Ioana-Sabina STOIAN
ioana-sabina.stoian@ens-paris-saclay.fr

Homework# 1
Probabilistic Graphical Models
Master MVA 2019

11/22/19

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.

We consider a number of N i.i.d samples: $(z^{(1)}, x^{(1)})$, $(z^{(2)}, x^{(2)})$, ..., $(z^{(N)}, x^{(N)})$.
We consider the following notations:

$$c_m^{(n)} = \begin{cases} 1 & \text{if } z^{(n)} = m \\ 0 & \text{if } z^{(n)} \neq m \end{cases}$$

$$d_k^{(n)} = \begin{cases} 1 & \text{if } x^{(n)} = k \\ 0 & \text{if } x^{(n)} \neq k \end{cases}$$

Where $z^{(n)}$ and $x^{(n)}$ are the n'th observations from the data set.

$$p(z^{(n)}) = \prod_{m=1}^{M} \pi_m^{1_{\{z^{(n)}=m\}}} \tag{1}$$

$$p(x^{(n)}|z^{(n)}) = \prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{mk}^{1_{\{z^{(n)}=m\}} \cdot 1_{\{x^{(n)}=k\}}} \tag{2}$$

Thus, by using Eq. (1) and Eq. (2) and the considered notations, we can compute the joint probability that is described in Eq. (3).

$$\begin{aligned} p((x^{(1)}, z^{(1)}), (x^{(2)}, z^{(2)}), ..., (x^{(N)}, z^{(N)})) &\overset{i.i.d}{=} \prod_{n=1}^{N} p(x^{(i)}|z^{(i)}) \cdot p(z^{(i)}) \\ &= \prod_{n=1}^{N} ((\prod_{m=1}^{M} \pi_m^{c_m^{(n)}}) \cdot (\prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{mk}^{c_m^{(n)} \cdot d_k^{(n)}})) \end{aligned} \tag{3}$$

The log-likelihood is presented in Eq. (4).

$$\begin{aligned} l(\pi, \theta) = \log L(\pi, \theta) &= \log[(\prod_{n=1}^{N} ((\prod_{m=1}^{M} \pi_m^{c_m^{(n)}}) \cdot (\prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{mk}^{c_m^{(n)} \cdot d_k^{(n)}}))] \\ &= (\sum_{n=1}^{N} \sum_{m=1}^{M} c_m^{(n)} \log \pi_m) + (\sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} c_m^{(n)} \cdot d_k^{(n)} \cdot \log \theta_{mk}) \end{aligned} \tag{4}$$

We need to find the parameters π and θ that maximize the log-likelihood $l(\pi, \theta)$. Because only the first term of Eq. (4) depends on π and only the second term depends on θ, we can find the parameters that maximize each term separately. Thus, we need to find π that maximizes $(\sum_{n=1}^{N} \sum_{m=1}^{M} c_m^{(n)} \log \pi_m)$ and we need to find θ that maximizes $(\sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} c_m^{(n)} \cdot d_k^{(n)} \log \theta_{mk})$.

Maximum likelihood estimator for π

We need to find π that maximizes $\sum_{n=1}^{N} \sum_{m=1}^{M} c_m^{(n)} \log \pi_m$.

$$f(\pi) = \sum_{n=1}^{N} \sum_{m=1}^{M} c_m^{(n)} \log \pi_m = \sum_{m=1}^{M} (\log \pi_m (\sum_{n=1}^{N} c_m^{(n)})) = \sum_{m=1}^{M} n_m \log \pi_m,$$

where $n_m = \sum_{n=1}^{N} c_m^{(n)}$, the number of samples z which are equal to m, under the constraint $\sum_{m=1}^{M} \pi_m = 1$.

As we saw in lecture one, the value for π that maximizes function f under the constraint $\sum_{m=1}^{M} \pi_m = 1$ is:

$$\hat{\pi}_m = \frac{n_m}{N} \tag{5}$$

where $n_m = \sum_{n=1}^{N} c_m^{(n)}$, for all $m = 1, ..., M$.

Maximum likelihood estimator for θ

We need to find θ that maximizes $\sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} c_m^{(n)} \cdot d_k^{(n)} \log \theta_{mk}$ under the constraint $\sum_{k=1}^{K} \theta_{mk} = 1$.

$$g(\theta) = \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} c_m^{(n)} \cdot d_k^{(n)} \log \theta_{mk} = \sum_{m=1}^{M} \sum_{k=1}^{K} \log \theta_{mk} (\sum_{n=1}^{N} c_m^{(n)} \cdot d_k^{(n)}) = \sum_{m=1}^{M} \sum_{k=1}^{K} n_{mk} \log \theta_{mk},$$

where $n_{mk} = \sum_{n=1}^{N} c_m^{(n)} \cdot d_k^{(n)}$, the number of samples for which $x^{(n)} = k$ and $z^{(n)} = m$.
Maximizing $g(\theta)$ under the constraint $\sum_{k=1}^{K} \theta_{mk} = 1$ is equivalent to minimizing
$h(\theta) = -g(\theta) = -\sum_{m=1}^{M} \sum_{k=1}^{K} n_{mk} \log \theta_{mk}$, subject to the constraint $\sum_{k=1}^{K} \theta_{mk} = 1$.

The Lagrangian of this problem is:

$$\mathcal{L}(\theta, \lambda) = -\sum_{m=1}^{M} \sum_{k=1}^{K} n_{mk} \log \theta_{mk} + \lambda (\sum_{k=1}^{K} \theta_{mk} - 1) \tag{6}$$

By computing the derivative of $\mathcal{L}(\theta, \lambda)$ from Eq. (6) with respect to θ_{mk} and by setting it to be equal to 0, we obtain the following result:

$$\frac{\partial \mathcal{L}(\theta, \lambda)}{\partial \theta_{mk}} = -\frac{n_{mk}}{\theta_{mk}} + \lambda = 0$$

Therefore,

$$\theta_{mk} = \frac{n_{mk}}{\lambda} \tag{7}$$

In order to determine the Lagrange multiplier λ, we substitute the result obtained in Eq. (7) into the constraint:

$$\sum_{k=1}^{K} \theta_{mk} - 1 = 0 \implies \sum_{k=1}^{K} \frac{n_{mk}}{\lambda} = 1 \implies \lambda = \sum_{k=1}^{K} n_{mk}$$

Thus,

$$\hat{\theta}_{mk} = \frac{n_{mk}}{\sum_{k'=1}^{K} n_{mk'}} \tag{8}$$

*for all $m = 1, ..., M$ and $k = 1, ..., K$.

2 Linear classification

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:

$$y \sim Bernoulli(\pi)$$

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

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

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

$$p(x|y=1) = \frac{1}{(2\pi)^{\frac{d}{2}} \cdot |\Sigma|^{\frac{1}{2}}} \cdot e^{\frac{-1}{2} \cdot (x-\mu_1)^T \Sigma^{-1} (x-\mu_1)}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{\frac{d}{2}} \cdot |\Sigma|^{\frac{1}{2}}} \cdot e^{\frac{-1}{2}\cdot(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)}$$

Because $x \in \mathbb{R}^2 \rightarrow d = 2$.

