Lucas Elbert
lucas.elbert@polytechnique.edu

Probabilistic Graphical Models
DM1

22/11/19

1 Learning in discrete graphical models

We have z taking on M different values labeled from 1 to M as follows: $z \in \{1, \dots, M\}$ with $p(z = m) = \pi_m$. Similarly $x \in \{1, \dots, K\}$ with $p(x = k|z = m) = \theta_{mk}$.
Suppose have N data points in $D = \{(x_i, z_i)\}_{1 \leq i \leq N}$. We call $\pi = \{\pi_m\}_{1 \leq m \leq M}$ and $\theta = \{\theta_{m,k}\}_{1 \leq m \leq M, 1 \leq k \leq K}$.

$$\begin{aligned} p(D|\pi, \theta) = L(D, \pi, \theta) &= \prod_{i=1}^{N} p_{\pi,\theta}(z = z_i, x = x_i) \\ &= \prod_{i=1}^{N} p_{\pi,\theta}(z = z_i) p_{\pi,\theta}(x = x_i | z = z_i) \\ &= \prod_{i=1}^{N} \pi_{z_i} \theta_{z_i x_i} \\ \ln L(D, \pi, \theta) &= \sum_{i=1}^{N} \ln \pi_{z_i} + \ln \theta_{z_i x_i} \end{aligned}$$

We now introduce two new variables. $n_m = |\{z_i \mid z_i = m, 1 \leq i \leq N\}|$ counts the number of data points with a z_i value that hits m. Similarly $n_{mk} = |\{(x_i, z_i) \mid x_i = k, z_i = m, 1 \leq i \leq N\}|$. This allows us to reorder our equation as follows:

$$\ln L(D, \pi, \theta) = \sum_{m=1}^{M} n_m \ln \pi_m + \sum_{k=1}^{K} \sum_{m=1}^{M} n_{mk} \ln \theta_{mk}$$

Which we need to maximize given the constraints $\sum_{m=1}^{M} \pi_m = 1$, and $\forall m, \sum_{k=1}^{K} \theta_{mk} = 1$. We thus form the following Lagrangian:

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

We differentiate our Lagrangian by all the parameters for each value of m and k:

$$\frac{\partial \mathcal{L}(D, \pi, \theta)}{\partial \pi_m} = \frac{n_m}{\pi_m} - \lambda \qquad \frac{\partial \mathcal{L}(D, \pi, \theta)}{\partial \theta_{mk}} = \frac{n_{mk}}{\theta_{mk}} - \lambda_m \qquad \frac{\partial \mathcal{L}(D, \pi, \theta)}{\partial \lambda} = \sum_{m=1}^{M} \pi_m - 1$$

Setting the first equation to 0 we get $\forall m,\ \lambda \pi_m = n_m$. Add all such equations: $\lambda \left(\sum_{m=1}^{M} \pi_m \right) = N \ \Rightarrow \ \lambda = N$. We repeat the same process to the second set of equations to get $\lambda_m = n_m$. Hence plugging back in we get that:

$$\pi_m = \frac{n_m}{N} \qquad \theta_{mk} = \frac{n_{mk}}{n_m}$$

Note: In the formulation of our maximization problem, we have not included the constraint that all probabilities should be in $[0, 1]$. However the results attained above respect these bounds nonetheless, so in the end it is not necessary to add these constraints.

2 Linear classification

2.1 Generative model (LDA)

We assume the following distribution of the data $\mathcal{D} = \{(x_n, y_n)\}_{1 \leq n \leq N}$

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

a.) Maximum likelihood estimators
The MLEs are:

$$\hat{\pi} = \frac{1}{N}\sum_{n=1}^{N} y_n \qquad \hat{\mu}_{0/1} = \frac{1}{|X_{0/1}|}\sum_{x \in X_{0/1}} x \qquad \hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{\mu}_{y_n})(x_n - \hat{\mu}_{y_n})^\top$$

With $X_0 = x_n|y_n = 0, X_1 = x_n|y_n = 1$.

