1 Linear classification

1.1 Generative Model (LDA)

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

D'après les hypothèses, on a :

$$p(y) = \pi^y(1-\pi)^y \text{ et } p(x|y=i) = \frac{1}{(2\pi)^{N/2}(\det\Sigma)^{N/2}}\exp(-\frac{1}{2}(x-\mu_i)^T\Sigma^{-1}(x-\mu_i))$$

D'après le théorème de Bayes : $p(y|x) \propto p(y)p(x|y)$.
Donc en introduisant C une constante et $z_i = \mathbf{1}_{\mathbf{x_i}\in\mathcal{C}_\mathbf{1}}$ la variable indiquant si la donnée x_i appartient au cluster 1 (ou bien $z_{ik} = \mathbf{1}_{x_i\in\mathcal{C}_k}$, pour $k \in \{0,1\}$), la vraisemblance du modèle s'écrit :

$$L_{(x_i,z_i)}(\pi,\mu,\Sigma) = C \times \prod_{i=1}^{N}\prod_{k\in\{0,1\}} \pi^{z_i}(1-\pi)^{(1-z_i)} \times \frac{1}{(2\pi)^{1/2}(\det\Sigma)^{1/2}}\exp(-\frac{1}{2}z_{ik}(x_i-\mu_k)^T\Sigma^{-1}(x_i-\mu_k))$$

$$\log L_{(x_i,z_i)}(\pi,\mu,\Sigma) = \sum_{i=1}^{N}\sum_{k\in\{0,1\}} z_i\log(\pi) + (1-z_i)\log(1-\pi) - \frac{N}{2}\log(\det(\Sigma)) - \frac{z_{ik}}{2}(x_i-\mu_k)^T\Sigma^{-1}(x_i-\mu_k) + C_1$$

où C_1 est une constante.

Maintenant, déterminons les estimateurs du maximum de vraisemblance de π,μ et Σ.

$$\begin{cases} \frac{\partial \log L_{(x_i,z_i)}}{\partial\pi} = \sum_{i=1}^{N}\frac{z_i}{\pi} - \frac{1-z_i}{1-\pi} = \pi^{-1}\mathbf{z}^T\mathbf{1}_N + (\mathbf{1}_N-\mathbf{z})^T(1-\pi)^{-1}\mathbf{1}_N \\ \frac{\partial \log L_{(x_i,z_i)}}{\partial\mu_k} = \sum_{i=1}^{N} z_i\Sigma^{-1T}(x_i-\mu_k) = \sum_{i=1}^{n} z_i\Sigma^{-1}(x_i-\mu_k) = \Sigma^{-1}\times(\sum_{i=1}^{n} z_i(x_i-\mu_k)) \\ \frac{\partial \log L_{(x_i,z_i)}}{\partial\Sigma_{j,l}} = \frac{N_0}{N_0}\sum_{i=1}^{N}(1-z_i)(x_i-\mu_0)^T(x_i-\mu_0) + \frac{N_1}{N_1}\sum_{i=1}^{N} z_i(x_i-\mu_1)^T(x_i-\mu_1) - N\Sigma \end{cases}$$

Comme les dérivées partielles précédentes sont toutes des fonctions convexes, on obtient leur maximum lorsque leur dérivée s'annule :

$$\begin{cases} \frac{\partial \log L_{(x_i,z_i)}}{\partial\pi} = 0 \iff \pi^{-1}\mathbf{z}^T\mathbf{1}_N - (\mathbf{1}_N-\mathbf{z})^T(1-\pi)^{-1}\mathbf{1}_N = 0 \\ \frac{\partial \log L_{(x_i,z_i)}}{\partial\mu_k} = 0 \iff \mu_k = \frac{\sum_{i=1}^{N} z_i x_i}{\sum_{i=1}^{N} z_i} \\ \frac{\partial \log L_{(x_i,z_i)}}{\partial\mu_k} = 0 \iff \Sigma = \frac{N_0}{N}\Sigma_0 + \frac{N_1}{N}\Sigma_1 \end{cases} \quad (1)$$

où :