We can write $p(x|y)$ more concisely:

$$p(x|y) = \frac{1}{(2\pi) \cdot |\Sigma|^{\frac{1}{2}}} \cdot (e^{\frac{-1}{2}\cdot(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)})^{(1-y)} \cdot (e^{\frac{-1}{2}\cdot(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)})^{y}$$

We consider N i.i.d samples for (x, y): $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(N)}, y^{(N)})$.

$$p((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(N)}, y^{(N)})) \overset{i.i.d}{=} \prod_{i=1}^{N} p(x^{(i)}|y^{(i)})p(y^{(i)})$$

Therefore, the log-likelihood is:

$$\begin{aligned} l_{(x^{(1)},y^{(1)}),...,(x^{(N)},y^{(N)})}(\pi, \mu_0, \mu_1) = &\sum_{i=1}^{N} y^{(i)} \log\pi + (N - \sum_{i=1}^{N} y^{(i)}) \log(1-\pi) - \frac{N}{2} \cdot \log|\Sigma| - N \cdot \log(2\pi) - \\ &- \frac{1}{2}\sum_{i=1}^{N}(1-y^{(i)}) \cdot (x^{(i)}-\mu_0)^T\Sigma^{-1}(x^{(i)}-\mu_0) - \\ &- \frac{1}{2}\sum_{i=1}^{N} y^{(i)} \cdot (x^{(i)}-\mu_1)^T\Sigma^{-1}(x^{(i)}-\mu_1)) \end{aligned} \tag{9}$$

Maximum likelihood estimator for π

Because only the first two terms depend on π (and they depend only on π), this means that we need to compute the value for π that maximizes the function $f(\pi) = \sum_{i=1}^{N} y^{(i)} \log\pi + (N - \sum_{i=1}^{N} y^{(i)}) \log(1-\pi)$.

If we compute the partial derivative of $f(\pi)$ with respect to π and we set it to 0, we will obtain the value of π that maximizes the function f (because $f(\pi)$ is a concave function).

$$\frac{\partial f(\pi)}{\partial \pi} = \frac{\sum_{i=1}^{N} y^{(i)}}{\pi} - \frac{N - \sum_{i=1}^{N} y^{(i)}}{1-\pi} = 0$$

Therefore,

$$\hat{\pi} = \frac{\sum_{i=1}^{N} y^{(i)}}{N} \tag{10}$$

Maximum likelihood estimator for μ_0

Because only the fifth term from Eq. (9) depends on μ_0, we will compute the value of μ_0 that maximizes $\frac{-1}{2}\sum_{i=1}^{N}(1-y^{(i)}) \cdot (x^{(i)}-\mu_0)^T\Sigma^{-1}(x^{(i)}-\mu_0) = h_0(\mu_0)$.

$$\frac{\partial h_0(\mu_0)}{\partial \mu_0} = \frac{1}{2}\sum_{i=1}^{N}(1-y^{(i)}) \cdot \frac{\partial (x^{(i)}-\mu_0)^T\Sigma^{-1}(x^{(i)}-\mu_0)}{\partial \mu_0} \tag{11}$$

As we have seen in the first lecture, we can write that:

$$\frac{\partial (x^{(i)}-\mu_0)^T\Sigma^{-1}(x^{(i)}-\mu_0)}{\partial \mu_0} = -2 \cdot \Sigma^{-1} \cdot (x^{(i)}-\mu_0) \tag{12}$$

By using Eq. (11) and Eq. (12), we can derive the following result:

$$\frac{\partial h_0(\mu_0)}{\partial \mu_0} = \sum_{i=1}^{N}(1-y^{(i)}) \cdot \Sigma^{-1} \cdot (x^{(i)}-\mu_0) \tag{13}$$

By setting the result from Eq. (13) to 0, we can conclude that:

$$\mu_0 = \frac{\sum_{i=1}^{N} (1 - y^{(i)}) \cdot x^{(i)}}{\sum_{i=1}^{N} (1 - y^{(i)})} \tag{14}$$

Maximum likelihood estimator for μ_1

Because only the last term from Eq. (9) depends on μ_1, we will compute the value of μ_1 that maximizes $\frac{-1}{2} \sum_{i=1}^{N} y^{(i)} \cdot (x^{(i)} - \mu_1)^T \Sigma^{-1} (x^{(i)} - \mu_1) = h_1(\mu_1)$.

$$\frac{\partial h_1(\mu_1)}{\partial \mu_1} = \sum_{i=1}^{N} y^{(i)} \cdot \Sigma^{-1} \cdot (x^{(i)} - \mu_1) \tag{15}$$

By using the same reasoning as above (when computing the MLE for μ_0), we will obtain the following formula for μ_1:

$$\mu_1 = \frac{\sum_{i=1}^{N} y^{(i)} \cdot x^{(i)}}{\sum_{i=1}^{N} y^{(i)}} \tag{16}$$

Maximum likelihood estimator for Σ

As we have seen in lecture 1, we can use the following result:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \bar{x})(x^{(i)} - \bar{x})^T \tag{17}$$

Then it is easy to extend the result above to:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \tag{18}$$

Eq. (18) can be written in the following way:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \mu_0 \cdot (1 - y^{(i)}) - \mu_1 \cdot y^{(i)})(x^{(i)} - \mu_0 \cdot (1 - y^{(i)}) - \mu_1 \cdot y^{(i)})^T \tag{19}$$

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

$$\begin{aligned}
p(y = 1|x) &= \frac{p(x|y = 1) \cdot p(y = 1)}{p(x|y = 1) \cdot p(y = 1) + p(x|y = 0) \cdot p(y = 0)} \\
&= \frac{1}{1 + e^{-(\log(\frac{\pi}{1-\pi}) - \frac{1}{2} \cdot (x - \mu_1)^T \Sigma^{-1} (x - \mu_1) + \frac{1}{2} \cdot (x - \mu_0)^T \Sigma^{-1} (x - \mu_0))}} \\
&= \sigma(\log(\frac{\pi}{1 - \pi}) - \frac{1}{2} \cdot (x - \mu_1)^T \Sigma^{-1} (x - \mu_1) + \frac{1}{2} \cdot (x - \mu_0)^T \Sigma^{-1} (x - \mu_0)) \\
&= \sigma(z)
\end{aligned} \tag{20}$$

$$\begin{aligned}
z &= \log(\frac{\pi}{1 - \pi}) - \frac{1}{2} \cdot (x - \mu_1)^T \Sigma^{-1} (x - \mu_1) + \frac{1}{2} \cdot (x - \mu_0)^T \Sigma^{-1} (x - \mu_0) \\
&= (\Sigma^{-1} \cdot (\mu_1 - \mu_0))^T \cdot x + \log(\frac{\pi}{1 - \pi}) - \frac{1}{2} \cdot (\mu_1 + \mu_0)^T \Sigma^{-1} (\mu_1 - \mu_0) \\
&= \beta^T x + \gamma
\end{aligned} \tag{21}$$

Where:

$\beta = \Sigma^{-1} \cdot (\mu_1 - \mu_0)$

$\gamma = \log(\frac{\pi}{1-\pi}) - \frac{1}{2} \cdot (\mu_1 + \mu_0)^T \Sigma^{-1} (\mu_1 - \mu_0)$

From Eq. 20 and Eq. 21, we can write the following:

$$p(y = 1|x) = \sigma(\beta^T x + \gamma) \tag{22}$$

By analysing Eq. 22, we can see the exact form of logistic regression, with the parameters β and γ described above.

