Probabilistic Graphical Models - Homework 1

Louis Dumont

Due on November the 22nd, 2019.

Instructors: Pierre Latouche - Nicolas Chopin

1 Learning in Discrete Graphical Models

We consider z and x two discrete variables taking values in $|[1, M]|$ and $|[1, K]|$ (we take the freedom of considering these values towards the subject for notation simplicity as the problem is invariant by the values themselves).
We consider the following model: $p(z = m) = \pi_m$, $p(x = k|z = m) = \theta_{k,m}$.
We consider $(Z_i, X_i)_{i\in|[1,n]|}$ iid samplings of the pair(z,x) with $n \in N^*$.

The likelihood of the sampling writes:

$$\begin{aligned} l_{(z_i),(x_i)}(\pi, \theta) &= \prod_{i=1}^{n} p(z_i, x_i|\pi, \theta) \text{ (independence of the pairs } ((Z_i, X_i) \\ &= \prod_{i=1}^{n} p(z_i|\pi, \theta) \times p(x_i|z_i, \pi, \theta) \text{ (conditional probabilities, assuming } p(z_i) \neq 0) \\ &= \prod_{i=1}^{n} \pi_{z_i} \theta_{z_i,x_i} \end{aligned}$$

We derive from it the log-likelihood:

$$L_{(z_i),(x_i)}(\pi, \theta) = \sum_{i=1}^{n} (log(\pi_{z_i}) + log(\theta_{z_i,x_i}))$$

Computing the maximum likelihood estimator is tantamount to maximising the log-likelihood, thus minimising its opposite. We can thus find the MLE by solving the following optimization problem:

$$\min_{\pi,\theta} -\sum_{i=1}^{n}(log(\pi_{z_i}) + log(\theta_{z_i,x_i}))$$
$$\text{s.t} \sum_{m=1}^{M} \pi_m = 1$$
$$\forall m \in |[1,M]|,\ \sum_{k=1}^{K} \theta_{m,k} = 1$$

This is a convex minimisation problem, that is strictly feasible , thus strong duality holds.

We now denote $n_m = \sum_{i=1}^{n} 1_{\{z_i=m\}}$ and $n_{m,k} = \sum_{i=1}^{n} 1_{\{z_i=m,x_i=k\}}$, denoting the number of times where vales m and (m,k) were reached. The objective function rewrites $-\sum_{m=1}^{M}(n_m log(\pi_m) + \sum_{k=1}^{K} n_{m,k} log(\theta_{m,k})$

With these notations, we write the Lagrangian of the problem:

$$L(\pi,\theta,\mu,\widetilde{\mu}) = -\sum_{i=1}^{M}(n_m log(\pi_m) + \sum_{k=1}^{K} n_{m,k} log(\theta_{m,k})) + \mu \sum_{m=1}^{M} \pi_m + \sum_{m=1}^{M} \widetilde{\mu_m}(\sum_{k=1}^{K}(\theta_{m,k} - 1))$$

As we are in the case of strong duality, we can note π, θ, μ, $\widetilde{\mu}$ the optimal parameters of the Lagrangian and look at the first order derivative to find them.

$$\frac{\partial L}{\partial \pi_m} = -\frac{n_m}{\pi_m} + \mu$$
$$\frac{\partial L}{\partial \pi_m} = 0 \Leftrightarrow \mu\pi_m = n_m$$
$$\frac{\partial L}{\partial \theta_{m,k}} = -\frac{n_{m,k}}{\theta_{m,k}} + \widetilde{\mu}_m$$
$$\frac{\partial L}{\partial \theta_{m,k}} = 0 \Leftrightarrow \widetilde{\mu}_m\theta_{m,k} = n_{m,k}$$

The optimal parameters for the Lagrangian (which leads to optimal parameters for the original function, as we are in the case of strong duality) thus respect the following equations:

$$\forall m \in |[1,M]|,\ \mu\pi_m = n_m \tag{1}$$
$$\forall m \in |[1,M]|, k \in |[1,K]|,\ \widetilde{\mu}_m\theta_{m,k} = n_{m,k} \tag{2}$$
$$\sum_{m=1}^{M} \pi_m = 1 \tag{3}$$
$$\forall m \in |[1,M]|,\ \sum_{k=1}^{K} \theta_{m,k} = 1 \tag{4}$$