$$\Sigma_0 = \frac{1}{N_0}\sum_{i=1}^{N}(1-z_i)(x_i-\mu_0)^T(x_i-\mu_0)$$

$$\Sigma_1 = \frac{1}{N_1}\sum_{i=1}^{N} z_i(x_i-\mu_1)^T(x_i-\mu_1)$$

Mais pour la première équation du système (1), on a :

$$\pi^{-1}\mathbf{z}^T\mathbf{1}_N = (\mathbf{1}_N-\mathbf{z})^T(1-\pi)^{-1}\mathbf{1}_N \iff \frac{N_1}{\pi} = \frac{(N-N_1)}{1-\pi}$$
$$\iff \pi = \frac{N_1}{N}$$

Finalement on obtient :

$$\begin{cases} \hat{\pi}_{ML} = \frac{\sum_{i=1}^{N} z_i}{N} \\ \hat{\mu}_{k,ML} = \frac{\sum_{i=1}^{N} z_{ik} x_i}{\sum_{i=1}^{N} z_{ik}} \\ \hat{\Sigma}_{ML} = \frac{\sum_{i=1}^{N} \sum_{k\in\{0,1\}} z_{ik}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} z_i} \end{cases}$$

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

On a d'après le théorème de Bayes,

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

Or, d'après ce qui précède :

$$\begin{cases} p(x|y=0)p(y=0) = (1-\pi)\exp(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)) + C_1 \\ p(x|y=1)p(y=1) = \pi\exp(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)) + C_2 \end{cases}$$

Donc :

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

avec :

$$\begin{cases} w = \Sigma^{-1}(\mu_1 - \mu_0) \\ b = -\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0) - \mu_0^T\Sigma^{-1}\mu_1 + \log(1-\pi) + \log(\pi) \end{cases}$$

On obtient alors :

$$p(y=1|x) = \frac{1}{1+\exp -(w^Tx+b)} = \frac{1}{1+\exp\left(-\theta^T\begin{pmatrix} x \\ 1 \end{pmatrix}\right)}$$

avec : $\theta = \begin{pmatrix} w \\ b \end{pmatrix} \in \mathbb{R}^3$

On remarque qu'il s'agit exactement de la forme de la régression logistique évalué en $\begin{pmatrix} x \\ 1 \end{pmatrix}$.

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

1.2 Logistic regression

a. Give the numerical values of the parameters learnt.

Pour implémenter l'algorithme, on utilise la méthode de Newton, car la fonction étudiée : $f(x) = w^Tx + b$ est convexe.

En effet, la loi de $Y|X = x$ suit une loi de Bernouilli de paramètre $\sigma(w^Tx+b) = \sigma(\theta^T\tilde{x})$ où σ est la fonction sigmoïd et on pose $\theta = \begin{pmatrix} w & b \end{pmatrix}$ et $\tilde{x} = \begin{pmatrix} x & \mathbf{1}_n \end{pmatrix}$ avec n le nombre d'observation (i.e. le nombre de lignes de x) et $\mathbf{1}_n$ le vecteur de taille n ne contenant que des 1. On a alors :

$$\begin{aligned} p(Y=y|X=x) &= \sigma(\theta^T\tilde{x})^y(1-\sigma(\theta^T\tilde{x}))^(1-y) \\ &= \sigma(\theta^T\tilde{x})^y\sigma(-\theta^T\tilde{x})^(1-y) \end{aligned}$$

car $1-\sigma(z) = \sigma(-z)$.

