i}$$

The log-likelihood as a function of all parameters is

$$\ell(\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}) = \sum_i (y_i \log \pi + (1-y_i)\log(1-\pi)) + \sum_i y_i \log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + \sum_i (1-y_i) \log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$$

The first sum depends only on π and is maximized by $\boxed{\hat{\pi} = n_1/n}$. Since we have for the terms depending on $\boldsymbol{\mu}_1$

$$\sum_i y_i \log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -1/2 \sum_i y_i (\mathbf{x}_i - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_1) + const.$$

By setting the derivatives with respect to $\boldsymbol{\mu}_1$ to 0 and by symmetry for $\boldsymbol{\mu}_0$: $\boxed{\hat{\boldsymbol{\mu}}_1 = (1/n_1) \sum y_i \mathbf{x}_i}$ and $\boxed{\hat{\boldsymbol{\mu}}_0 = (1/n_0) \sum (1-y_i) \mathbf{x}_i}$. The terms depending on $\boldsymbol{\Sigma}$ are

$$-\frac{1}{2} \sum_i y_i \left(\log|\boldsymbol{\Sigma}| + (\mathbf{x}_i - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_1) \right) - \frac{1}{2} \sum_i (1-y_i) \left(\log|\boldsymbol{\Sigma}| + (\mathbf{x}_i - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_0) \right) =$$

$$-\frac{n}{2} \log|\boldsymbol{\Sigma}| - \frac{n}{2} \mathrm{tr}(\boldsymbol{\Sigma}^{-1} \mathbf{S})$$

Where $\mathbf{S} = \frac{n_1}{n} \left(\frac{1}{n_1} \sum_{y_i=1} (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^T \right) + \frac{n_0}{n} \left(\frac{1}{n_0} \sum_{y_i=0} (\mathbf{x}_i - \boldsymbol{\mu}_0)(\mathbf{x}_i - \boldsymbol{\mu}_0)^T \right)$ convex sum of the covariances in each class. We can then find that maximum likelihood estimator for $\boldsymbol{\Sigma}$ is $\boxed{\hat{\boldsymbol{\Sigma}} = \mathbf{S}}$ by setting the derivatives to 0 in the above expression. We now study $p(y|\mathbf{x}) \propto p(\mathbf{x}|y)p(y)$

$$p(y|\mathbf{x}) \propto \pi^y (1-\pi)^{1-y} \mathcal{N}(\mathbf{x}|\mu_1, \boldsymbol{\Sigma})^y \mathcal{N}(\mathbf{x}|\mu_0, \boldsymbol{\Sigma})^{1-y} \propto \exp\left(y \left(\log \frac{\pi}{1-\pi} + \frac{1}{2}(\mu_0^T \boldsymbol{\Sigma}^{-1} \mu_0 - \mu_1^T \boldsymbol{\Sigma}^{-1} \mu_1) \right) + y(\mu_1 - \mu_0)^T \boldsymbol{\Sigma}^{-1} X \right)$$

Thus, $p(y = k|\mathbf{x}) = \dfrac{\exp(ya + y\mathbf{b}^T\mathbf{x})}{1 + \exp(a + \mathbf{b}^T\mathbf{x})}$ and $\boxed{p(y = 1|\mathbf{x}) = \sigma(a + \mathbf{b}^T\mathbf{x})}$ with a and $\mathbf{b}$ defined in the above formula. It is therefore equivalent to a logistic regression with a closed formula for the coefficients based on the empirical means, covariances and class repartition of the data.