Summing (1) over all m and applying (3), we obtain $\mu = \sum_{m=1}^{M} n_m = n$.
For each m, summing (2) over all k and applying (4), we obtain $\widetilde{\mu}_m = \sum_{k=1}^{k} n_{m,k} = n_m$
Using these expressions in (1) and (2), we finally get:

$$\forall m \in |[1, M]|,\ \pi_m = \frac{n_m}{n}$$
$$\forall m \in |[1, M]|, k \in |[1, K]|,\ \theta_{m,k} = \frac{n_{m,k}}{n_m}$$

These values minimise the opposite of the log-likelihood, thus maximise the log-likelohood, thus the likelyhood. The MLE of (π, θ) is thus:

$$(\pi_m = \frac{n_m}{n})$$
$$(\theta_{m,k} = \frac{n_{m,k}}{n_m})$$

2 Linear Classification

1- Generative Model (LDA)

a)

Following the notations from the subject, the likelihood of the model writes (considering $(X_i, Y_i), i \in |[1, n]|$ iid samples following (x, y)):

$$\begin{aligned} l_{(X_i,Y_i)}(\pi, \mu, \Sigma) &= \prod_{i=1}^{n} p(x_i, y_i|\pi, \mu\Sigma) \\ &= \prod_{i=1}^{n} p(y_i|\pi, \mu\Sigma) p(x_i|y_i, \pi, \mu, \Sigma) \end{aligned}$$

Thus the log-likelihood writes:

$$
\begin{aligned}
L_{(X_i,Y_i)}(\pi,\mu,\Sigma) &= \sum_{i=1}^{n} log(p(y_i|\pi,\mu\Sigma)) + log(p(x_i|y_i,\pi,\mu,\Sigma)) \\
&= \sum_{i=1}^{n} log(y_i\mu + (1-y_i)(1-\mu)) + log(\frac{1}{2\Pi|\Sigma|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_{y_i})^T\Sigma^{-1}(x_i-\mu_{y_i})}) \\
&= \sum_{i=1}^{n} log(y_i\mu + (1-y_i)(1-\mu)) - \frac{1}{2}(x_i-\mu_{y_i})^T\Sigma^{-1}(x_i-\mu_{y_i}) + log(|\Sigma^{-1}|) + C^{te}
\end{aligned}
$$

Maximising the log-likelihood can then be done variable by variable:

$$
\begin{aligned}
\frac{\partial L}{\partial \pi} &= n_1\frac{1}{\pi} + n_0\frac{1}{1-\pi} \quad (n_i = \sum_{j=1}^{n} 1_{y_j=i}) \\
\frac{\partial L}{\partial \pi} &= 0 \Leftrightarrow \pi = \frac{n_1}{n_1+n_0} \\
\frac{\partial L}{\partial \mu_i} &= \sum_{j:y_j=1} 2\Sigma^{-1}x - 2\Sigma^{-1}\mu_i \\
\frac{\partial L}{\partial \mu_i} &= 0 \Leftrightarrow \mu_i = \frac{1}{n_i}\sum_{j:y_j=i} x_j \\
\frac{\partial L}{\partial \Sigma^{-1}} &= -\frac{1}{2}\sum_{i=1}^{n}((x_i-\mu_{y_i})^T(x_i-\mu_{y_i})) + \frac{n}{2}\Sigma \\
\frac{\partial L}{\partial \Sigma^{-1}} &= 0 \Leftrightarrow \Sigma = \frac{1}{n}\sum_{i=1}^{n}((x_i-\mu_{y_i})^T(x_i-\mu_{y_i}))
\end{aligned}
$$

The MLE thus writes:

$$
\begin{aligned}
\hat{\pi} &= \frac{n_1}{n_1+n_0} \\
\hat{\mu_i} &= \frac{1}{n_i}\sum_{j:y_j=i} x_j, \ \forall i \in \{0,1\} \\
\hat{\Sigma} &= \frac{1}{n}\sum_{i=1}^{n}((x_i-\mu_{y_i})^T(x_i-\mu_{y_i}))
\end{aligned}
$$

b)