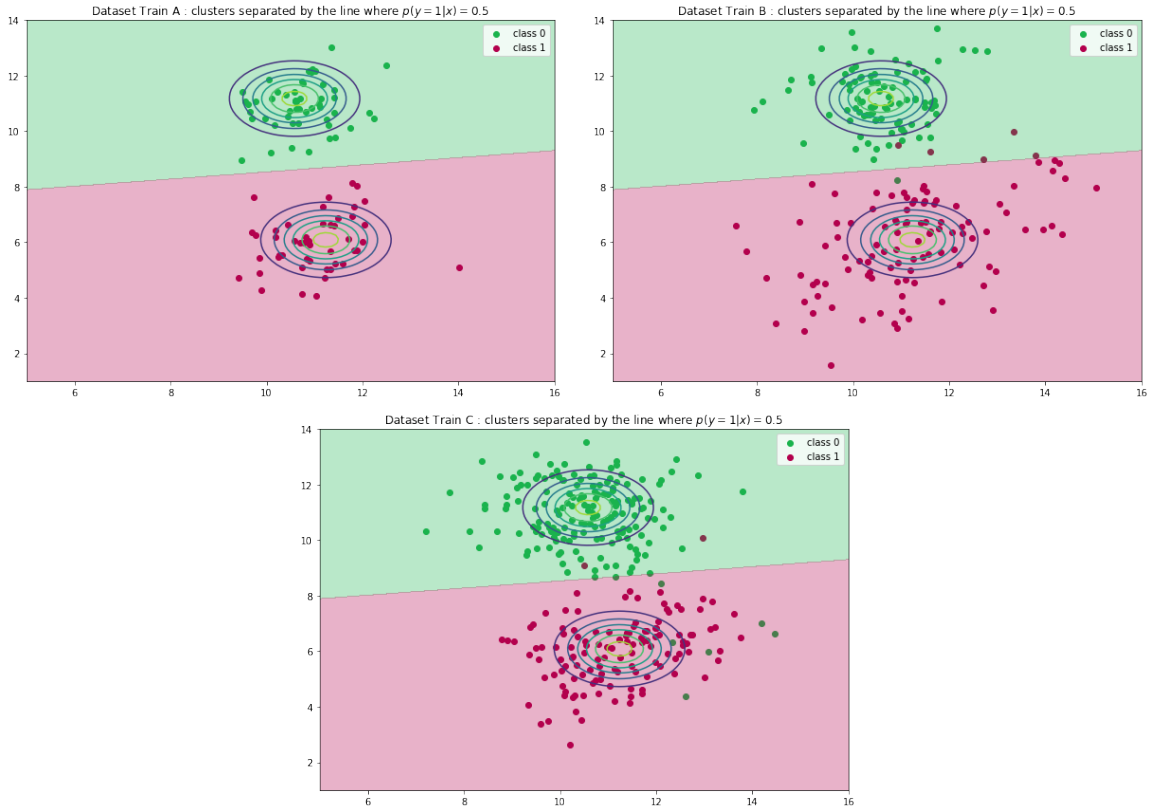

Figure 1: Modèle LDA : Résultats graphiques sur les trois datasets. (Valeurs numériques données en section 1.2.a)

On calcule maintenant la log-vraisemblance qu'on va chercher à minimiser.

$$l(w) = \sum_{i=1}^{n} y_i \log \sigma(\theta^T \tilde{x}_i) + (1 - y_i)\sigma(-\theta^T \tilde{x}_i)$$

Par composition de fonctions convexes, on peut minimiser la log-vraisemblance. Calculons son gradient.

$$\begin{aligned}
\nabla_w l(w) &= \sum_{i=1}^{n} y_i \tilde{x}_i \frac{\sigma(\theta^T \tilde{x}_i)\sigma(-\theta^T \tilde{x}_i)}{\sigma(\theta^T \tilde{x}_i)} - (1 - y_i)\tilde{x}_i \frac{\sigma(\theta^T \tilde{x}_i)\sigma(-\theta^T \tilde{x}_i)}{\sigma(-\theta^T \tilde{x}_i)} \\
&= \sum_{i=1}^{n} y_i \tilde{x}_i \sigma(-\theta^T \tilde{x}_i) - (1 - y_i)\tilde{x}_i \sigma(\theta^T \tilde{x}_i) \\
&= \sum_{i=1}^{n} y_i \tilde{x}_i (1 - \sigma(\theta^T \tilde{x}_i)) - (1 - y_i)\tilde{x}_i \sigma(\theta^T \tilde{x}_i) \\
&= \sum_{i=1}^{n} \tilde{x}_i (y_i - \sigma(\theta^T \tilde{x}_i))
\end{aligned}$$

où on a utilisé le fait que $\sigma'(z) = \sigma(z)(1 - \sigma(z)) = \sigma(z)\sigma(-z)$.