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

By setting Eq. 20 equal to 0.5, we can write the equation of the line $p(y = 1|x) = 0.5$, presented in the following:

$$\beta^T x + \gamma = 0 \tag{23}$$

The graphical representations of the line defined by this equation for all the three training and testing datasets (trainA, trainB, trainC, testA, testB, test C) are presented in Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5, Fig. 6.

2.2 Logistic regression

a Give the numerical values of the parameters learnt.

The numerical values of the learnt parameters are presented in the following.

- **trainA**

$$w = \begin{pmatrix} 4.258 \\ -5.427 \end{pmatrix} \quad b = 1.102$$

- **trainB**

$$w = \begin{pmatrix} 3.216 \\ -4.076 \end{pmatrix} \quad b = 3.025$$

- **trainC**

$$w = \begin{pmatrix} 0.838 \\ -1.918 \end{pmatrix} \quad b = 6.008$$

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

As we have seen in the previous exercise, the equation of the line $p(y = 1|x) = 0.5$ is $w^T x + b = 0$. The graphical representations of the line defined by this equation on all the three training and testing datasets (trainA, trainB, trainC, testA, testB, test C) are presented in the Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11, Fig. 12.

2.3 Linear regression

a Give the numerical values of the parameters learnt.

The numerical values of the learnt parameters are presented in the following.

- **trainA**

$$w = \begin{pmatrix} 0.055 \\ -0.176 \end{pmatrix} \quad b = 1.383$$

- **trainB**

$$w = \begin{pmatrix} 0.082 \\ -0.147 \end{pmatrix} \quad b = 0.882$$

- **trainC**

$$w = \begin{pmatrix} 0.016 \\ -0.158 \end{pmatrix} \quad b = 1.640$$

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

The equation of the line $p(y = 1|x) = 0.5$ is $w^T x + b - 0.5 = 0$. The graphical representations of the line defined by this equation on all the three training and testing datasets (trainA, trainB, trainC, testA, testB, test C) are presented in the Fig. 13, Fig. 14, Fig. 15, Fig. 16, Fig. 17, Fig. 18.

2.4 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. 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.

The misclassification errors for every method on all the training and testing datasets are presented in Table 1.

Method	Err. trainA	Err. trainB	Err. trainC	Err. testA	Err. testB	Err. testC
LDA	0.0%	2.0%	2.67%	1.0%	4.5%	4.0%
Logistic regression	0.0%	2.0%	3.33%	2.0%	5.0%	4.33%
Linear regression	0.0%	2.0%	2.67%	1.0%	4.5%	4.0%
QDA	0.0%	1.5%	2.67%	1.0%	2.5%	4.33%

Table 1: Misclassification errors for every method on all the training and testing datasets

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.

By inspecting the results depicted in Table 1, we can observe that all the algorithms obtain 0.0% misclassification error on the training set A. This can be explained by the fact that, as can be seen from the figures, this training set is linearly separable, so it can be learned by all the methods (LDA, logistic regression and linear regression can only learn perfectly linearly separable datasets; QDA can also learn nonlinear decision boundaries). Furthermore, this is the smallest training dataset (it has only 100 pairs (x, y)), which also makes it easier to learn by all the methods.

On the dataset trainB, QDA yields the best performance (1.5% error), whereas all the other methods provide a 2.0% misclassification error. This could be explained by the fact that QDA has a nonlinear classification boundary, which allows it to better classify data which is not linearly separable, as is the case for trainB (it's easy to see from Fig. 20).

On the dataset trainC, logistic regression performs poorly (3.33% error) and all the other methods obtain identical results on this dataset (2.67%). This dataset is the most difficult to learn, since the datapoints from the two classes are not as easy to separate visually into different clusters as for the previous training sets.

On the dataset testA, logistic regression obtains again the highest misclassification error (2.0% error) and, once again, all the other methods obtain identical results on this dataset (1.0% error).

Similarly, on the dataset testB, the logistic regression yields to the lowest accuracy, whereas LDA and Linear regression obtain identical results and QDA obtains the best performance, by surpassing considerably all the other methods. This could be because QDA is the only method which has a nonlinear decision boundary.

On the last dataset, LDA and Linear regression yield the same result (4.0% misclassification error), while QDA and Logistic regression obtain identical results, slightly less performant (4.33% miscalssification error).

The overall worst performance is obtained by the logistic regression method. Linear regression and LDA obtain identical results on all training and test datasets, which may be an indicator that LDA and linear regression are, actually, highly similar methods. QDA obtains the best results on all datasets, except for testC. This is probably not very surprising, since it is the only classifier with a nonlinear decision boundary, so the most expressive. However, this added expressivity comes at the cost of more trainable parameters, so a higher probability to overfit. Overfitting doesn't seem to happen for these datasets, though, probably because the number of parameters for each method is much lower than the number of training points for every dataset: linear and logistic regression both have 3 trainable parameters, LDA has 9 and QDA has 13, while the training sets contain hundreds of examples.

Overall, it can be seen that the performance of every method decreases from trainA to trainB and from trainB to trainC. This is what we would expect, intuitively, since trainA is easier than trainB and trainB is easier than trainC. This is because trainA is linearly separable, while trainB and trainC are not and there are more blue points in the green cluster and vice versa in trainC than in trainB. This observation also applies between testA and testB, but not between testB and testC. Perhaps this could be motivated by the fact that both trainC and testC are too difficult to learn perfectly both by linear classifiers and by QDA, so the differences between the classifiers become harder to interpret.

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

a Implement the maximum likelihood estimator and apply it to the data. Provide the numerical values of the learnt parameters.

