DM1 : Probabilistic Graphical models 2019/2020

AL HOUCINE KILANI
SALMANE NAOUMI

November 17, 2019

EMAILS: al_houceine.kilani@ens-paris-saclay.fr salmane.naoumi@ens-paris-saclay.fr

1 Part 1 - Learning in discrete graphical models

Let's consider the following model: z and x are discrete variables taking respectively M and K different values with $p(z = m) = \pi_m$ and $p(x = k|z = m) = \theta_{mk}$.

Let's assume we observe N i.i.d. samples : (x_i, z_i).

We have :

$$\begin{aligned}
\mathcal{L}(\theta, \pi|X, Z) &= p_{\theta,\pi}(X, Z) \\
&= \prod_{i=1}^{N} p_{\theta,\pi}(x_i, z_i) \\
&= \prod_{i=1}^{N} \prod_{k=1}^{K} \prod_{m=1}^{M} P_{\theta,\pi}(x_i = k, z_i = m) \\
&= \prod_{i=1}^{N} \prod_{k=1}^{K} \prod_{m=1}^{M} P_{\theta}(x_i = k|z_i = m) P_{\pi}(z_i = m) \\
&= \prod_{i=1}^{N} \prod_{k=1}^{K} \prod_{m=1}^{M} P_{\theta}(x_i = k|z_i = m) \prod_{i=1}^{N} \prod_{m=1}^{M} P_{\pi}(z_i = m)
\end{aligned} \tag{1}$$

where $p_{\theta,\pi}(X, Z)$ is the joint distribution (likelihood) of the variables X and Z parametrized by θ and π

To write this equation for all the samples at once, let us consider the quantities :

$\zeta_{k,x_i} = 1$ if $x_i = k$ and 0 otherwise

$\zeta_{m,z_i} = 1$ if $z_i = m$ and 0 otherwise

then we may rewrite the previous likelihood as :

$$\mathcal{L}(\theta, \pi|X, Z) = \prod_{i=1}^{N} \prod_{k=1}^{K} \prod_{m=1}^{M} \theta_{mk}^{\zeta_{k,x_i}\zeta_{m,z_i}} \prod_{i=1}^{N} \prod_{m=1}^{M} \pi_m^{\zeta_{m,z_i}} \tag{2}$$

To compute the MLE, we have to solve the following problem :

$(\theta_{MLE}, \pi_{MLE}) = \text{argmax}_{\theta,\pi}\, \mathcal{L}(\theta, \pi|X, Z) = \text{argmax}_{\theta,\pi} \log(\mathcal{L}(\theta, \pi|X, Z))$
Such that

$$\sum_{i=1}^{M} \pi_i = 1$$

$$\forall m \in [1, M], \sum_{k=1}^{K} \theta_{mk} = 1$$

We use the Lagrangian to inject the constraint into the function to maximize.

$$\begin{aligned} L(\theta, \pi, \lambda) &= \log(\mathcal{L}(\theta, \pi|X, Z)) + \lambda(1 - \sum_{m=1}^{M} \pi_m) + \sum_{m=1}^{M} \mu_m(\sum_{k=1}^{K} 1 - \theta_{mk}) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M} \zeta_{k,x_i}\zeta_{m,z_i} \log(\theta_{mk}) + \sum_{i=1}^{N}\sum_{m=1}^{M} \zeta_{m,z_i} \log(\pi_m) + \lambda(1 - \sum_{m=1}^{M} \pi_m) + \sum_{m=1}^{M} \mu_m(\sum_{k=1}^{K} 1 - \theta_{mk}) \\ &= \sum_{k=1}^{K}\sum_{m=1}^{M} \log(\theta_{mk}) \sum_{i=1}^{N} \zeta_{k,x_i}\zeta_{m,z_i} + \sum_{m=1}^{M} \log(\pi_m) \sum_{i=1}^{N} \zeta_{m,z_i} + \lambda(1 - \sum_{m=1}^{M} \pi_m) + \sum_{m=1}^{M} \mu_m(\sum_{k=1}^{K} 1 - \theta_{mk}) \end{aligned} \tag{3}$$

.