Ainsi minimiser la log-vraisemblance revient à : $\nabla_w l(w) = 0 \iff \sum_{i=1}^{n} x_i(y_i - \sigma(\theta^T \tilde{x}_i)) = 0$. Le problème est qu'il s'agit d'une équation non-linéaire, c'est pourquoi nous allons utiliser la méthode de Newton qui est un algorithme d'optimisation itératif.
On calcule maintenant la matrice hessienne de la log-vraisemblance.

$$\begin{aligned}
\nabla_w^2 l(w) &= -\sum_{i=1}^{n} \tilde{x}_i \tilde{x}_i{}^T \sigma'(\theta^T \tilde{x}_i) \\
&= -\sum_{i=1}^{n} \tilde{x}_i{}^T \sigma(\theta^T \tilde{x}_i)(1 - \sigma(\theta^T \tilde{x}_i))\tilde{x}_i \\
&= -\tilde{x}^T \text{Diag}(\sigma(\theta^T \tilde{x}_i))(1 - \sigma(\theta^T \tilde{x}_i)))\tilde{x}
\end{aligned}$$

On applique alors l'algorithme suivant (basé sur la méthode de Newton) :

Algorithm 1 Logistic regression *(Newton's method)*

procedure LOGISTIC REGRESSION(x, y, iter)
  Poser $X = \begin{pmatrix} x & \mathbf{1}_n \end{pmatrix}$ pour tenir compte de l'offset.
  Générer un vecteur $theta$ aléatoire
  for $i \in \{1, ..., iter\}$ **do**
    Calculer :

$$\begin{cases} \nabla_w l(w) = X^T(y - \sigma(\theta^T x) \\ \nabla_w^2 l(w) = X^T \text{Diag}(\sigma(\theta^T x)(1 - \sigma(\theta^T x)))X \end{cases}$$

    Mettre à jour θ :

$$\theta^{t+1} = \theta^t + (\nabla_w^2 l(w))^{-1} \nabla_w l(w)$$

  Return θ

Voici les valeurs numériques apprises par le modèle sur les trois datasets :

	Nb de points	w_1	w_2	b
TrainA	99	19.392207	-76.812105	431.976872
TrainB	199	1.842293	-3.713777	13.429778
TrainC	299	-0.276725	-1.913841	18.8018644