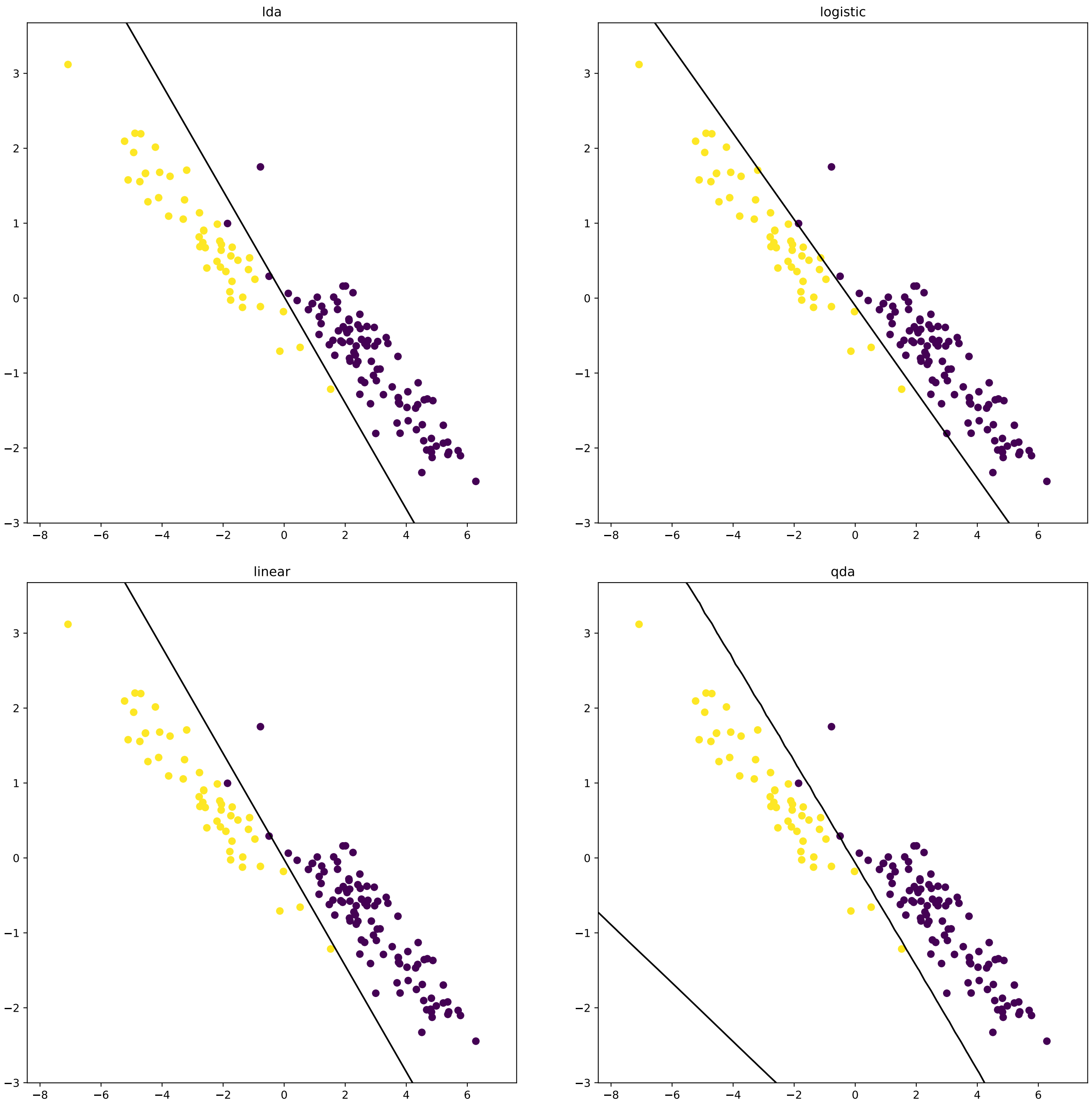
3. Linear classification - QDA model (2.5 (a))

Results are the same as above for $\hat{\pi}$, $\hat{\boldsymbol{\mu}}_1$, $\hat{\boldsymbol{\mu}}_0$ but $\hat{\boldsymbol{\Sigma}}_1$ and $\hat{\boldsymbol{\Sigma}}_0$ must be computed separately. The term of the log-likelihood that depends on $\boldsymbol{\Sigma}_1$ is

$$-\frac{1}{2} \sum_i y_i \left(\log|\boldsymbol{\Sigma}_1| + (\mathbf{x}_i - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_1) \right) = -\frac{n_1}{2} \log|\boldsymbol{\Sigma}_1| - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}_1^{-1} \sum_{i|y_i=1} (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^T \right)$$