where λ is the Lagrangian factor associated with the constraint on π
and μ is the Lagrangian factor associated with the constraint on θ
We have :
$\sum_{i=1}^{N} \zeta_{k,x_i}\zeta_{m,z_i}$ is the number of samples such that $x_i = k$ and $z_i = m$. Let us denote it n_{mk}
$\sum_{i=1}^{N} \zeta_{m,z_i}$ is the number of samples such that $z_i = m$. Let us denote it n_m
then we have

$$L(\theta, \pi, \lambda) = \sum_{k=1}^{K}\sum_{m=1}^{M} n_{mk} \log(\theta_{mk}) + \sum_{m=1}^{M} n_m \log(\pi_m) + \lambda(1 - \sum_{m=1}^{M} \pi_m) + \sum_{m=1}^{M} \mu_m(\sum_{k=1}^{K} 1 - \theta_{mk}) \tag{4}$$

. We then differetiate w.r.t :
- π_m for each $m \in [1, M]$ and set to 0 :

$$\frac{\partial L}{\partial \pi_m} = \frac{n_m}{\pi_m} - \lambda = 0$$

thus

$$\frac{n_m}{\lambda} = \pi_m$$

but since $\sum_{m=1}^{M} \pi_m = 1$ then $\lambda = \sum_{m=1}^{M} n_m = N$

therefore

$$\pi_m = \frac{n_m}{N}$$

- θ_{mk} for each $(m,k) \in [1,M] \times [1,K]$ and set to 0 :

$$\frac{\partial L}{\partial \theta_{mk}} = \frac{n_{mk}}{\pi_m} - \mu_m = 0$$

thus

$$\frac{n_{mk}}{\mu_m} = \theta_{mk}$$

but since $\forall m, \sum_{k=1}^{K} \theta_{mk} = 1$ then $\mu_m = \sum_{k=1}^{K} n_{mk} = n_m$
therefore

$$\theta_{mk} = \frac{n_{mk}}{n_m}$$

In conclusion, the MLE estimator is :

$$\theta_{MLE} = (\frac{n_{mk}}{n_m})_{(m,k)\in[1,M]\times[1,K]}$$
$$\pi_{MLE} = \frac{1}{N}(n_m)_{m\in[1,M]}$$

2 Part 2 - Linear classification

2.1 :: Generative model (LDA) ::

The data are assumed to be Gaussian with different means for different classes but with the same covariance matrix.

$$y \sim \text{ Bernoulli } (\pi), x|y = i \sim \text{Normal}\,(\mu_i, \Sigma) \tag{5}$$

Therefore, the joint distribution has the form :

$$P_{\pi,\Sigma,_{1},_{2}}(\mathrm{x}, y) = \frac{1}{(2\pi)^{1/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left[\mathbf{x} - \mu_y\right]^T \Sigma^{-1}\left[\mathbf{x} - \mu_y\right]\right) \pi^y (1-\pi)^{1-y} \tag{6}$$

Let's assume we observe N i.i.d. sample of observations: (x_i, y_i):

$$\begin{aligned} \mathcal{L}(\pi, \Sigma, _{1}, _{2}|X, Y) &= \prod_{i=1}^{N} P_{\pi,\Sigma,_{0},_{1}}(x_i, y_i) \\ &= \prod_{i=1}^{N} \frac{1}{(2\pi)^{1/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left[\mathbf{x_i} - \mu_{y_i}\right]^T \Sigma^{-1}\left[\mathbf{x_i} - \mu_{y_i}\right]\right) \pi^{y_i}(1-\pi)^{1-y_i} \end{aligned} \tag{7}$$

To get the MLE estimator, we solve the following optimisation problem : $\hat{\Sigma}, \hat{\pi}, \hat{_{1}}, \hat{_{2}} = \text{argmax}\,\mathcal{L}(\pi, \Sigma, _{1}, _{2}|x_i, y_i) = \text{argma}$

Let us denote by L the log-likelihood.
We have :

$$L(\pi, \Sigma, _1, _2|X, Y) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x_i} - \mu_{y_i})^T\Sigma^{-1}(\mathbf{x_i} - \mu_{y_i}) + \sum_{i=1}^{N} y_i log(\pi) + (1-y_i)log(1-\pi)$$

Now we differentiate w.r.t the parameters and set to 0 :

$$\frac{\partial L}{\partial \pi} = \sum_{i=1}^{N}(\frac{y_i}{\pi} - \frac{1-y_i}{1-\pi}) = 0 \implies \frac{N_1}{\pi} - \frac{N-N_1}{1-\pi} = 0$$

$$\implies \pi = \frac{N_1}{N}$$

Where N_1 is the number of observations with label 1.