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

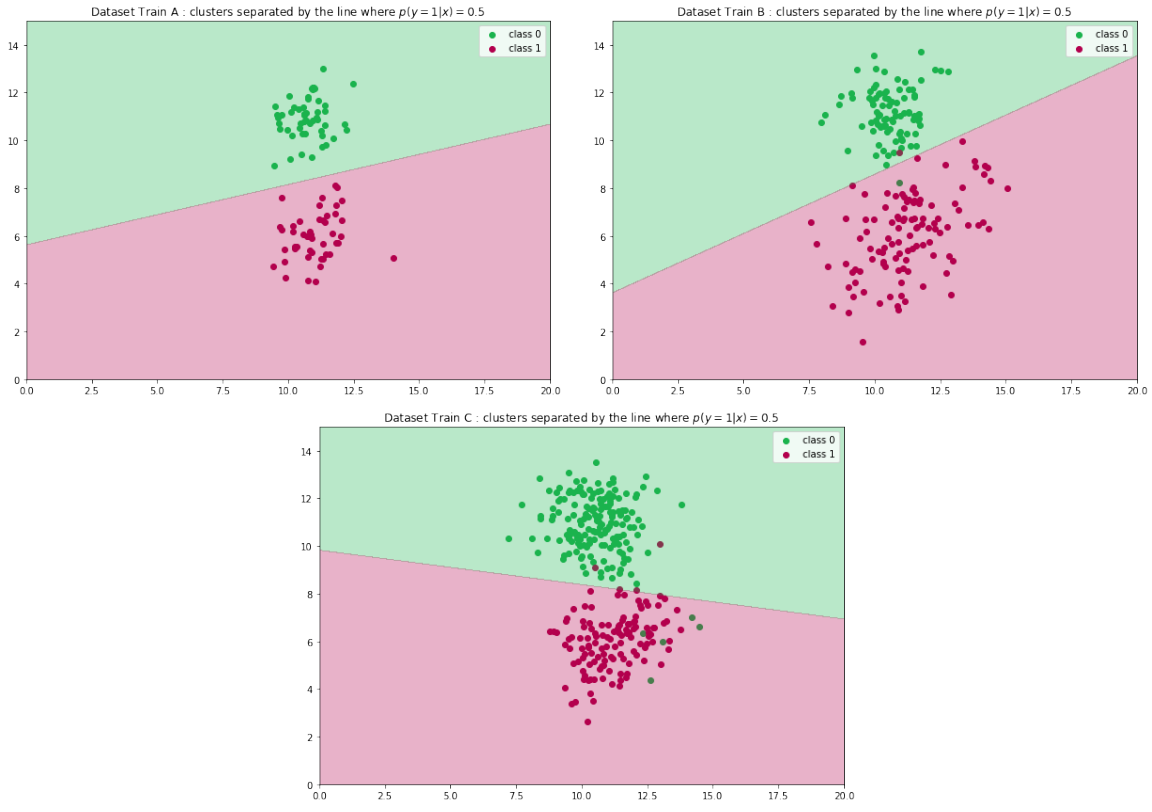

Figure 2: Logistic regression : Résultats graphiques sur les trois datasets. (Valeurs numériques données en section 1.2.a)

1.3 Linear regression

a. Give the numerical values of the parameters learnt.

Voici les valeurs numériques apprises par le modèle sur les trois datasets :

	Nb of pts	w1	w2	b
trainA	99	0.054230	-0.176174	1.398347
trainB	199	0.081954	-0.147484	0.887907
trainC	299	0.016744	-0.158977	1.640302