By using a similar reasoning as for the LDA model (the only difference is that, in this case, the covariances for the two Gaussian distributions $x|y = 1$ and $x|y = 0$ are different), we will obtain the following maximum likelihood estimators:

$$\hat{\pi} = \frac{\sum_{i=1}^{N} y^{(i)}}{N} \tag{24}$$

$$\mu_0 = \frac{\sum_{i=1}^{N} (1 - y^{(i)}) \cdot x^{(i)}}{\sum_{i=1}^{N} (1 - y^{(i)})} \tag{25}$$

$$\mu_1 = \frac{\sum_{i=1}^{N} y^{(i)} \cdot x^{(i)}}{\sum_{i=1}^{N} y^{(i)}} \tag{26}$$

$$\Sigma_0 = \frac{\sum_{i=1}^{N}((x^{(i)} - \mu_0)) \cdot (x^{(i)} - \mu_0))^T)(1 - y^{(i)})}{\sum_{i=1}^{N}(1 - y^{(i)})} \tag{27}$$

$$\Sigma_1 = \frac{\sum_{i=1}^{N}((x^{(i)} - \mu_0)) \cdot (x^{(i)} - \mu_0))^T)y^{(i)}}{\sum_{i=1}^{N} y^{(i)}} \tag{28}$$

The numerical values of the learnt parameters are presented in the following.

- **trainA**

$\pi = 0.48$

$\mu_0 = \begin{pmatrix} 10.732 \\ 10.939 \end{pmatrix}$

$\mu_1 = \begin{pmatrix} 11.032 \\ 5.992 \end{pmatrix}$

$\Sigma_0 = \begin{pmatrix} 0.464 & 0.098 \\ 0.098 & 0.713 \end{pmatrix}$

$\Sigma_1 = \begin{pmatrix} 0.722 & 0.182 \\ 0.182 & 0.934 \end{pmatrix}$

- **trainB**

$\pi = 0.55$

$$\mu_0 = \begin{pmatrix} 10.582 \\ 11.171 \end{pmatrix}$$

$$\mu_1 = \begin{pmatrix} 11.247 \\ 6.095 \end{pmatrix}$$

$$\Sigma_0 = \begin{pmatrix} 0.761 & 0.053 \\ 0.053 & 1.107 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 2.365 & 1.231 \\ 1.231 & 2.840 \end{pmatrix}$$

- **trainC**

$\pi = 0.416$

$$\mu_0 = \begin{pmatrix} 10.619 \\ 10.838 \end{pmatrix}$$

$$\mu_1 = \begin{pmatrix} 11.184 \\ 6.042 \end{pmatrix}$$

$$\Sigma_0 = \begin{pmatrix} 1.285 & -0.433 \\ -0.433 & 1.826 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 1.267 & 0.457 \\ 0.457 & 1.441 \end{pmatrix}$$

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

The graphical representations of the conic defined by this equation on all three training and testing datasets (trainA, trainB, trainC, testA, testB, test C) are presented in Fig. 19, Fig. 20, Fig. 21, Fig. 22, Fig. 23, Fig. 24.

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

The misclassification errors for QDA for both train and test data are presented in Table 1.

d Comment the results as previously.

Comments were provided in Exercise 2.4 b, as was also done for LDA, logistic regression and linear regression.

A Figures

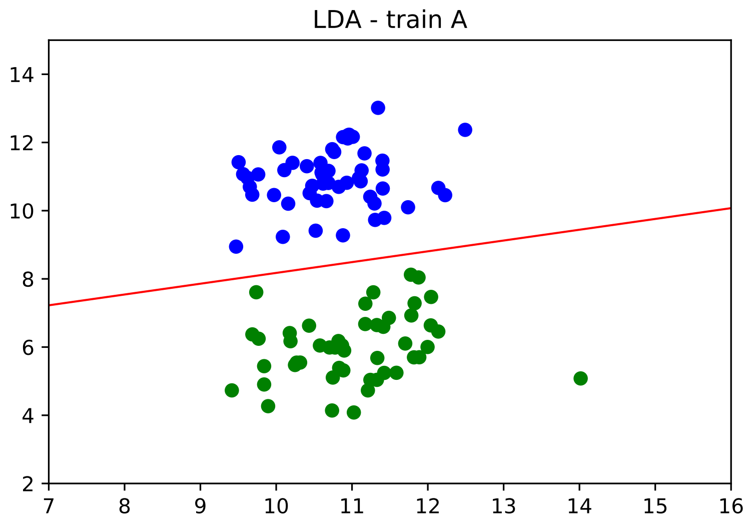

Figure 1: LDA - trainA.

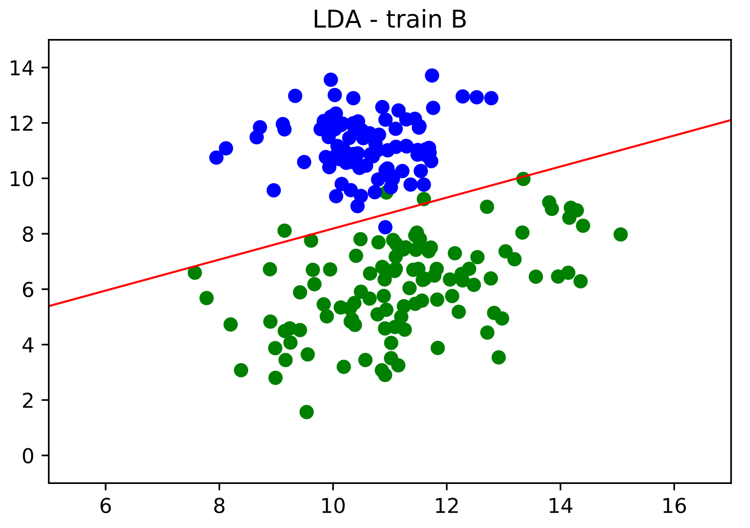

Figure 2: LDA - trainB.

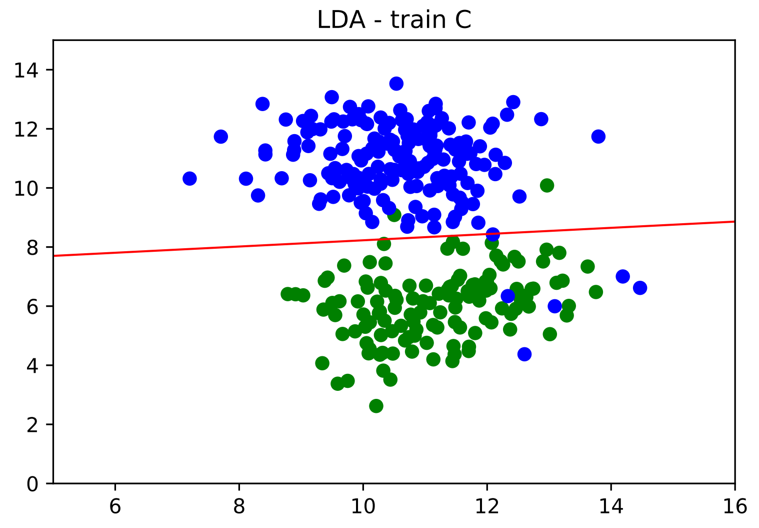

Figure 3: LDA - trainC.