$$\frac{\partial L}{\partial \mu_0} = \frac{-1}{2}\Sigma^{-1}\sum_{\substack{i=1 \\ y_i=0}}^{N}(-2x_i + 2\mu_0) = 0 \implies \sum_{\substack{i=1 \\ y_i=0}}^{N} x_i = \sum_{\substack{i=1 \\ y_i=0}}^{N}\mu_0$$

$$\implies \sum_{i=1}^{N} x_i(1-y_i) = N_0\mu_0$$

thus

$$\mu_0 = \frac{\sum_{i=1}^{N} x_i(1-y_i)}{N_0}$$

where N_0 is the number of observations with label 0

$$\frac{\partial L}{\partial \mu_1} = \frac{-1}{2}\Sigma^{-1}\sum_{\substack{i=1 \\ y_i=1}}^{N}(-2x_i + 2\mu_1) = 0 \implies \sum_{\substack{i=1 \\ y_i=1}}^{N} x_i = \sum_{\substack{i=1 \\ y_i=1}}^{N}\mu_1$$

$$\implies \sum_{i=1}^{N} x_i y_i = N_1\mu_1$$

thus

$$\mu_1 = \frac{\sum_{i=1}^{N} x_i y_i}{N_1}$$

- $\frac{\partial L}{\partial \Sigma} = \frac{-N}{2}\frac{1}{|\Sigma|}|\Sigma|(\Sigma^{-1})^T + \frac{1}{2}(\Sigma^{-1})^T\sum_{i=1}^{N}(\mathbf{x_i} - \mu_{y_i})(\mathbf{x_i} - \mu_{y_i})^T(\Sigma^{-1})^T = 0$ (Matrix cookbook)
thus

$$(\Sigma^{-1})^T\sum_{i=1}^{N}(\mathbf{x_i} - \mu_{y_i})(\mathbf{x_i} - \mu_{y_i})^T = N * I_n$$

$$(\Sigma^{T})^{-1}\sum_{i=1}^{N}(\mathbf{x_i} - \mu_{y_i})(\mathbf{x_i} - \mu_{y_i})^T = N * I_n$$

$$\sum_{i=1}^{N}(\mathbf{x_i} - \mu_{y_i})(\mathbf{x_i} - \mu_{y_i})^T = N * \Sigma^T$$

$$\frac{\sum_{i=1}^{N}(\mathbf{x_i}-\mu_{y_i})(\mathbf{x_i}-\mu_{y_i})^T}{N}=\Sigma$$

$$\frac{\sum_{\substack{i=1\\y_i=0}}^{N}(x_i-\mu_0)(x_i-\mu_0)^T+\sum_{\substack{i=1\\y_i=1}}^{N}(x_i-\mu_1)(x_i-\mu_1)^T}{N}=\Sigma$$

The form of the conditional distribution p(y = 1—x).
We have that

$$p(y=1|x)=\frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0)+p(x|y=1)p(y=1)}$$

since
$p(y=1)=\pi$
$p(y=0)=1-\pi$
$p(x|y=1)=(2\pi|\Sigma|)^{-1/2}\exp(\frac{-1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1))$
$p(x|y=0)=(2\pi|\Sigma|)^{-1/2}\exp(\frac{-1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0))$
we find that after calculating :

$$p(y=1|x)=\frac{1}{1+\frac{1-\pi}{\pi}\exp(\frac{-1}{2}[(x-\mu_1)^T\Sigma^{-1}(x-\mu_1))-(x-\mu_0)^T\Sigma^{-1}(x-\mu_0))])}$$

The term inside the exponential is :

$$x^T\Sigma^{-1}(\mu_0-\mu_1)+(\mu_0^T-\mu_1^T)\Sigma^{-1}x+\mu_1^T\Sigma^{-1}\mu_1+\mu_0^T\Sigma^{-1}\mu_0$$

since $x^T\Sigma^{-1}(\mu_0-\mu_1)$ is a scalar and $x^T\Sigma^{-1}(\mu_0-\mu_1)=((\mu_0^T-\mu_1^T)\Sigma^{-1}x)^T$
then $x^T\Sigma^{-1}(\mu_0-\mu_1)+(\mu_0^T-\mu_1^T)\Sigma^{-1}x=2x^T\Sigma^{-1}(\mu_0-\mu_1)$
Thus :