Hence $\boxed{\hat{\boldsymbol{\Sigma}}_1 = \frac{1}{n_1} \sum_{i|y_i=1} (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^T}$ and $\boxed{\hat{\boldsymbol{\Sigma}}_0 = \frac{1}{n_0} \sum_{i|y_i=0} (\mathbf{x}_i - \boldsymbol{\mu}_0)(\mathbf{x}_i - \boldsymbol{\mu}_0)^T}$.

We also have $\boxed{p(y = 1|\mathbf{x}) = \sigma\left(a + \mathbf{b}^T\mathbf{x} + \mathbf{x}^T\mathbf{C}\mathbf{x}\right)}$ with $a = \log \frac{\pi}{1-\pi} + \frac{1}{2}(\mu_0^T \boldsymbol{\Sigma}_{\mathbf{0}}^{-1} \mu_0 - \mu_1^T \boldsymbol{\Sigma}_{\mathbf{1}}^{-1} \mu_1)$, $\mathbf{b} = \mu_1^T \boldsymbol{\Sigma}_{\mathbf{1}}^{-1} - \mu_0^T \boldsymbol{\Sigma}_{\mathbf{0}}^{-1}$ and $\mathbf{C} = \boldsymbol{\Sigma}_{\mathbf{0}}^{-1} - \boldsymbol{\Sigma}_{\mathbf{1}}^{-1}$. The model can be interpreted as a logistic regression in the space of the features $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1\mathbf{x}_2, \mathbf{x}_1^2, \mathbf{x}_2^2)$ with closed formula for the coefficients.