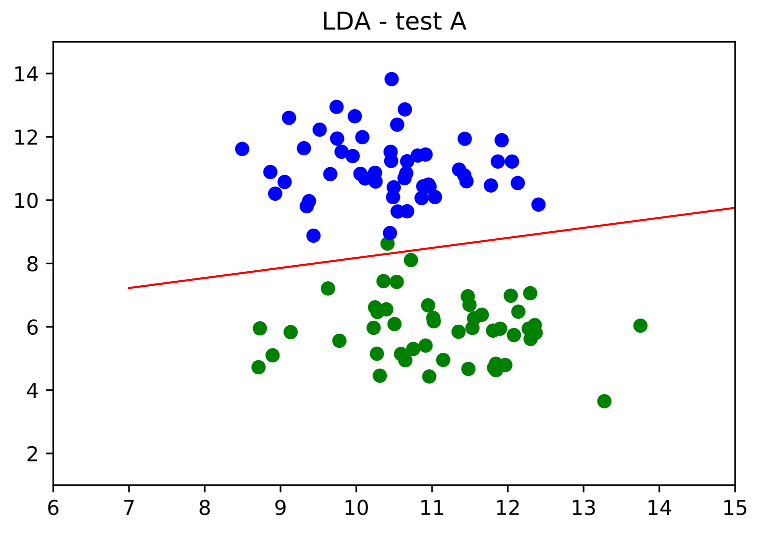

Figure 4: LDA - testA.

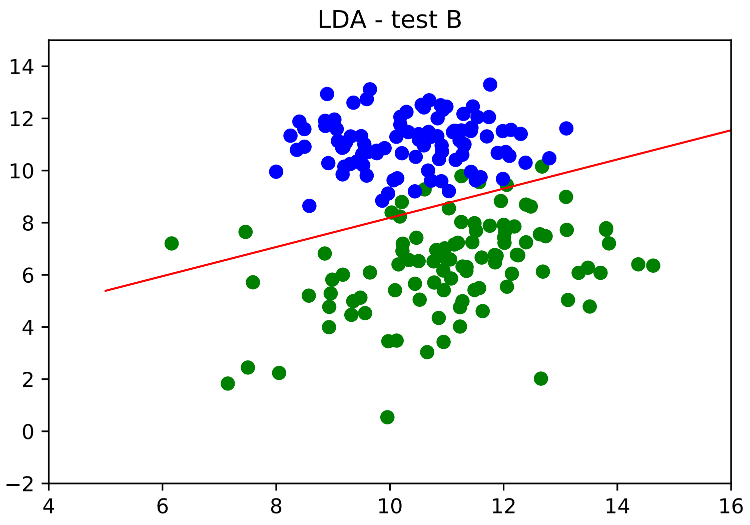

Figure 5: LDA - testB.

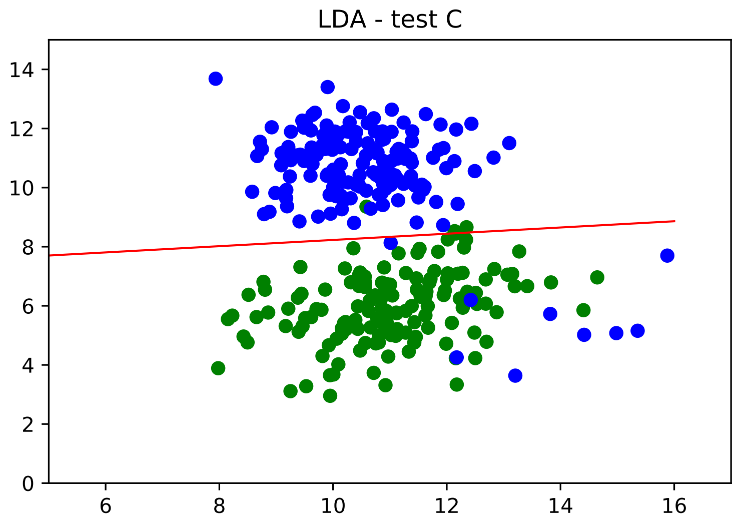

Figure 6: LDA - testC.

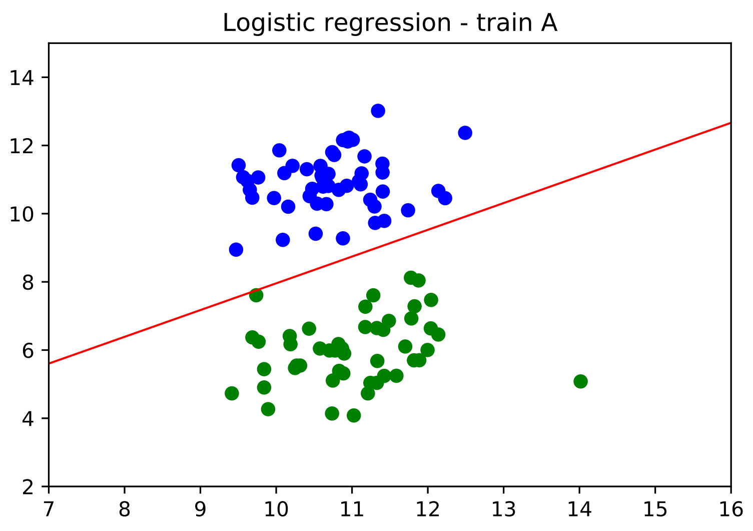

Figure 7: Logistic regression - trainA.