To derive these MLEs we maximize the likelihood function $L(\pi, \mu_0, \mu_1, \Sigma, \mathcal{D})$ which is equivalent to maximizing the log-likelihood $\ln(L)$. We first write down the log-likelihood function using the definitions of Bernoulli and Gaussian distributions.

$$\begin{aligned}
\ln(L)(\pi, \mu_0, \mu_1, \Sigma, \mathcal{D}) &= \ln \prod_{n=1}^{N} p(x_n|y_n)p(y_n) \\
&= \sum_{n=1}^{N} \ln\big(\frac{1}{\sqrt{(2\pi)^2|\Sigma|}} \exp(\frac{-1}{2}(x_n - \mu_{y_n})^\top \Sigma^{-1}(x_n - \mu_{y_n})\big) \\
&\quad + \sum_{n=1}^{N} y_n \ln(\pi) + (1 - y_n)\ln(1 - \pi) \\
&= N \ln\Big(\frac{1}{\sqrt{(2\pi)^2|\Sigma|}}\Big) - \frac{1}{2}\sum_{n=1}^{N}(x_n - \mu_{y_n})^\top \Sigma^{-1}(x_n - \mu_{y_n}) \\
&\quad + \big(\ln(\pi) - log(1 - \pi)\big)\sum_{n=1}^{N} y_n + N\ln(1 - \pi)
\end{aligned}$$

As the log-likelihood of a gaussian and of a bernoulli are concave we have the sum of concave functions which is again concave. If we could find a local maximum it would therefore be global. This is what we search for. We now compute the derivatives.
With respect to π:

$$\frac{\partial \ln L}{\partial \pi} = \frac{1}{\pi(1 - \pi)}\sum_{n=1}^{N} y_n - \frac{N}{1 - \pi}$$

hence

$$\frac{\partial \ln L}{\partial \pi} = 0 \iff \pi = \frac{1}{N}\sum_{n=1}^{N} y_n$$

With respect to μ_0 (μ_1 analogous):

$$\begin{aligned}
\frac{\partial \ln L}{\partial \mu_0} &= \frac{-1}{2}\sum_{x_n \in X_0} \frac{\partial (x - \mu_0)^\top \Sigma^{-1}(x - \mu_0)}{\partial \mu_0} \\
&= \frac{1}{2}\sum_{x \in X_0} 2(x - \mu_0)^\top \Sigma^{-1} \\
&= (\sum_{x \in X_0} x^\top - |X_0|\mu_0^\top)\Sigma^{-1}
\end{aligned}$$

as Σ^{-1} is invertible this means

$$\begin{aligned}
\frac{\partial \ln L}{\partial \mu_0} = 0 &\iff \sum_{x \in X_0} x^\top - |X_0|\mu_0^\top = 0 \\
&\iff \mu_0 = \frac{1}{|X_0|}\sum_{x \in X_0} x
\end{aligned}$$

With respect to Σ^{-1}:
From the formula of the log-likelihood we can slightly simplify the derivative to start with the following

$$\frac{\partial \ln L}{\partial \Sigma^{-1}} = \frac{\partial \frac{-N}{2} \ln |\Sigma|}{\partial \Sigma^{-1}} - \frac{\partial \frac{1}{2} \sum_{n=1}^{N} (x_n - \mu_{y_n})^\top \Sigma^{-1} (x_n - \mu_{y_n})}{\partial \Sigma^{-1}}$$

We now use two identities. First we know for some matrix A and a vector v we have $v^\top A v = \sum_{i,j} v_i v_j a_{ij}$ which implies $\frac{v^\top A v}{\partial A} = vv^\top$. Secondly it is $\frac{\partial \ln(|A|)}{\partial A} = A^{-\top}$ which is a direct corollary from Jacobi's formular for determinant derivatives. With these identities we continue

$$... = \frac{N}{2}\Sigma - \frac{1}{2}\sum_{n=1}^{N}(x_n - \mu_{y_n})(x_n - \mu_{y_n})^\top$$

Which finally leads to

$$\frac{\partial \ln L}{\partial \Sigma^{-1}} = 0 \iff \Sigma = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{y_n})(x_n - \mu_{y_n})^\top$$

We therefore found expressions for a critical point of $\ln L$ and verify that indeed it is also valid in the sense that $\pi \in [0, 1]$ and Σ a valid covariance matrix. As already stated this critical point must also be a global maximum because of the concavity of $\ln L$.

b.) We compute a form of $p(y = 1|x)$ for the LDA and will finally see that this can be written in the same form as in logistic regression.