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

Which looks like a sigmoid function the same used in logistic regression.
Also :

$$\begin{aligned}p(y=1|x)=0.5 &\implies \frac{1-\pi}{\pi}\exp((\mu_0^T-\mu_1^T)\Sigma^{-1}x-1/2\mu_1^T\Sigma^{-1}\mu_1-1/2\mu_0^T\Sigma^{-1}\mu_0)=1\\ &\implies (\mu_0^T-\mu_1^T)\Sigma^{-1}x-1/2\mu_1^T\Sigma^{-1}\mu_1-1/2\mu_0^T\Sigma^{-1}\mu_0+log(\frac{1-\pi}{\pi})=0\\ &\implies Ax+b=0\end{aligned}$$

The decision function is then a hyper-plan separating the two classes.
This result is confirmed after we plot the decision function for the three datasets.

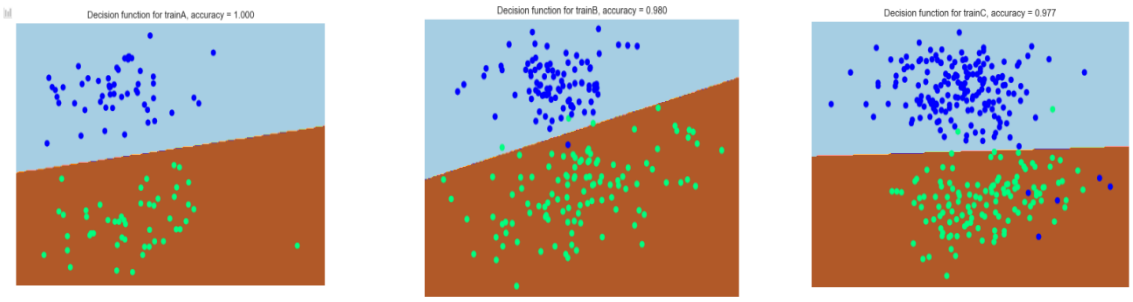

Figure 1: LDA Decision regions for the three datasets

2.2 :: Logistic regression ::

In this section, we implement logistic regression for an affine function f(x)=wx+b where the value 1 is added to vector X.

$$p(y=1|x) = \frac{1}{1+\exp(-(w^T x+b))}$$

In the case of Logistic regression, there is no close form for the solution of the parameters. Therefore, a gradient descent algorithm is performed to minimize a loss such as the cross-entropy. The Ralph-Newton algorithm is commonly used for this purpose. In fact, we use the Hessian matrix to perform each iteration of GD.

Actually, we demonstrate the following results :
$\nabla \log p\left(x,t\|\omega\right) = X^\top (y-t)$
$\nabla^2 \log[p(x,t|\omega)] = Diag(y_n, 1-y_n) \in \mathcal{M}_{n,n}(R)$

Then :
$w^{new} = w^{old} - H^{-1}\nabla E\left(w^{old}\right)$
$H = \sum_{n=1}^{N} x_n y_n \left(1-y_n\right) x_n^\top = X^\top RX$ **Where** $R = diag(y_n\left(1-y_n\right))$

Then the algorithm is :

$$\begin{aligned}
\widetilde{w}^{new} &= w^{old} - \underbrace{\left(x^\top Rx\right)^{-1}}_{H} \underbrace{x^\top (y\cdot t)}_{\nabla E(w^{old})} \\
&= \left(X^\top RX\right)^{-1}\left[\left(X^\top RX\right)w^{old} - X^\top (y-t)\right] \\
&= \left(X^\top RX\right)^{-1} X^\top R\left[Xw^{old} - R^{-1}(y-t)\right]
\end{aligned}$$

Implementing it for our datasets, we get the following parameters :

Train 1 :

$$w_1 = \begin{pmatrix} 2.66 \\ -8.90 \end{pmatrix} \quad b_1 = 47.66 \tag{8}$$

Train 2 :

$$w_2 = \begin{pmatrix} 1.84 \\ -3.71 \end{pmatrix} \quad b_2 = 13.43 \tag{9}$$

Train 3 :

$$w_3 = \begin{pmatrix} -0.27 \\ -1.91 \end{pmatrix} \quad b_3 = 18.80 \tag{10}$$