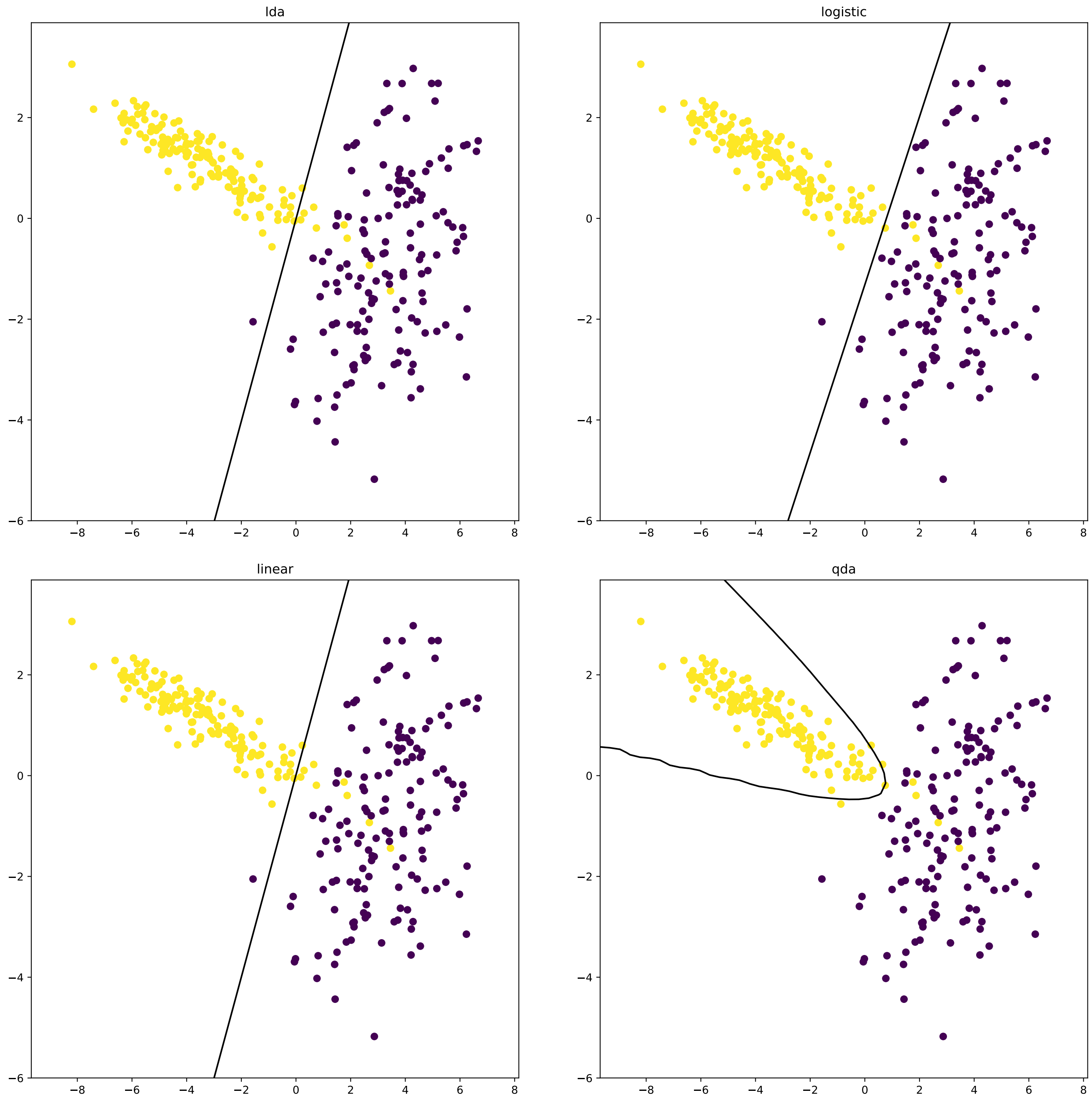


Method	LDA	Logistic	Linear	QDA
Train Error	0.0134	0.0000	0.0134	0.0067
Test Error	0.0207	0.0354	0.0207	0.0193

Linear Regression and LDA give very similar results, with very close decision boundaries.