$$\begin{aligned}
p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\
&= \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1) + p(x|y=0)p(y=0)} \\
&= \frac{1}{1 + \frac{p(y=0)}{p(y=1)}\frac{p(x|y=0)}{p(x|y=1)}} \\
&= \frac{1}{1 + \frac{1-\pi}{\pi}\frac{\mathcal{N}(x;\mu_0,\Sigma)}{\mathcal{N}(x;\mu_1,\Sigma)}} \\
&= \frac{1}{1 + \frac{1-\pi}{\pi}\exp\Big(-\frac{1}{2}\big((x-\mu_0)^\top\Sigma^{-1}(x-\mu_0) - (x-\mu_1)^\top\Sigma^{-1}(x-\mu_1)\big)\Big)} \\
&= \frac{1}{1 + \frac{1-\pi}{\pi}\exp\Big(-\frac{1}{2}\big(-2(\mu_0-\mu_1)^\top\Sigma^{-1}x + (\mu_0+\mu_1)^\top\Sigma^{-1}(\mu_0-\mu_1)\big)\Big)} \\
&= \frac{1}{1 + \exp(\beta^\top x + c)}
\end{aligned}$$

for $\beta^\top = (\mu_0 - \mu_1)^\top\Sigma^{-1}$ and $c = \ln\left(\frac{1-\pi}{\pi}\right) - \frac{1}{2}(\mu_0 + \mu_1)^\top\Sigma^{-1}(\mu_0 - \mu_1)$. Which is exactly the form of a logistic regression.

c.) $p(y = 1|x) = 0.5$ can be found by using **b.)** and requiring $\beta^\top x + c = 0$ which defines a hyperplane, so in our case a line.
On the provided data sets we get the following results. The upper image always shows the training dataset with a visualization of the estimated gaussians and the linear separation line. The lower image shows the very same estimated gaussians and separation line only with the points taken from the test dataset.