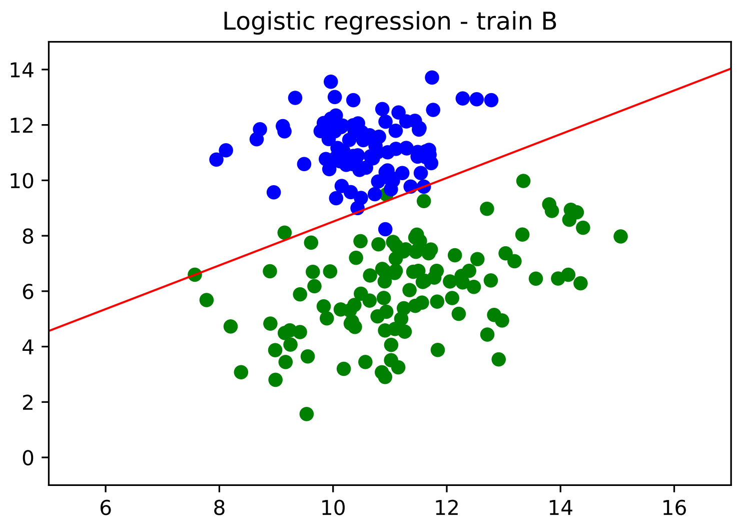

Figure 8: Logistic regression - trainB

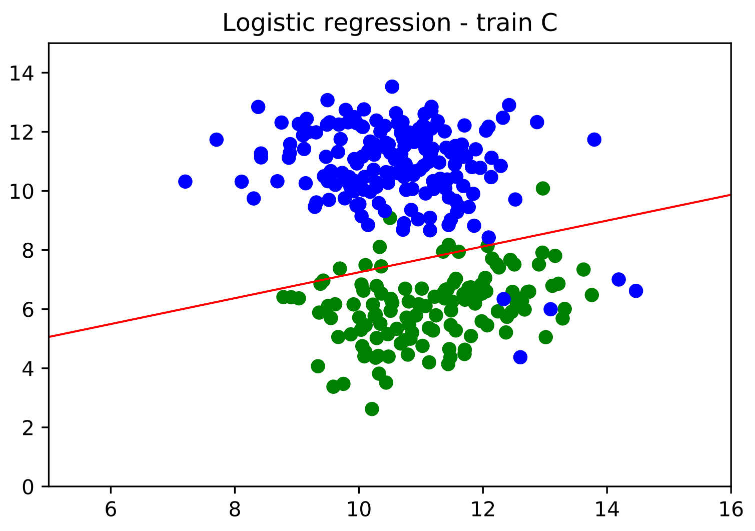

Figure 9: Logistic regression - trainC

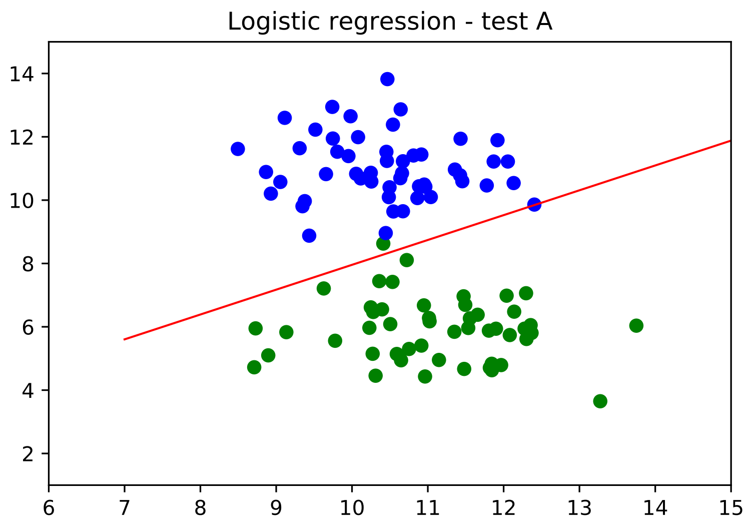

Figure 10: Logistic regression - testA

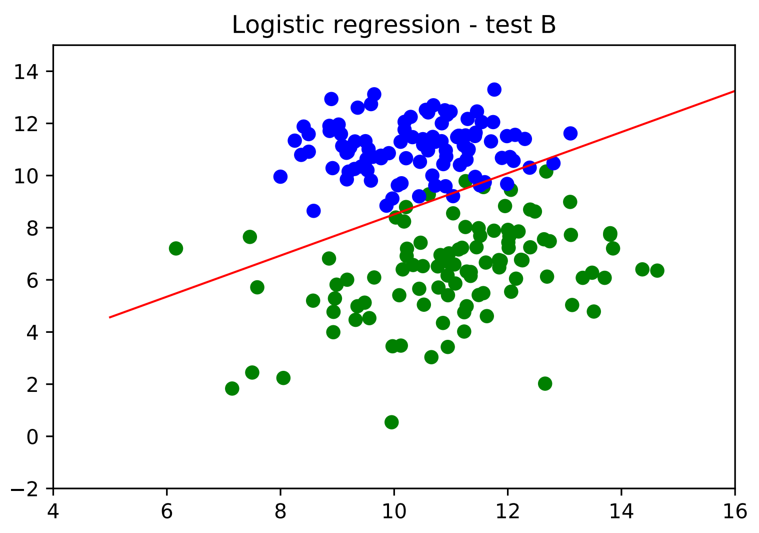

Figure 11: Logistic regression - testB

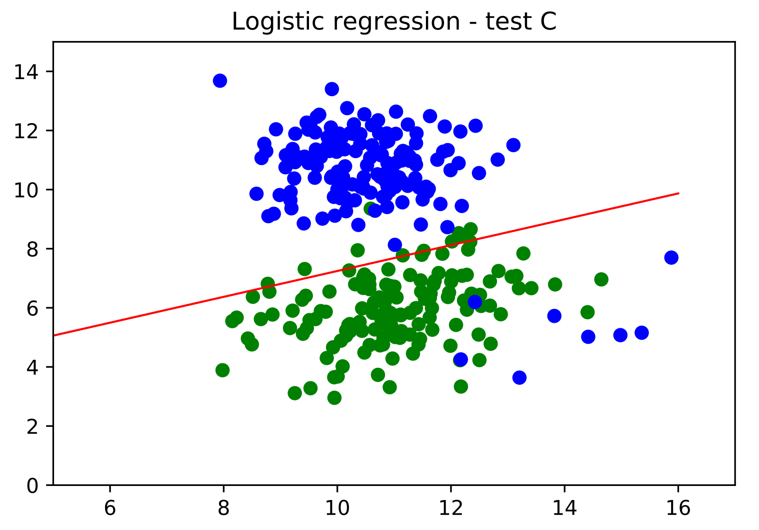

Figure 12: Logistic regression - testC

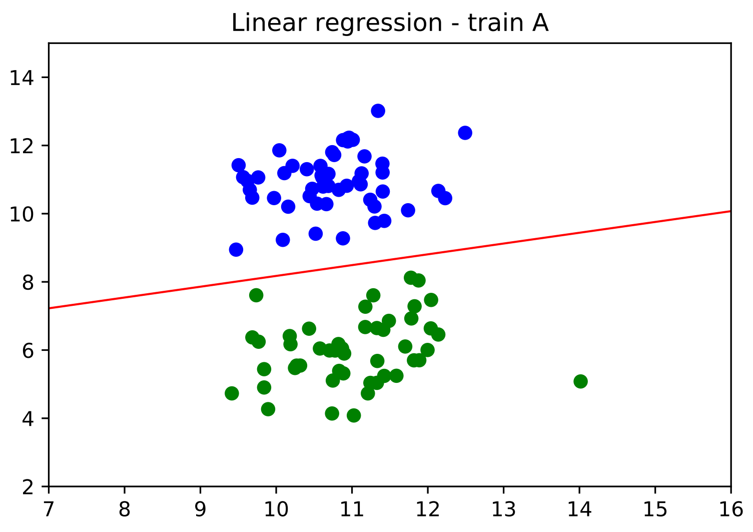

Figure 13: Linear regression - trainA.