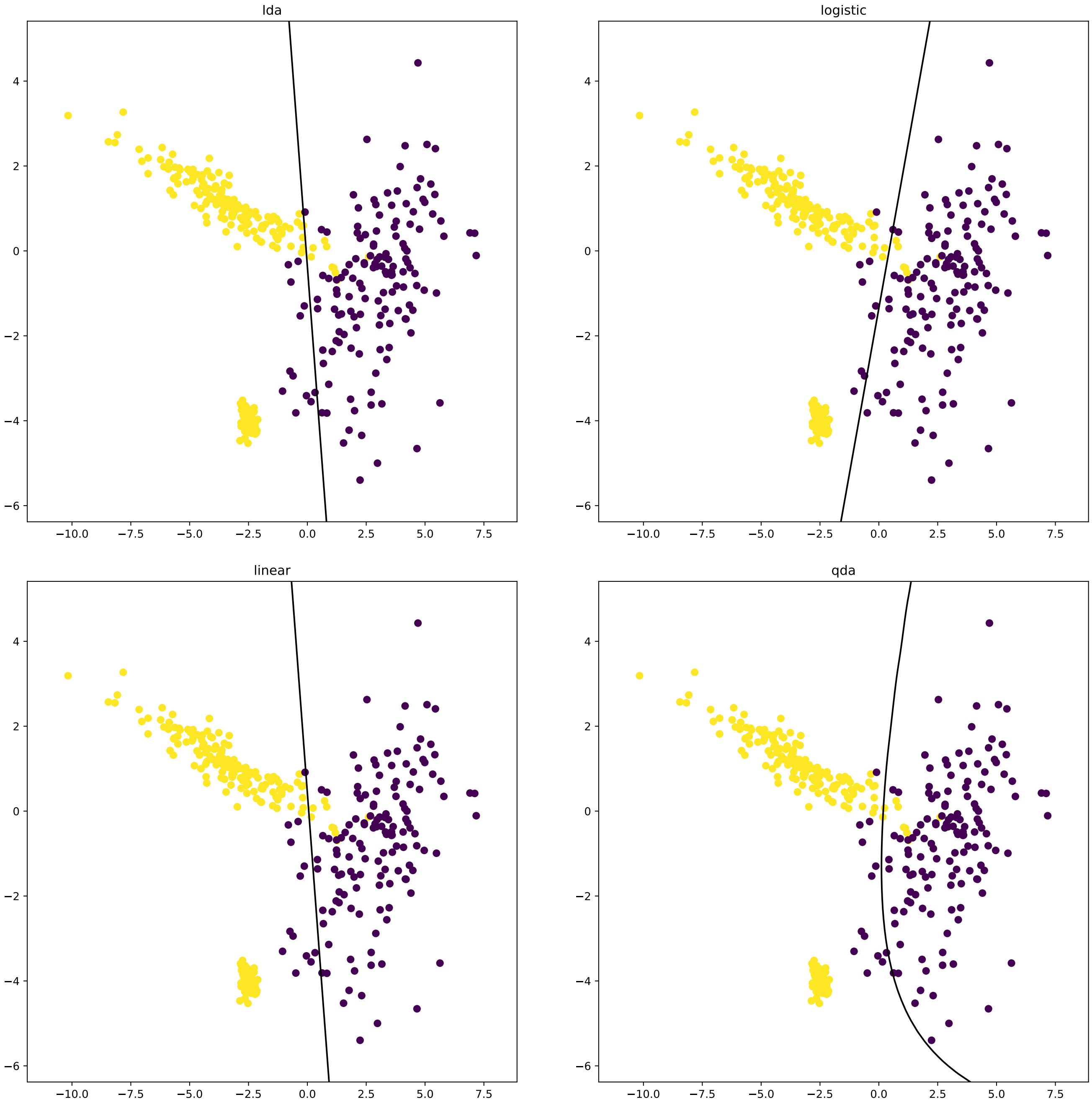
The QDA outperforms the other three methods in terms of classification error on the test set.

Since the data is linearly separable, the logistic regression is able to find a separating hyperplane, achieving a 100% classification score on the train set at the expense of the test error that is higher than for all other methods.



The results for linear regression and LDA are again very close. Their decision boundaries don't seem very good compared to the others. For the LDA this is in part due to the fact that the two sets have very different covariances, and assuming they are equal harms the performance of the classification. For the linear regression, the points farther to the left of the yellow set have a much higher least square penalty (linear regression penalizes values far from the objective either above or below) than the points of the blue set, leading to a shift of the decision boundary to the left.
Logistic regression performs better than LDA and linear regression but still is limited by the linear nature of the boundary it assumes.
The QDA model performs very well because it can cope with the geometric differences between the two sets by letting the covariances being different in the model.

Method	LDA	Logistic	Linear	QDA
Train Error	0.0301	0.0201	0.0301	0.0234
Test Error	0.0415	0.0430	0.0415	0.0235



Method	LDA	Logistic	Linear	QDA
Train Error	0.0551	0.0401	0.0551	0.0526
Test Error	0.0420	0.0227	0.0423	0.0403

LDA and logistic regression are affected by the variance and mean shift that the blob of data introduces. The least square penalty moves the frontier such that the blob would be classified closer to 1. The LDA assumes the data is centered in between the two yellow blobs because it cannot deal with separate sets, which affects its performance negatively.
The logistic regression give better results and is able to handle the blob of data because it is robust to a large number of data points localized in a set and tries to optimize an error function which depends on the quality of classification.
The QDA isn't the most effective model for this data distribution. It is able to handle non linear boundaries that are useful for this kind of complex dataset, but still assumes that data within a class has a Gaussian distribution, which is not the case here.