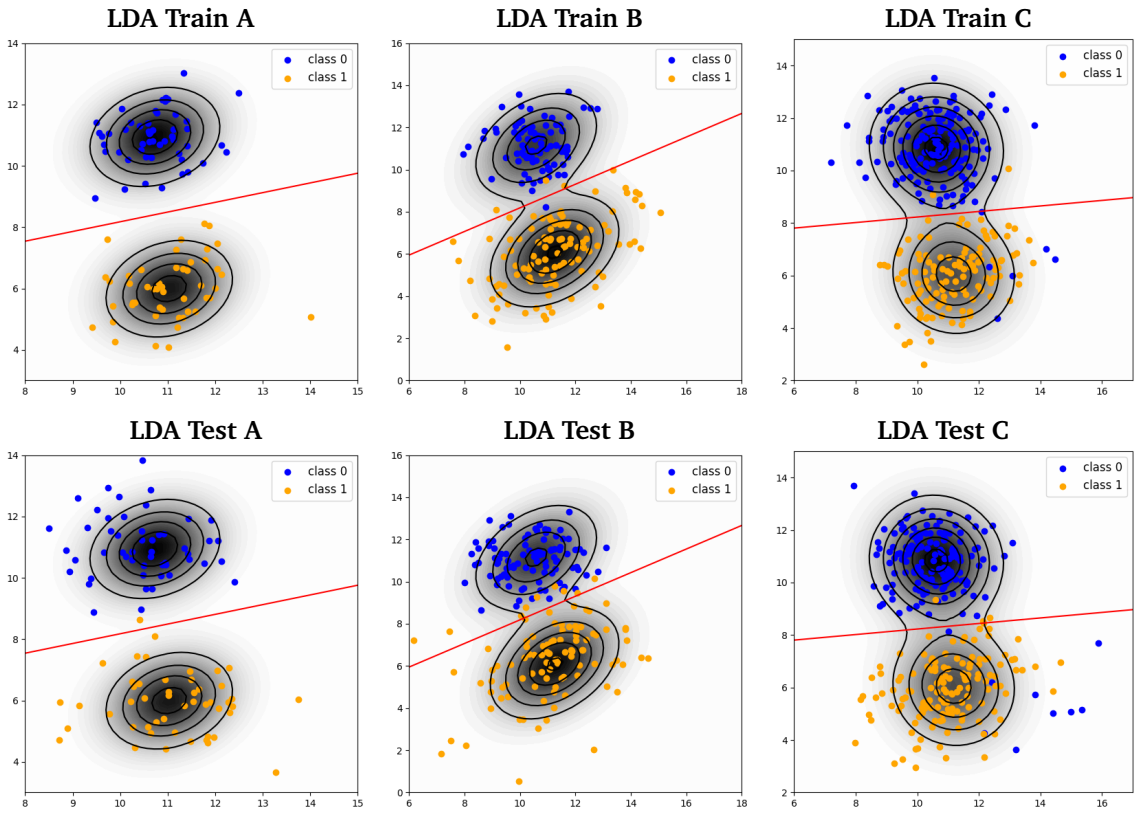

Figure 1: LDA visualizations on the three train and test sets

	μ_0	μ_1	Σ
A	$\begin{bmatrix} 10.732 \\ 10.939 \end{bmatrix}$	$\begin{bmatrix} 11.032 \\ 5.992 \end{bmatrix}$	$\begin{bmatrix} 0.588 & 0.139 \\ 0.139 & 0.819 \end{bmatrix}$
B	$\begin{bmatrix} 10.582 \\ 11.171 \end{bmatrix}$	$\begin{bmatrix} 11.247 \\ 6.095 \end{bmatrix}$	$\begin{bmatrix} 1.643 & 0.701 \\ 0.701 & 2.060 \end{bmatrix}$
C	$\begin{bmatrix} 10.619 \\ 10.838 \end{bmatrix}$	$\begin{bmatrix} 11.184 \\ 6.042 \end{bmatrix}$	$\begin{bmatrix} 1.278 & -0.062 \\ -0.062 & 1.665 \end{bmatrix}$

Table 1: LDA estimated parameters up to three decimals

2.2 Logistic regression

We wish to estimate the parameters for logistic regression of an affine function, hence we are looking for w and b that give the best results if we assume:

$$p(y = 1 \mid x) = \frac{1}{1 + \exp(w^\top x + b)}$$

We use the iterative re-weighted least squares algorithm described in the lecture notes. In the figures below, the blacker the background, the closer $p(y = 1 \mid x)$ is to 1. The cutoff where $w^\top x + b = 0$ and our regression switches from predicting $y = 1$ to $y = 0$ is, as before, the red line.