Using the Bayes formula, we have:

$$
\begin{aligned}
p(y=1|x) &= \frac{p(y=1)p(x|y=1)}{p(y=1)p(x|y=1)+p(y=0)p(x|y=0)} \\
&= \frac{\pi \frac{1}{2\Pi|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)}}{\pi \frac{1}{2\Pi|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)} + (1-\pi)\frac{1}{2\Pi|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}e^{\frac{1}{2}f(x,\mu_0,\mu_1)}} \\
\text{Where: } f(x,\mu_0,\mu_1) &= (x-\mu_1)^T\Sigma^{-1}(x-\mu_1) - (x-\mu_0)^T\Sigma^{-1}(x-\mu_0) \\
&= -2((\mu_1-\mu_0))^T\Sigma^{-1}(x-\mu_1) - (\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0) \\
&\text{(by decomposing } x-\mu_0 = x-\mu_1\mu_1-\mu_0 \text{ and rearranging)} \\
&= -2(\mu_1-\mu_0))^T\Sigma^{-1}x + (\mu_1+\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0)
\end{aligned}
$$

Thus:

$$
\begin{aligned}
p(y=1|x) &= \frac{1}{1+e^(-(\beta^T x+\gamma)} \\
\text{Where: } \beta &= \Sigma^{-1}(\mu_1-\mu_0) \\
\gamma &= -\frac{1}{2}(\mu_1+\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0) - log(\frac{1-\pi}{\pi})
\end{aligned}
$$

We recognise the marginal probability of a logistic regression problem.
Plotting the line $p(y=1|x)=0.5$ then corresponds to plotting the line $\beta^T x+\gamma=0$.

c)

The line $p(y|x)=0.5$ is shown on Figure 1.

2- Logistic Regression

With logistic regression there is no closed-form solution for the MLE. We thus use a library (SCIPY.OPTIMIZE) to optimise the log-likelihood.

We simply recall here the expression of the log-likelihood for this model:

$$
L_{(y_1,\ldots,y_n|x_1,\ldots,x_n)}(w,b) = \sum_{i:y_i=1} log(\frac{1}{1+e^{-w^T x_i+b}}) + \sum_{i:y_i=0} log(\frac{1}{1+e^{w^T x_i+b}})
$$

a)

The learned values for w and b on the different datasets can be found in part 4 of the exercise.

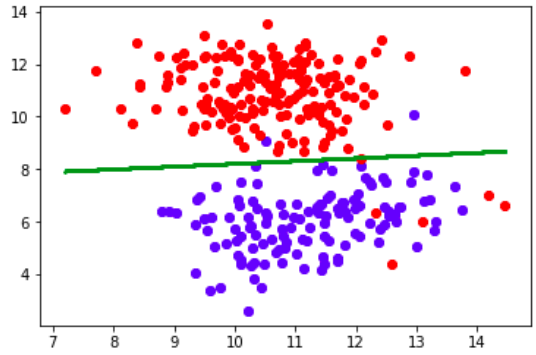

Figure 1: Line $p(y|x) = 0.5$ for the LDA classifier on dataset C

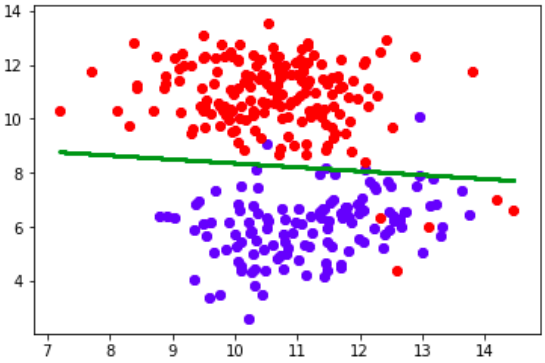

Figure 2: Line $p(y|x) = 0.5$ for the Logistic Regression classifier on dataset C

b)

The line $p(y|x) = 0.5$ is shown on Figure 2.