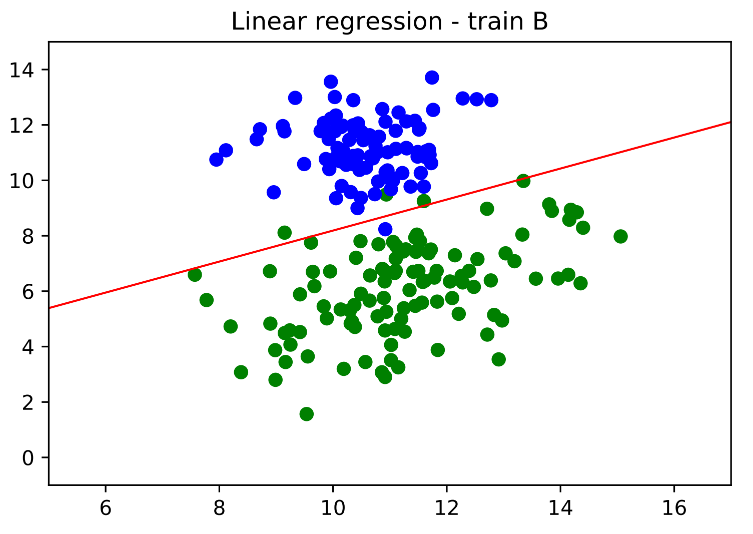

Figure 14: Linear regression - trainB

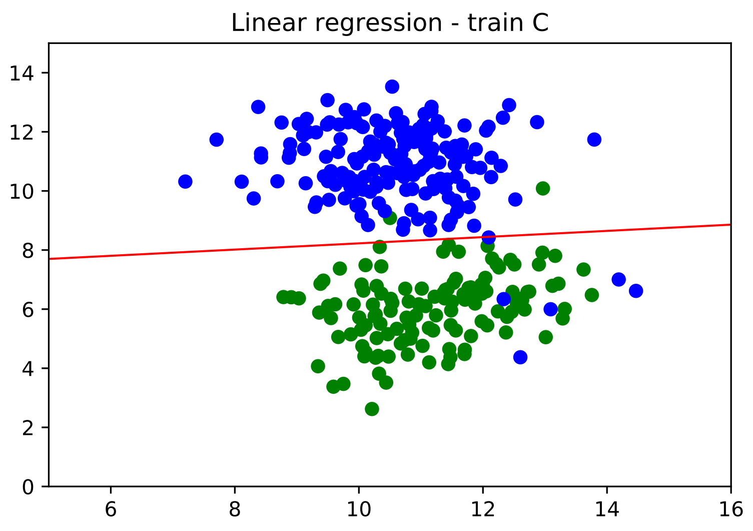

Figure 15: Linear regression - trainC

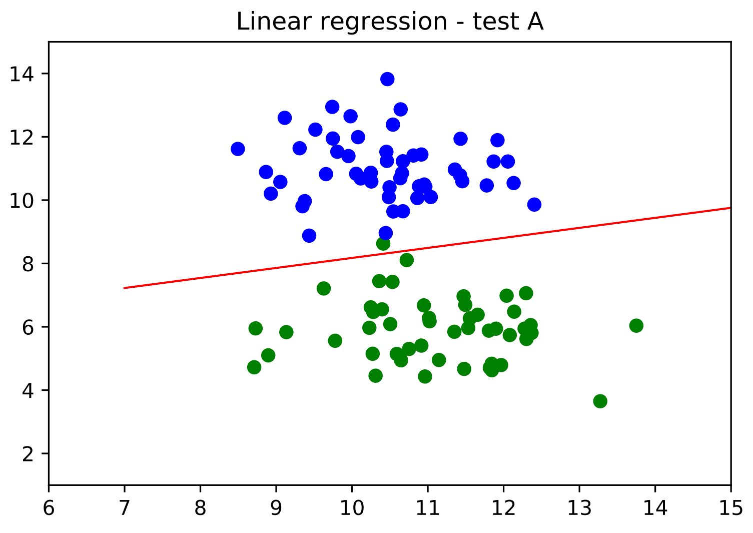

Figure 16: Linear regression - testA

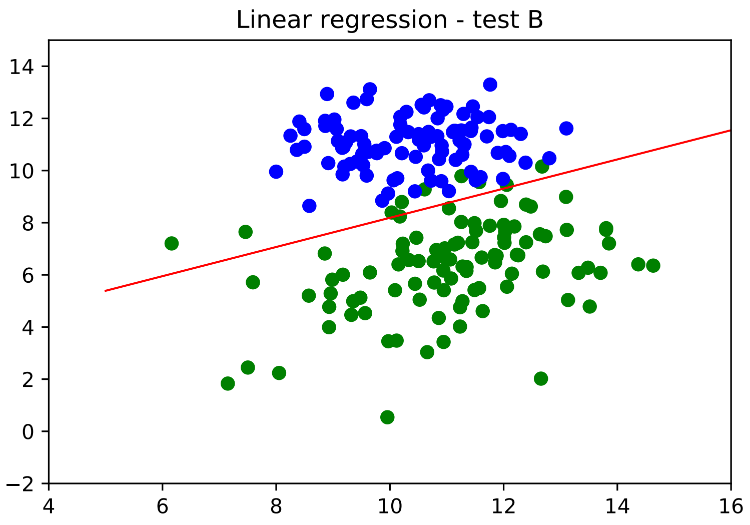

Figure 17: Linear regression - testB

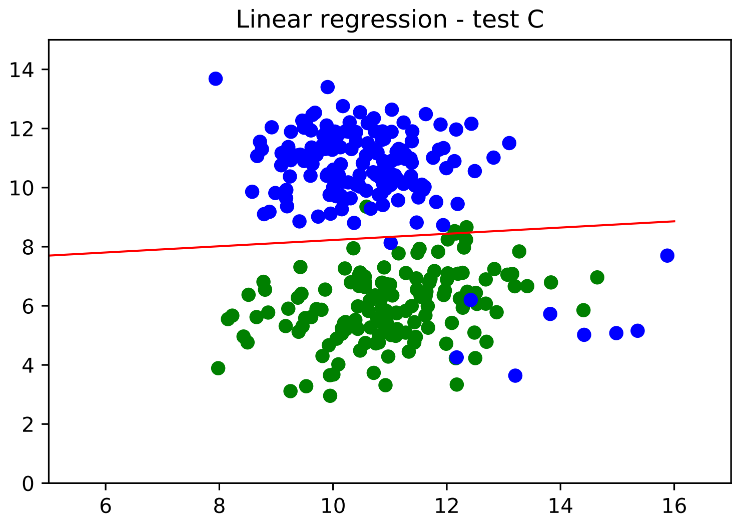

Figure 18: Linear regression - testC