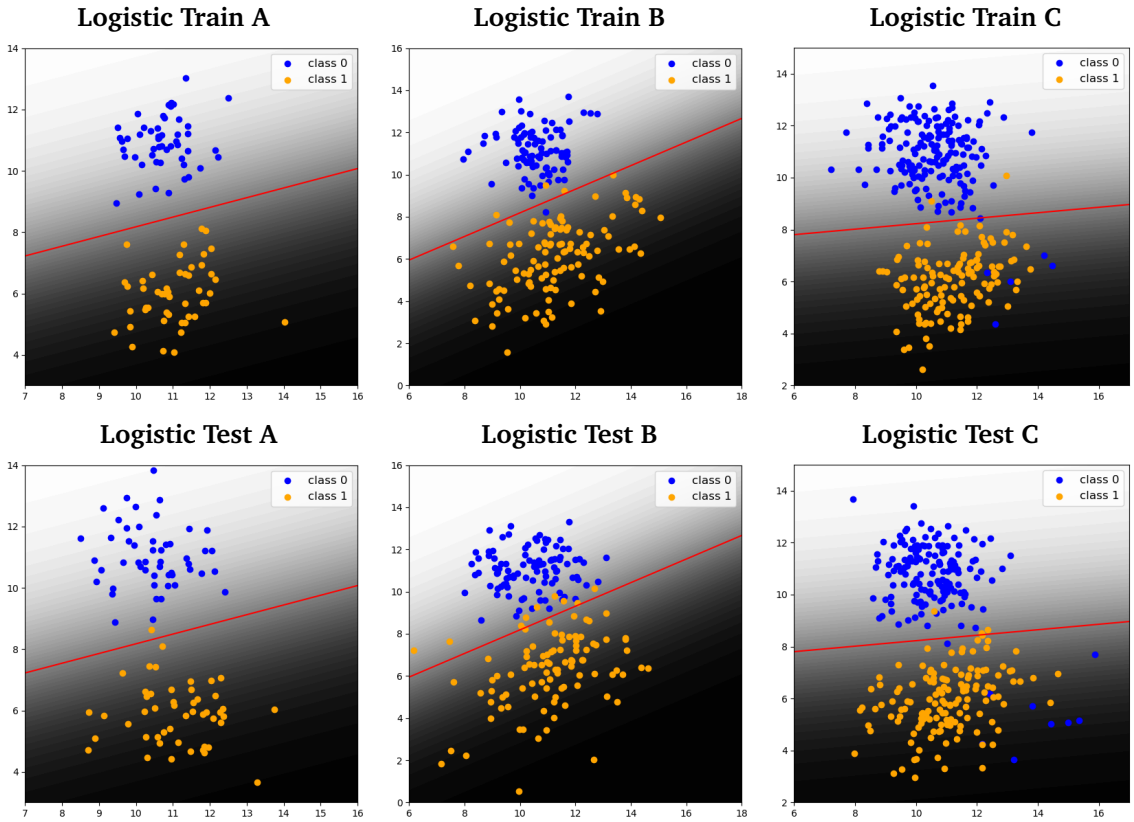

Figure 2: Logistic Regression visualizations on the three train and test sets

	w	b
A	$\begin{bmatrix} 0.223 \\ -0.705 \end{bmatrix}$	3.534
B	$\begin{bmatrix} 0.330 \\ -0.590 \end{bmatrix}$	1.530
C	$\begin{bmatrix} 0.067 \\ -0.636 \end{bmatrix}$	4.561

Table 2: Logistic Regression estimated parameters up to three decimals

2.3 Linear regression

Here we consider that $y \in \mathbb{R}$, allowing us to do a linear regression through the calculation of the normal equations (easy in numpy, using the numpy.linalg.pinv command). However this means that the results we predict for $y = w^\top x + b$ are not necessarily in $[0, 1]$.

We have represented this visually: the background is uniformly black for any value of $w^\top x + b > 1$ and uniformly white for any value $w^\top x + b < 0$. This makes the background color change in a piecewise linear fashion. From this we can tell the graphs apart from the previous ones, which would otherwise look identical, as the cutoff line seems to be in exactly the same position.