3- Linear Regression

We consider y as a real valued variable taking values in $0, 1$ only, and consider the linear regression model: $p(y|\tilde{x}, \tilde{w}, \sigma^2)$ follows $N(\tilde{w}^T\tilde{(x)}, \sigma^2)$ (where $\tilde{(x)} = (x_1, x_2, 1)$ and $\tilde{(w)} = (w_1, w_2, b)$).
Computing the first order conditions gives the following closed form solution for the MLE:

$$\tilde{w} = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T Y$$

From which we derive w and b.

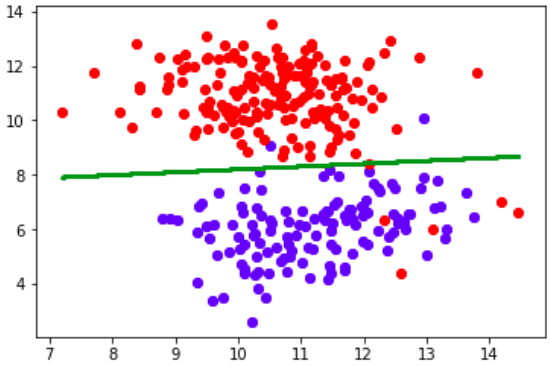

Figure 3: Line $p(y|x) = 0.5$ for the Linear Regression classifier on dataset C

a)

The learned values for w and b for the different datasets can be found in part 4 of the exercise.

b)

The line $p(y|x) = 0.5$ is shown on Figure 3.

4- Applications

a)

We use the decision function to compute the indicator $(g(y) = 1 \Leftrightarrow p(y = 1|x) \geq 0.5)$.

The learned parameters are presented in table 2:

	Linear Regression		Logistic Regression	
	w	b	w	b
Dataset A	(0.056, -0.176)	1.383	(33.92, -44.753)	47.057
Dataset B	(0.083, -0.148)	0.882	(1.842, -3.714)	13.43
Dataset C	(0.017, -0.159)	1.64	(-0.227, -1.914)	18.807

Table 1: Learned parameters for Linear and Logistic Regression on datasets A,B and C.

The results are presented in table 2:

b)

The first notice is that on these datasets all models are quite efficient (precision remains above 95% in every case). LDA and linear regression yield a slightly lower error on dataset B, whereas logistic

	LDA		Linear Regression		Logistic Regression	
	Train	Test	Train	Test	Train	Test
Dataset A	0.0	0.01	0.0	0.01	0.0	0.01
Dataset B	0.02	0.045	0.02	0.045	0.01	0.035
Dataset C	0.027	0.04	0.027	0.04	0.03	0.047

Table 2: Misclassification error of the different classifiers on datasets A,B and C.

regression yield the best results on dataset C. All classifiers have exactly the same error on dataset A.

In general, error is higher on dataset C than in dataset B, and higher in dataset B than in dataset A. This seems logic when looking at the data: the two distribution (for positive and negative labels) are more entangled in dataset C than in dataset B, etc, from where any classifier resulting in a linear decision boundary will see its error rate increase from one dataset to the other.

The LDA and Linear Regression yield very similar results in term of error, whereas logistic regression yields different results from these two methods.

The error on the test sets is generally higher than in the training sets, which seems coherent with intuition as the model has been specifically trained to minimize the error on the training sets but not on the test ones, where the samples differ.

5- QDA model

For the QDA model, the MLE is computed similarly to the computation of LDA's MLE. The log-likelyhood is now:

$$L_{(X_i,Y_i)}(\pi,\mu,\Sigma) = \sum_{i=1}^{n} log(y_i\mu + (1-y_i)(1-\mu)) + log(\frac{1}{2\Pi|\Sigma|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_{y_i})^T\Sigma_{y_i}^{-1}(x_i-\mu_{y_i})})$$