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

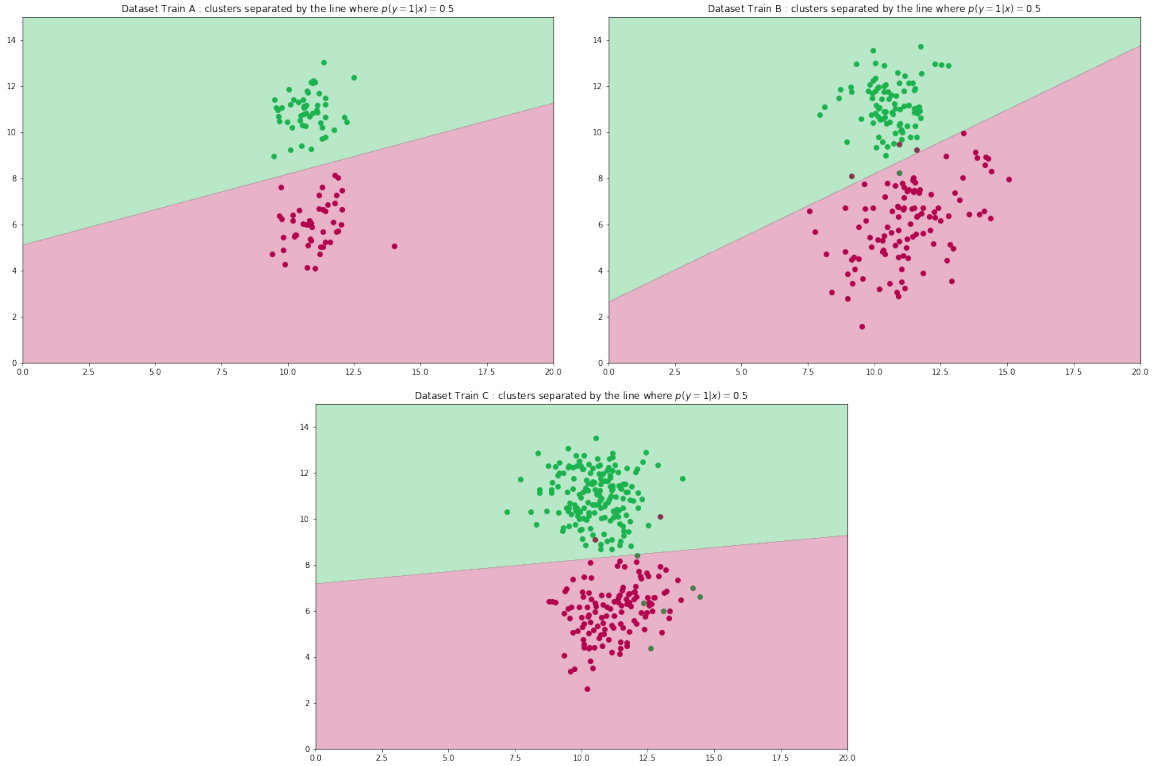

Figure 3: Regression linéaire : Résultats graphiques sur les trois datasets. (Valeurs numériques données en section 1.2.a)

1.4 Application

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.

Voici les erreurs de classification sur le dataset d'entraînement :

	Nb de pts	LDA	Logit	Linear
trainA	99	0	0	0
trainB	199	6	4	4
trainC	299	9	14	8

Voici les erreurs de classification sur le dataset de test :

	Nb de pts	LDA	Logit	Linear
testA	99	1	1	1
testB	199	8	7	9
testC	299	11	16	12

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.

On remarque qu'il y a moins d'erreurs de classification lorsque qu'on applique les modèles à des datasets de plus petites tailles. Pour le dataset *trainA*, on remarque qu'aucun des modèles ne commet d'erreur de classification, ce qui n'est pas surprenant étant donné que les deux clusters sont linéairement séparables. En revanche, lorsque le nombre de points dans les clusters augmentent et que des données des clusters sont excentrées par rapport au centre du cluster (la probabilité d'apparition de ces données augmente avec la taille du dataset), les clusters ne sont plus linéairement séparables et les algorithmes commettent des erreurs. Sur les training tests, les performances des trois algorithmes étaient plutôt similaires bien que la régression logistique commette un peu plus d'erreurs sur le dataset trainC.
De manière générale, les erreurs de classification sont un peu plus importantes sur les datasets d'entrainements que sur les datasets de validation (bien qu'elles restents du même ordre de grandeur), ce qui peut montrer un peu d'overfitting notamment pour la régression linéaire, mais qui vient aussi du fait que les données peuvent sortir de la queue de distribution liée au cluster et sont donc plus suceptibles de se mélanger avec les données de l'autre cluster, auquel cas, les clusters ne sont plus linéairement séparables.
La régression linéaire est la méthode pour laquelle il y a le plus de différences entre les erreurs sur le dataset d'entraînement et sur celui de validation. L'analyse discriminante linéaire est celle qui obtient les résultats les plus similaires entre le dataset d'entraînement et celui de validation.
L'analyse discriminante est la méthode qui obtient les meilleurs résultats sur les datasets car elle admet plus de degrés de liberté quant à l'établissement du modèle, puisque les données suivent une loi normale multivariée, qui correspond mieux aux clusters étudiés, comme ils ont des centres différents et pas toujours la même variance.

2 Gaussian mixture models and EM

2.1 Math

Write the EM algorithm for such model, giving all the exact update formula for the E and M steps. Justify (prove) the use of these updates.

On a : $p(x|\pi,\mu,\Sigma)) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i;\mu_k,\Sigma_k)$.
Donc :

$$\begin{aligned} l_{(x_i)_i}(\pi,\mu,\Sigma) &= \log L_{(x_i)_i}(\pi,\mu,\Sigma) \\ &= \sum_{i=1}^{n} \log p(x_i|\pi,\mu,\Sigma) \\ &= \sum_{i=1}^{n} \log[\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i;\mu_k,\Sigma_k)] \end{aligned}$$