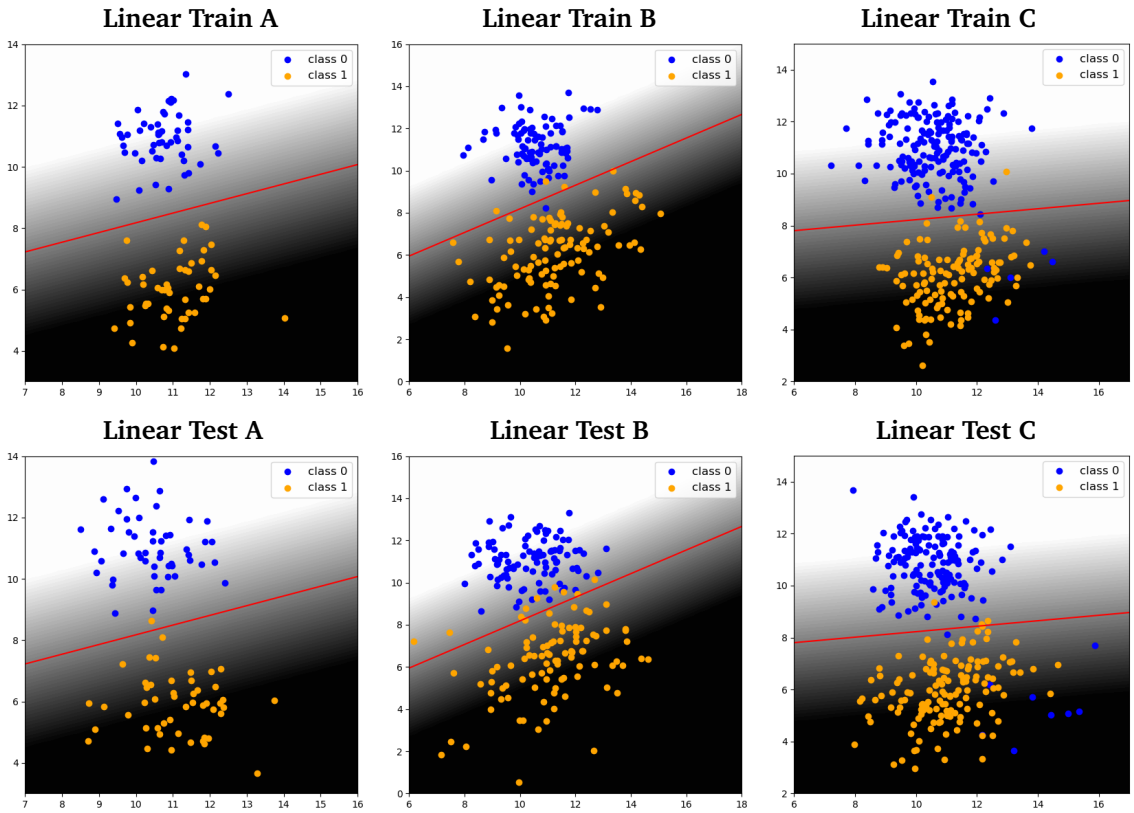

Figure 3: Linear Regression visualizations on the three train and test sets

	w	b
A	$\begin{bmatrix} 0.056 \\ -0.176 \end{bmatrix}$	1.383
B	$\begin{bmatrix} 0.083 \\ -0.148 \end{bmatrix}$	0.882
C	$\begin{bmatrix} 0.017 \\ -0.159 \end{bmatrix}$	1.640

Table 3: Linear Regression estimated parameters up to three decimals

2.4 Application

a.)

	A		B		C	
	Train	Test	Train	Test	Train	Test
LDA	0	0.01	0.02	0.045	0.02666	0.04
Logistic regression	0	0.01	0.02	0.045	0.02666	0.04
Linear regression	0	0.01	0.02	0.045	0.02666	0.04
QDA	0	0.01	0.015	0.025	0.02666	0.04333

Table 4: Misclassification errors of the different methods

b.)
We can see in the table above that LDA, Logistic Regression, and Linear Regression give us the exact same preformances on the data set. Though the three use different methods, the result in the end is a linear separation line between the data points, which end up extremely similar to each other, hence the similar results.

In every case the model performed a little worse on the test data. The difference is a little under a factor of 2. This could be attributed to two factors: 1) slight overfitting to the training data, since our training set is not very large; and 2) bad luck with outlier data points on the test sets.

These separation lines can only be so good when it comes to separating data that is not always linearly separable, so we now turn our attention to another method.

2.5 QDA model

To derive the MLEs the calculations are almost the same as for the LDA. Only the derivative with respect to Σ will now be taken once w.r.t. to Σ_0 and once w.r.t to Σ_1. It leads to the following MLEs:

$$\hat{\pi} = \frac{1}{N}\sum_{n=1}^{N} y_n \qquad \hat{\mu}_{0/1} = \frac{1}{|X_{0/1}|}\sum_{x \in X_{0/1}} x \qquad \hat{\Sigma}_{0/1} = \frac{1}{|X_{0/1}|}\sum_{x_n \in X_{0/1}} (x_n - \hat{\mu}_{y_n})(x_n - \hat{\mu}_{y_n})^\top$$

a.)
The from the training datasets A,B,C estimated parameters are shown in Table 5.

	μ_0	μ_1	Σ_0	Σ_1
A	$\begin{bmatrix} 10.732 \\ 10.939 \end{bmatrix}$	$\begin{bmatrix} 11.032 \\ 5.992 \end{bmatrix}$	$\begin{bmatrix} 0.464 & 0.098 \\ 0.098 & 0.713 \end{bmatrix}$	$\begin{bmatrix} 0.722 & 0.182 \\ 0.182 & 0.934 \end{bmatrix}$
B	$\begin{bmatrix} 10.582 \\ 11.171 \end{bmatrix}$	$\begin{bmatrix} 11.247 \\ 6.095 \end{bmatrix}$	$\begin{bmatrix} 0.761 & 0.053 \\ 0.053 & 1.107 \end{bmatrix}$	$\begin{bmatrix} 2.365 & 1.231 \\ 1.231 & 2.840 \end{bmatrix}$
C	$\begin{bmatrix} 10.619 \\ 10.838 \end{bmatrix}$	$\begin{bmatrix} 11.184 \\ 6.042 \end{bmatrix}$	$\begin{bmatrix} 1.285 & -0.433 \\ -0.433 & 1.826 \end{bmatrix}$	$\begin{bmatrix} 1.267 & 0.457 \\ 0.457 & 1.441 \end{bmatrix}$

Table 5: QDA estimated parameters up to three decimals

b.)
Figure 4 shows the points and conic separations of the datasets. The top row shows the training points with estimated gaussians and conic separation line. The bottom row shows the same estimated distributions and separation lines and overlays the test points.