The computation is exactly similar for π and (μ_0, μ_1) as the first order conditions of optimality do not involve Σ.
The computation of (Σ_0, Σ_1) writes:

$$\frac{\partial L}{\partial \Sigma_0^{-1}} = -\frac{1}{2}\sum_{i:y_i=0}((x_i-\mu_0)^T(x_i-\mu_0)) + \frac{n_0}{2}\Sigma_0$$
$$\frac{\partial L}{\partial \Sigma_0^{-1}} = 0 \Leftrightarrow \Sigma_0 = \frac{1}{n_0}\sum_{i:y_i=0}((x_i-\mu_0)^T(x_i-\mu_0))$$
$$\frac{\partial L}{\partial \Sigma_1^{-1}} = -\frac{1}{2}\sum_{i:y_i=1}((x_i-\mu_1)^T(x_i-\mu_1)) + \frac{n_1}{2}\Sigma_1$$
$$\frac{\partial L}{\partial \Sigma_1^{-1}} = 0 \Leftrightarrow \Sigma_1 = \frac{1}{n_1}\sum_{i:y_i=1}((x_i-\mu_1)^T(x_i-\mu_1))$$

We can then compute $p(y=1|x)$:

$$\begin{aligned} p(y=1|x) &= \frac{\pi \frac{1}{2\Pi|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}}{\pi \frac{1}{2\Pi|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)} + (1-\pi)\frac{1}{2\Pi|\Sigma_0|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T\Sigma_0^{-1}(x-\mu_1)}} \\ &= \frac{1}{1+e^{f(x)}} \\ \text{Where } \quad f(x) &= \frac{1}{2}((x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) - (x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)) + log(\frac{(1-\pi|\Sigma_1|}{\pi|\Sigma_0|}) \end{aligned}$$

Thus the decision boundary $p(y|x) = 0.5$ corresponds to $f(x) = 0$ where $f(x)$ is a polynomial expression in x, and can be computed by solving:

$$x^T(\frac{\Sigma_1^{-1}-\Sigma_0^{-1}}{2})x + x^T(\Sigma_0^{-1}\mu_0 - \Sigma_1^{-1}\mu_1) + \frac{\mu_1\Sigma_1^{-1}\mu_1 - \mu_0\Sigma_0^{-1}\mu_0}{2} + log(\frac{(1-\pi)|\Sigma_1|}{\pi|\Sigma_0|}) = 0$$

a)

The learned values of the parameters (denoted A for the quadratic term ($\frac{\Sigma_1^{-1}-\Sigma_0^{-1}}{2}$), B for the linear one ($\Sigma_0^{-1}\mu_0 - \Sigma_1^{-1}\mu_1$) and c for the constant term that remains) can be found in the table 3:

	A	B	c
Dataset A	[[-0.380, 0.011],[0.011, -0.160]]	[6.069, 8.900]	-87.236
Dataset B	[[-0.386, -0.086],[-0.086, -0.226]]	[8.533, 9.339]	-94.411
Dataset C	[[0.023, -0.242],[-0.242, 0.094]]	[2.898, 7.010]	-54.772

Table 3: Learned parameters of the QDA classifier on datasets A,B and C.

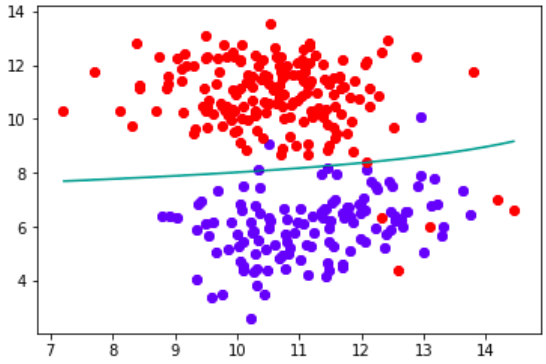

Figure 4: Line $p(y|x) = 0.5$ for the QDA classifier on dataset C

b)

The line $p(y|x) = 0.5$ is shown on Figure 4.

c)

The misclassification error on the different datasets can be found in the table 4:

	QDA	
	Train	Test
Dataset A	0.0	0.01
Dataset B	0.015	0.03
Dataset C	0.27	0.037

Table 4: Misclassification error of the QDA classifier on datasets A,B and C.

d)

As one would expect, the QDA classifier yields better classification results on the train dataset than on the test dataset (for the same reason as the other classifier, see question 4). It yields results similar to those of the other classifiers on dataset 1, but improved results on datasets B and C (for the test data).
This can be interpreted as the fact that the set of decision functions it leads to is more expressive (the set of quadratic functions, instead of only the linear ones), which makes it able to fit more closely the data without overfitting.