Examen ING2-INFO EILCO - Ingénierie Mathématique 3

Automne 2023

Nom :
Prénom :

Total: 28 points
Durée: 2h
Instructions générales: L'examen comprend 2 parties (Chacune de ces parties reprenant différentes sous-questions). Vous êtes libres de rédiger vos réponses sur des pages supplémentaires en veillant toutefois à bien indiquer le numéro de chaque question. Une fois l'examen terminé, assurez vous de bien écrire votre nom (de façon lisible) sur chacune des pages. Répondez à un maximum de questions, en commençant par les questions qui vous semblent les plus abordables.

Partie 1 (14pts)

1. [5*pts*] *Indiquer si les affirmations suivantes sont vraies ou fausses*

   Vrai / Faux *L'itération de descente de gradient pour la fonction de coût donnée par la somme des carrés des résidus peut s'écrire*

   $$\boldsymbol{\beta}_j \leftarrow \boldsymbol{\beta}_j + \eta \sum_{i=1}^{N} \left(t^{(i)} - (\beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_D x_D^{(i)}) \right) (-\tilde{x}_j^{(i)}) \quad \text{où } \tilde{\boldsymbol{x}} = [1, \boldsymbol{x}]$$

   Vrai / Faux *La différence entre la régression Ridge et la régression LASSO provient du fait que dans le cas de la régression Ridge, la pénalité sur le vecteur des coefficients de régression est donnée par le carré de la norme ℓ_2 alors que dans le cas de la régression LASSO, cette pénalité est donnée par la norme ℓ_1*

   Vrai / Faux *Dans l'approche de sélection du meilleur sous-ensemble, le nombre de modèles à tester pour des données de dimension D est* $\sum_{k=1}^{D+1} \binom{D+1}{k}$

   Vrai / Faux *La matrice $\boldsymbol{X}^T\boldsymbol{X}$ dans les équations normales doit être inversible pour assurer l'unicité du vecteur des coefficients de régression*

   Vrai / Faux *L'estimateur de Maximum de Vraissemblance est équivalent à un estimateur de Maximum à Posteriori défini avec un "a priori" uniforme*

   Vrai / Faux *La régression Lasso est efficace pour la sélection de caractéristiques car elle a tendance à annuler exactement certains coefficients, excluant ainsi ces caractéristiques du modèle*

   Vrai / Faux *Dans le cadre de la classification linéaire à K classes, l'approche "un contre un" requiert l'entrainement de $K(K+1)/2$ discriminants*

2. [*5pts*] *On dispose d'un ensemble de données $\left\{\boldsymbol{x}^{(i)}, t^{(i)}\right\}$ où $t^{(i)} \in \mathbb{R}$ et $\boldsymbol{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}) \in \mathbb{R}^2$ pour lesquelles on considère les deux modèles linéaires suivants:*

   $$\begin{array}{lll} y_1(x_1) & = \beta_1 x_1 & \text{(Modèle I)} \\ y_2(x_1, x_2) & = \gamma_1 x_1 + \gamma_2 x_2 & \text{(Modèle II)} \end{array} \tag{1}$$

Notez qu'aucun des deux modèles ne contient de biais. Les coefficients β_1, γ_1 et γ_2 sont déterminés à l'aide des données. Soit $\boldsymbol{x}_1 = (x_1^{(1)}, \ldots, x_1^{(N)})$, $\boldsymbol{x}_2 = (x_2^{(1)}, \ldots, x_2^{(N)})$ et $\boldsymbol{t} = (t^{(1)}, \ldots, t^{(N)})$. On souhaite montrer que si les caractéristiques x_1 et x_2 sont suffisamment décorrélées, alors $\beta_1 \approx \gamma_1$. Afin d'atteindre cet objectif, on procédera comme suit:

(a) [1*pt*] *Donner, à l'aide des équations normales, l'expression de $\beta_1 \in \mathbb{R}$ en fonction de $\boldsymbol{t}$ et $\boldsymbol{x}_1$*

(b) [2*pts*] *Donner, en fonction de $\boldsymbol{t}$, $\boldsymbol{x}_1$ et $\boldsymbol{x}_2$, à l'aide des équations normales, l'expression du vecteur $\boldsymbol{\gamma} = (\gamma_1, \gamma_2) \in \mathbb{R}^2$*

(c) [2*pts*] *En utilisant le fait que l'inverse d'une matrice 2×2 est donnée par*

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{2}$$

simplifier l'expression obtenue au point (b). Supposer ensuite $\boldsymbol{x}_1^T \boldsymbol{x}_2 = \varepsilon \sqrt{(\boldsymbol{x}_1^T \boldsymbol{x}_1)(\boldsymbol{x}_2^T \boldsymbol{x}_2)}$ ainsi que $\min(\boldsymbol{x}_1^T \boldsymbol{x}_1, \boldsymbol{x}_2^T \boldsymbol{x}_2) > c > 0$ ou c est une constante pour en déduire que $\gamma_1 = \frac{1}{1-\varepsilon^2}\beta_1 + O(\frac{\varepsilon}{1-\varepsilon^2})$. Pour rappel, la notation $f(x) = O(g(x))$ signifie simplement qu'il existe une constante C telle que $|f(x)| \leq C|g(x)|$.