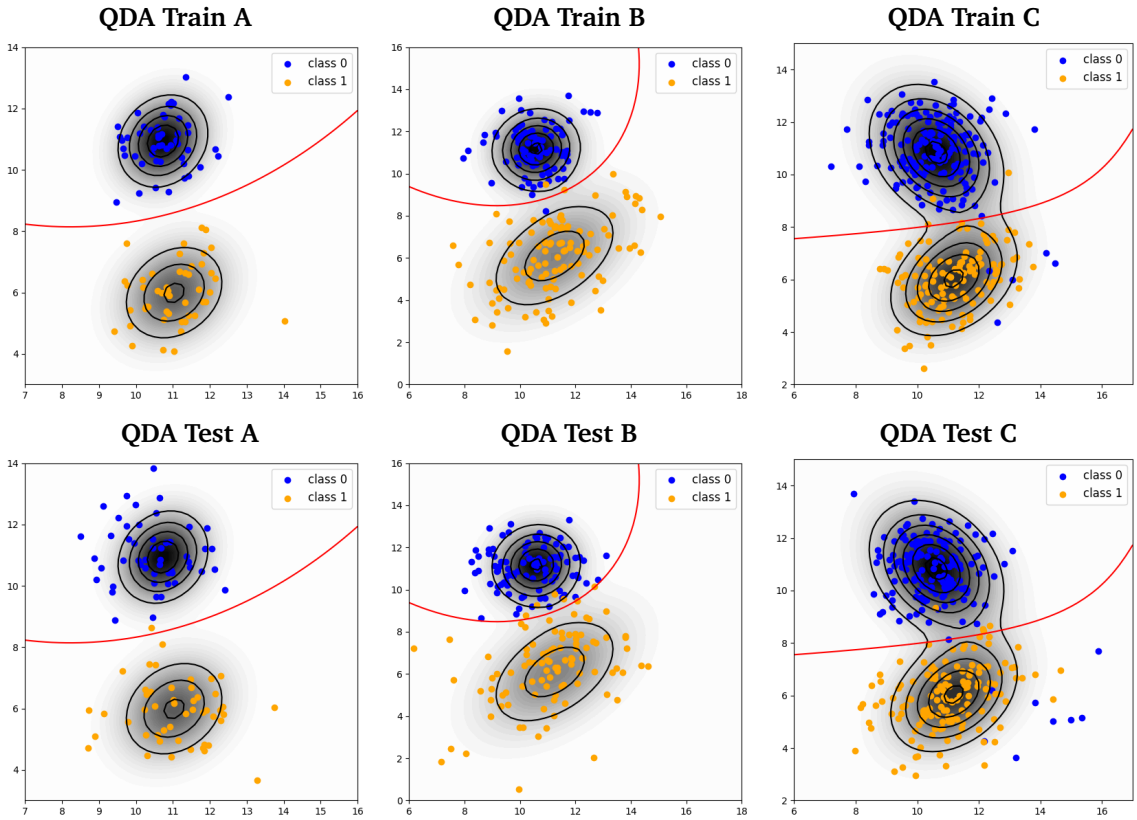

Figure 4: QDA visualizations on the three train and test sets.

Similar to the exercise LDA we can here compute $p(y = 1|x)$ as

$$p(y = 1|x) = \frac{1}{1 + \frac{1-\pi}{\pi}\sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}}\exp\Big(-\frac{1}{2}(x-\mu_0)^\top\Sigma_0^{-1}(x-\mu_0) - (x-\mu_1)^\top\Sigma_1^{-1}(x-\mu_1))\Big)}$$

So setting $p(y = 1|x) = 0.5$ can therefore be written as

$$(x-\mu_0)^\top\Sigma_0^{-1}(x-\mu_0) - (x-\mu_1)^\top\Sigma_1^{-1}(x-\mu_1) = -2\ln\Big(\frac{\pi}{1-\pi}\sqrt{\frac{|\Sigma_0|}{|\Sigma_1|}}\Big)$$

This is a quadratic equation in x which is used to define the cone.

c.)
The misclassification errors are

	A		B		C	
	Train	Test	Train	Test	Train	Test
QDA	0	0.01	0.015	0.025	0.02666	0.04333

Table 6: QDA misclassification errors

d.)
As seen before, the first three methods produced the same separation line (up to some tiny constant) and also the same misclassification errors. In dataset A this linear decision boundary was already good enough to separate the data. The QDA which has more freedom with a conic decision boundary profits the most in dataset B where it shows a lower misclassification error. In dataset C most of the misclassifications are due to outliers that we could not expect to classify correctly.