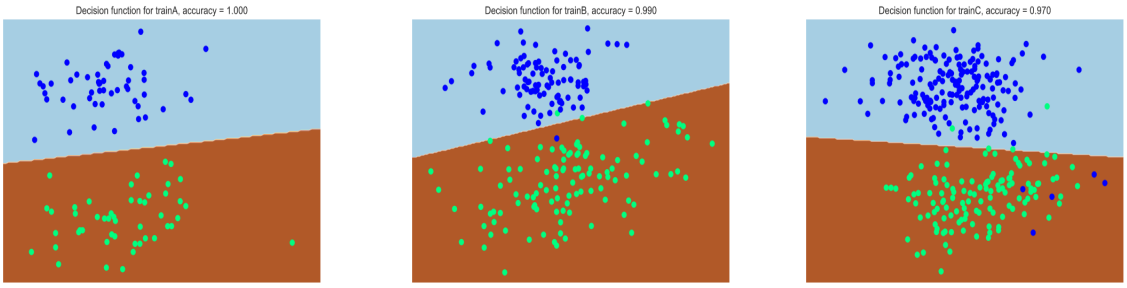

Figure 2: Logistic Regression decision regions for the three datasets

2.3 :: Linear regression ::

In this section , we will implement linear regression f(x)=wx+b . by solving the normal equations where the value 1 is added to vector X. The solution for LSE problem with L2 regularization is the following :

$$w = (X^T X + \lambda I)^{-1} X^T Y \tag{11}$$

We get the following parameters for the three datasets :

Train 1 :

$$w_1 = \begin{pmatrix} 0.111 \\ -0.352 \end{pmatrix} \quad b_1 = 1.766 \tag{12}$$

Train 2 :

$$w_2 = \begin{pmatrix} 0.165 \\ -0.295 \end{pmatrix} \quad b_2 = 0.764 \tag{13}$$

Train 3 :

$$w_3 = \begin{pmatrix} 0.033 \\ -0.317 \end{pmatrix} \quad b_3 = 2.280 \tag{14}$$

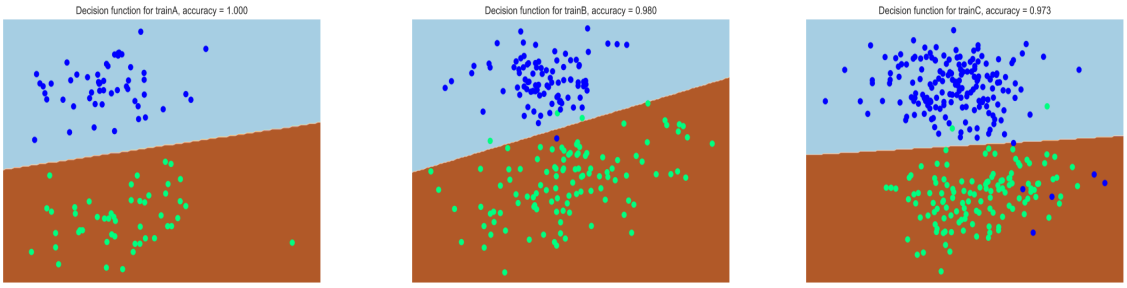

Figure 3: Linear Regression decision regions for the three datasets

2.4 :: Application ::

Applying the three algorithms on the test data, we get the following results:

Algorithm	Dataset	Train error	Test error
LDA	1	0.0%	1.0%
LDA	2	2.0%	4.5%
LDA	3	2.33%	4.33%
LIN REG	1	0.0%	1.0%
LIN REG	2	2.0%	4.5%
LIN REG	3	2.66%	4.0%
LOG REG	1	0.0%	1.0%
LOG REG	2	1.0%	3.5%
LOG REG	3	3.0%	4.6%

For all the models and datasets we see the training error is lower than the test error leading us to conclude that there is a bit of overfitting happening during training.

There is a big similarity between the decision boundaries (on the dataset A and B) of LDA and logistic regression (in terms of perfomance as well). The difference on the dataset C is that there are some outliers (blue dots mixed with the green ones) that influence the decision boundary a lot for LDA.