3. [4*pts*] *On souhaite implémenter différents modèles de classification. Pour ce faire, on dispose des extraits de code 1, 2 et 3 (Figures 1 et 2 et 6).*

   (a) [2*pts*] *En vous basant sur le résultat donné à la figure 3, commencez par compléter les lignes manquantes des extraits 1 et 2 en utilisant les espaces prévus à cet effet dans le tableau 1.*

   (b) [2*pts*] *Dans un second temps, on souhaite définir et entrainer un modèle de classification sur les données représentées à la figure 4. À nouveau, vous disposez d'un extrait de code correspondant repris à la figure 6 et dont le résultat est représenté à la figure 5. Compléter les lignes manquantes en remplissant le tableau 2.*

[1]	
[2]	
[3]	
[4]	

Table 1: À compléter avec les lignes manquantes des extraits 1 et 2 (Figures 1 et 2).

[1]	
[2]	
[3]	
[4]	
[5]	
[6]	

Table 2: À compléter avec les lignes manquantes de l'extrait 3 (Figure 6).

```python
import numpy as np
from sklearn.linear_model import ??????? #[1]

X = np.vstack((data_class1, data_class2))

N0 = np.shape(data_class1)[0]
N1 = np.shape(data_class2)[0]

target = np.zeros((N0+N1,))
target[N0:] = 1

my_classifier = ??????? #[2]
my_classifier.??????? #[3]

```

Figure 1: Extrait 1.

```python
x1min = np.min(X[:,0])
x1max = np.max(X[:,0])
x2min = np.min(X[:,1])
x2max = np.max(X[:,1])

x1 = np.linspace(x1min, x1max, 500)
x2 = np.linspace(x2min, x2max, 500)
xx1, xx2 = np.meshgrid(x1, x2)

Xgrid = np.vstack((xx1.flatten(), xx2.flatten())).T

prediction = my_classifier.??????? #[4]
prediction = prediction>1/2

from matplotlib.colors import ListedColormap
cm_bright = ListedColormap(["#FF0000", "#0000FF"])

plt.contourf(xx1,xx2,prediction.reshape(np.shape(xx1)), \
             alpha=0.2, cmap = cm_bright)
plt.scatter(X[:,0], X[:,1], c = target , cmap = cm_bright)
plt.show()
```

Figure 2: Extrait 2.

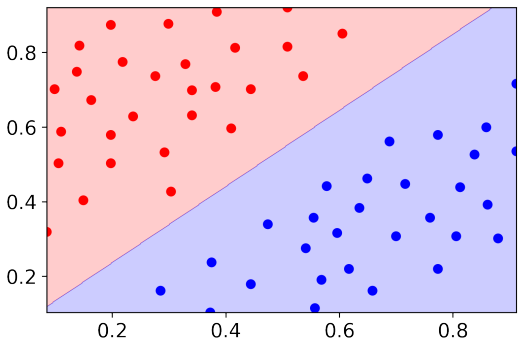

Figure 3: Résultat des extraits 1 et 2.

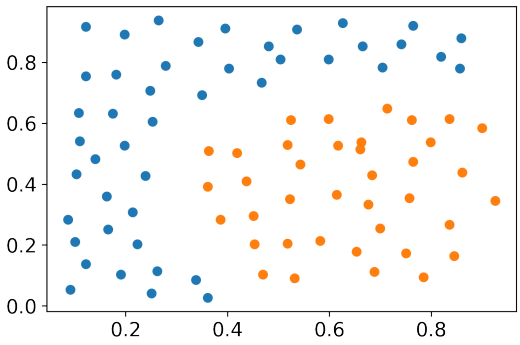

Figure 4: Données utilisées à la question 1.3b.

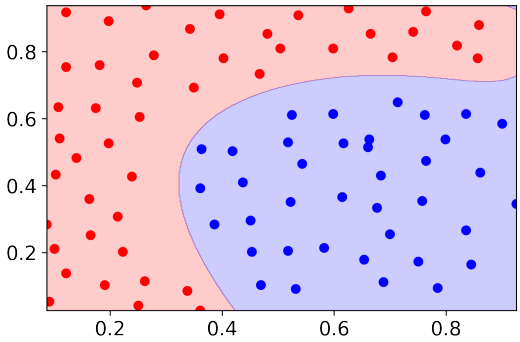

Figure 5: Résultat souhaité pour l'extrait 3 (Figure 6)

```
from sklearn.preprocessing import ??????? #[1]
from sklearn.linear_model import ??????? #[2]
from sklearn import linear_model

Xtraining = np.vstack((data_class1, data_class2))
N0 = np.shape(data_class1)[0]
N1 = np.shape(data_class2)[0]
target = np.zeros((N0+N1, ))
target[N0:] = 1

feature_transform = ??????? #[3]
Xtilde = feature_transform.??????? #[4]

# Instantiation et entrainement d'un modèle linéaire avec
# pénalité de type Ridge sur les coefficients

MyClassifier = ??????? #[5]
MyClassifier.??????? #[6]
```

Figure 6: Extrait 3.