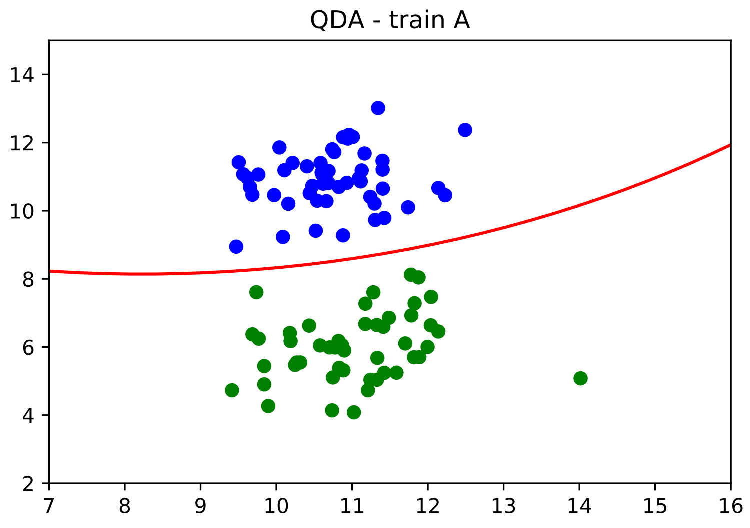

Figure 19: QDA - trainA

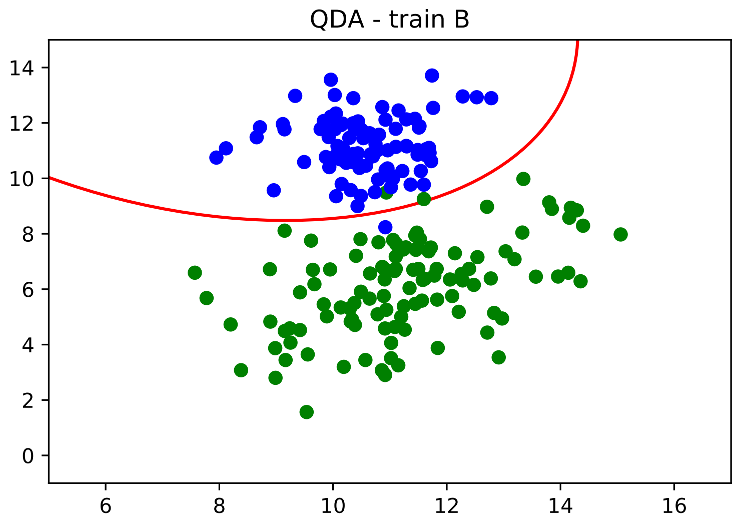

Figure 20: QDA - trainB

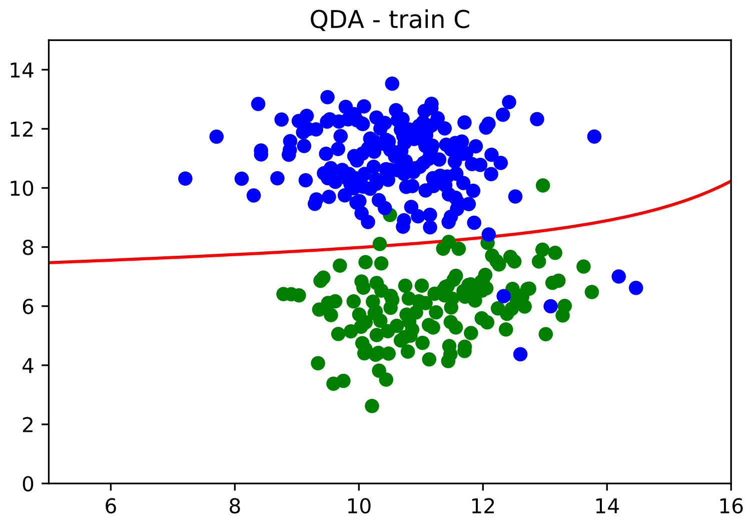

Figure 21: QDA - trainC

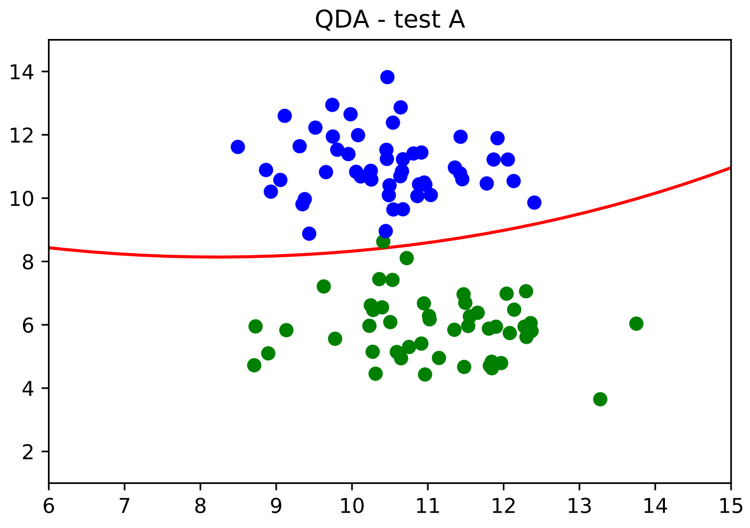

Figure 22: QDA - testA

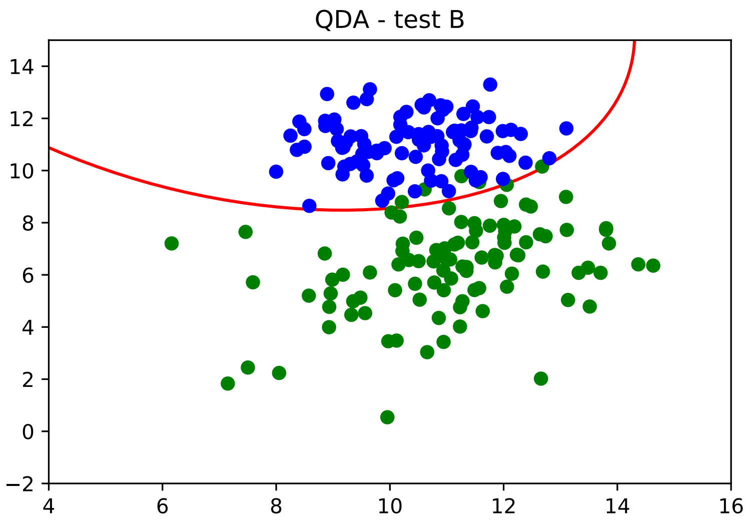

Figure 23: QDA - testB

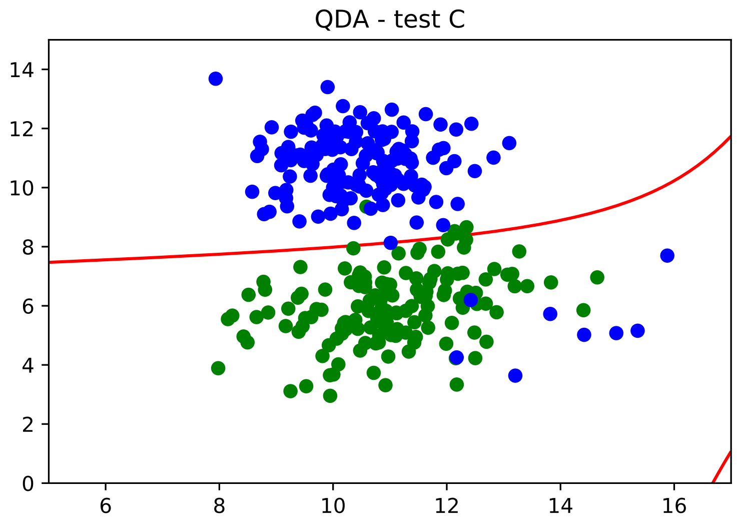

Figure 24: QDA - testC