2.5 :: QDA Model ::

For the QDA we get the same results as LDA except for the covariance matrix (we have to differentiate w.r.t Σ_1 and Σ_2). Thus we get that :

$$\mu_0 = \frac{\sum_{i=1}^{N} x_i(1-y_i)}{N_0}$$

$$\mu_1 = \frac{\sum_{i=1}^{N} x_i y_i}{N_1}$$

$$\pi = \frac{N_1}{N}$$

$$\frac{\sum_{\substack{i=1 \\ y_i=0}}^{N} (x_i-\mu_0)(x_i-\mu_0)^T}{N} = \Sigma_0$$

$$\frac{\sum_{\substack{i=1 \\ y_i=1}}^{N} (x_i-\mu_1)(x_i-\mu_1)^T}{N} = \Sigma_1$$

The same way as before, we write the form of the conditional distribution p(y = 1—x) but this time with :

$p(x|y=1) = (2\pi|\Sigma_1|)^{-1/2} \exp(\frac{-1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1))$

$p(x|y=0) = (2\pi|\Sigma_0|)^{-1/2} \exp(\frac{-1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0))$

The decision function is then a conic separating the two classes. (we will not bother ourselves with writing the full formula this time, we will just use the fact that the former conditional probability is proportional to the two latters) (In the code we will normalize it by dividing by the sum of the two).

Concerning the performances, there is not a huge increase in accuracy because the datasets A and B were lineraly separable (except for got on the boudary that even with conic boundaries, we can not classify them well without overfitting). Same comment for dataset C that contains a lot of outliers on both sides.

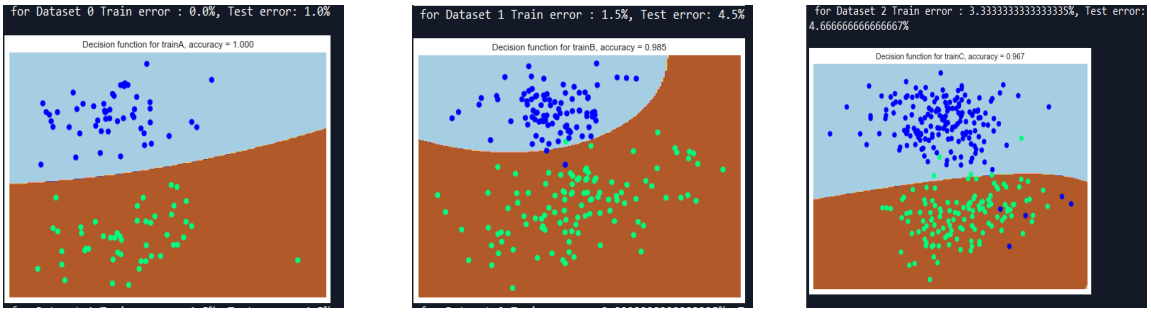

Figure 4: QDA decision regions for the three datasets

Train 1 :

$$\mu_0 = \begin{pmatrix} 10.73 \\ 10.93 \end{pmatrix} \quad \mu_1 = \begin{pmatrix} 11.03 \\ 5.99 \end{pmatrix} \quad \Sigma_0 = \begin{pmatrix} 0.24 & 0.05 \\ 0.05 & 0.37 \end{pmatrix} \quad \Sigma_1 = \begin{pmatrix} 0.34 & 0.08 \\ 0.08 & 0.44 \end{pmatrix} \quad \pi = 0.48 \tag{15}$$

Train 2 :

$$\mu_0 = \begin{pmatrix} 0.34 \\ 0.02 \end{pmatrix} \quad \mu_1 = \begin{pmatrix} 0.02 \\ 0.49 \end{pmatrix} \quad \Sigma_0 = \begin{pmatrix} 0.34 & 0.02 \\ 0.02 & 0.49 \end{pmatrix} \quad \Sigma_1 = \begin{pmatrix} 1.30 & 0.67 \\ 0.67 & 1.56 \end{pmatrix} \quad \pi = 0.55 \tag{16}$$

Train 3 :

$$\mu_0 = \begin{pmatrix} 0.75 \\ -0.25 \end{pmatrix} \quad \mu_1 = \begin{pmatrix} -0.2 \\ 1.06 \end{pmatrix} \quad \Sigma_0 = \begin{pmatrix} 0.52 & 0.19 \\ 0.19 & 0.60 \end{pmatrix} \quad \Sigma_1 = \begin{pmatrix} 10.62 & 10.83 \\ 10.83 & 6.04 \end{pmatrix} \quad \pi = 0.41 \tag{17}$$

References