Partie 2 (14pts)

1. [5*pts*] *Indiquer si les affirmations suivantes sont vraies ou fausses*

   Vrai / Faux *L'Analyse Discriminante Gaussienne (GDA) est un modèle génératif, ce qui signifie qu'elle modélise la distribution jointe des caractéristiques et des valeurs cibles*

   Vrai / Faux *Si un modèle est particulièrement performant sur les données d'entrainement mais peu précis sur les données de test, c'est que le modèle est vraissemblablement associé à un terme de biais élevé*

   Vrai / Faux *Lors de l'étape d'apprentissage, ajouter un terme de régularization à la fonction de coût conduira probablement à une augmentation du biais et à une diminution de la variance*

   Vrai / Faux *La fonction d'entropie binaire croisée est symétrique par rapport aux valeurs cibles et aux prédictions. I.e. intervertir les cibles et les prédictions ne change pas la valeur de la fonction.*

   Vrai / Faux *Une fois entrainé, un réseau de neurones multi-couches dont toutes les fonctions d'activation, hormis pour l'unité de sortie, sont des fonctions linéaires est équivalent à un modèle de régression logistique si la fonction d'activation de sortie est une fonction sigmoïde*

   Vrai / Faux *La dérivée de la fonction sigmoïde $\sigma(a)$ par rapport à la valeur de préactivation "a" est donnée par $\sigma(a)(1-\sigma(a))$*

   Vrai / Faux *Si l'on suppose des entrées binaires $x_1, x_2 \in \{0,1\}$, un simple preceptron de la forme $y(x) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$ où σ est la fonction de Heaviside $\sigma(x) = \begin{cases} 1 & x \geq 0 \\ 0 & sinon \end{cases}$ peut représenter n'importe laquelle des primitives booléennes* AND, OR, NAND *et* NOR
   NOR = "not" OR , NAND = "not" AND

2. [7*pts*] *On considère le réseau de neurones représenté à la figure 7. On supposera que toutes les fonctions d'activation (indiquées par la lettre 'σ' sur le diagramme) sont données par la fonction sigmoïde. Bien qu'ils ne soient pas explicitement indiqués, on supposera également que chaque neurone est muni d'un biais, i.e. la sortie de chaque neurone s'écrira donc $\sigma\left(\sum_j w_{ij}^{(\ell)} z_j^{(\ell-1)} + w_{i0}^{(\ell)}\right)$*

   (a) [*1pt*] *Représenter la fonction sigmoïde*

   (b) [*2pts*] *Donner l'expression détaillée de la fonction $y(\boldsymbol{x})$ correspondant à la sortie du réseau*

   (c) [*1pt*] *Donner l'expression de l'entropie binaire croisée $L(y, t^{(i)})$ pour une donnée d'entrainement $\left\{\boldsymbol{x}^{(i)}, t^{(i)}\right\}$.*

   (d) [*3pts*] *On souhaite utiliser l'algorithme de backpropagation pour calculer le gradient par rapport au poids $w_{11}^{(2)}$, $\frac{\partial L}{\partial w_{11}^{(2)}}$. Détaillez votre raisonnement en veillant bien à développer chaque étape.*

3. [2*pts*] *Expliquer la différence entre modèle de classification génératif et modèle de classification discriminant.*

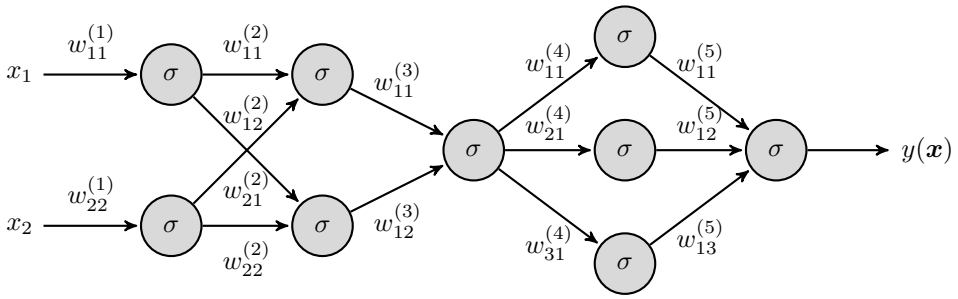

Figure 7: Réseau de neurones utilisé pour la question 1.2