On introduit une variable intermédiaire $z_{ik} = \mathbf{1}_{x_i \in \mathcal{C}_k}$ indiquant si la donnée x_i appartient au cluster k. La nouvelle log-vraisemblance est :

$$\begin{aligned} l_{(x_i|z_i)_i}(\pi,\mu,\Sigma) &= \log L_{(x_i|z_i)_i}(\pi,\mu,\Sigma) \\ &= \log p((x_i|z_i)_i|\pi,\mu,\Sigma) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log(\pi_k \mathcal{N}(x_i;\mu_k,\Sigma_k)) \end{aligned}$$

L'espérance prise contre la loi conditionnelle des z_{ik} sachant les x_i de la log-vraisemblance devient :

$$\mathbb{E}_{z_{ik}|x_i}[l_{(x_i,z_i)_i}(\pi,\mu,\Sigma)] = \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{P}(z_i = k|x_i) \log(\pi_k \mathcal{N}(x_i;\mu_k,\Sigma_k))$$

Le maximum de cette espérance se trouve en prenant le gradient par rapport aux π_k, aux μ_k et aux Σ_k et en les égalisant à 0, ce qui amène aux formules de mise à jour suivantes :

$$\begin{cases} \hat{\pi}_{k,ML} = \frac{\sum_{i=1}^{n} z_{ik}}{} \\ \hat{\mu}_{k,ML} = \frac{\sum_{i=1}^{n} z_{ik} x_i}{\sum_{i=1}^{n} z_{ik}} \\ \hat{\Sigma}_{k,ML} = \frac{\sum_{i=1}^{n} z_{ik}(x_i - \hat{\mu_k})(x_i - \hat{\mu_k})^T}{\sum_{i=1}^{n} z_{ik}} \end{cases}$$

où la mise à jour pour les Σ_k fait intervenir les identités suivantes :

$$\begin{cases} \frac{\partial \log \circ \det}{\partial \Sigma}(\Sigma) = \Sigma^{-1} \\ \frac{\partial (x-\mu)^T \Sigma^{-1}(x-\mu)}{\partial \Sigma}(\Sigma) = \frac{\partial \Sigma}{\partial \Sigma^{-1}} \frac{\partial (x-\mu)^T \Sigma^{-1}(x-\mu)}{\partial \Sigma^{-1}}(\Sigma) = -(\Sigma^{-1})^2 (x-\mu)^T (x-\mu) \end{cases}$$

Voici comment on définit l'algorithme EM pour le modèle de mélange gaussien :

Algorithm 2 Algorithme EM pour *GMM*

procedure EM
    K-Means pour initialiser les μ_i et les Σ_i
    Effectuer un nombre prédéfini d'itérations :
    E step : (π, θ) fixés et $\mathcal{L}(R((z_i)_i), \pi, \theta)$ maximisé par rapport à R. Calcul des $\mathbb{P}(z_i = k|x_i)$ pour tout i, k :

$$\mathbb{P}(z_i = k|x_i) = \frac{\pi_k \mathcal{N}(x_i; _k, \Sigma_k)}{\sum_{l=1}^{N} \pi_l \mathcal{N}(x_i; _l, \Sigma_l)}$$

    M step : R fixé et $\mathcal{L}$ maximisé par rapport à (π, θ). Calcul des nouveaux estimateurs du modèle :

$$\begin{cases} \hat{\pi}_k = \frac{\sum_{i=1}^{n} \mathbb{P}(z_i=k|x_i)}{n} \\ \hat{\mu}_k = \frac{\sum_{i=1}^{n} \mathbb{P}(z_i=k|x_i) x_i}{\sum_{i=1}^{n} \mathbb{P}(z_i=k|x_i)} \\ \hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \mathbb{P}(z_i=k|x_i)(x_i-\hat{\mu_k})(x_i-\hat{\mu_k})^T}{\sum_{i=1}^{n} \mathbb{P}(z_i=k|x_i)} \end{cases}$$

    Return $\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k$

2.2 Implementation

(cf. notebook)

2.3 Application

Les 3 clusters séparent les athlètes en 3 groupes en fonction de leurs performances respectives aux différents sports. On peut peut-être voir ces 3 clusters comme rattachant les athlètes à leur type de morphologie, certaines morphologies avantageant dans certains sports au détriment de performances moindres dans d'autres.