Introduction to Statistical Supervised Machine Learning

Master 1 MIASHS (2023-2024)

Guillaume Metzler

Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Abstract

Le travail se décompose de 5 exercices indépendants qui reprennent les différents algorithmes vus en cours.

Les calculs peuvent se faire à l'aide de votre calculatrice mais il s'agira de détailler les calculs quand ces derniers sont demandés et ne pas uniquement donner les réponse.

Ce travail est à rendre au plus tard le **8 novembre 2023**, afin que je puisse le corriger en prévision de votre examen qui se déroulera courant décembre.

Exercice 1 : Régression Linéaire et Logistique

Régression linéaire pénalisée

Dans cette première partie, on considère un modèle linéaire de la forme

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\varepsilon},$$

où $\mathbf{y} \in \mathbb{R}^m, \mathbf{X} \in \mathscr{M}_{m,d+1}$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

Le jeu de données à notre disposition est le suivant :

y	1	4	2	4.5	2.2	4.2
x_1	-2	3	-1	1	-4	3
x_2	0	2	0	0	0	-2

On considère le problème de régression *ridge* suivant

$$\arg\min_{\boldsymbol{\theta} \in \mathbb{R}^{d+1}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{1}$$

1. Montrer que ce problème d'optimisation est convexe.
2. En déduire l'expression du paramètre optimal $\boldsymbol{\theta}$.
3. Donner la valeur de $\boldsymbol{\theta}$ à l'aide des données de l'énoncé.

On considère maintenant deux vecteurs $\boldsymbol{\theta}_1$ et $\boldsymbol{\theta}_2$ solutions respectives des problèmes suivants :

$$\arg\min_{\boldsymbol{\theta} \in \mathbb{R}^{d+1}} \frac{1}{m}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2, \quad \text{et} \quad \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^{d+1}} \frac{1}{m}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2.$$

4. Expliquer quel vecteur solution est celui qui optimise (*i.e.*, minise) le risque empirique $\mathcal{R}_S$ défini en cours.
5. Que peut-on dire sur la solution obtenue, pour le problème de régression (1), lorsque l'on remplace la norme $\|\cdot\|_2$ du terme de régularisation par une norme $\|\cdot\|_1$.

Régression polytomique

En cours, nous avons étudié la régression logistique qui est notamment utilisée pour effectuer de la classification **binaire**. Dans cette modélisation on suppose que le logarithme du rapport de deux probabilités suit un modèle linéaire, *i.e.*,

$$\ln\left(\frac{Pr[Y=1 \mid X=x]}{Pr[Y=0 \mid X=x]}\right) = \langle \boldsymbol{\theta}, \mathbf{x} \rangle.$$

Supposons maintenant que l'on dispose de plusieurs classes $\{1, 2, \ldots, q\}$ et considérons que l'on souhaite faire de la classification à l'aide de modèles logistiques. Le but est alors de construire la suite de modèles suivants :

$$\begin{aligned}
\ln\left(\frac{Pr(Y=1 \mid \mathbf{x})}{Pr(Y=q \mid \mathbf{x})}\right) &= \langle \boldsymbol{\theta}^{(1)}, \mathbf{x} \rangle, \\
\ln\left(\frac{Pr(Y=2 \mid \mathbf{x})}{Pr(Y=q \mid \mathbf{x})}\right) &= \langle \boldsymbol{\theta}^{(2)}, \mathbf{x} \rangle, \\
\cdots &= \cdots \\
\ln\left(\frac{Pr(Y=q-2 \mid \mathbf{x})}{Pr(Y=q \mid \mathbf{x})}\right) &= \langle \boldsymbol{\theta}^{(q-2)}, \mathbf{x} \rangle, \\
\ln\left(\frac{Pr(Y=q-1 \mid \mathbf{x})}{Pr(Y=q \mid \mathbf{x})}\right) &= \langle \boldsymbol{\theta}^{(q-1)}, \mathbf{x} \rangle.
\end{aligned}$$

1. A partir de l'ensemble des équations précédentes, montrer que l'on

$$Pr(Y=q \mid \mathbf{x}) = \frac{1}{1 + \sum_{k=1}^{q-1} \exp\left(\langle \boldsymbol{\theta}^{(k)}, \mathbf{x} \rangle\right)}.$$

2. En déduire que pour tout $k \in \{1, \ldots, q-1\}$

$$Pr(Y=k \mid \mathbf{x}) = \frac{\exp\left(\langle \boldsymbol{\theta}^{(k)}, \mathbf{x} \rangle\right)}{1 + \sum_{l=1}^{q-1} \exp\left(\langle \boldsymbol{\theta}^{(l)}, \mathbf{x} \rangle\right)}.$$

3. Proposer alors une règle qui permet de classification d'un individu $\mathbf{x}$ à l'aide des résultats précédemment établis.

Exercice 2 : Support Vector Machine

On considère un jeu d'entraînement $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ utilisé dans le cadre d'un problème de classification binaire. Les caractéristiques ce jeu de données sont présentées dans la

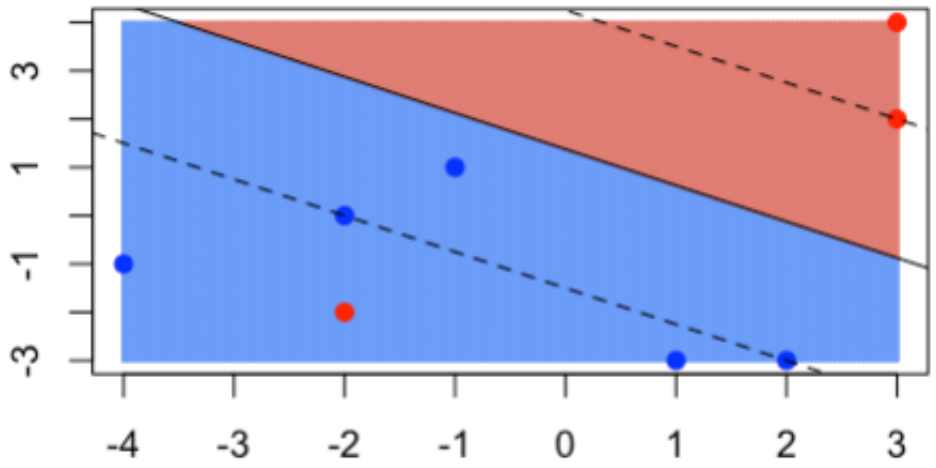

Figure 1: Classifieur SVM linéaire. La zone bleue représente la zone de prédiction négative, *i.e.* $y = -1$ et la zone rouge représente la zone de prédiction positive, *i.e.* $y = +1$.

table ci-dessous :

y	-1	1	-1	-1	-1	1	1	-1
x_1	-2	3	-1	1	-4	-2	3	2
x_2	0	2	1	-3	-1	-2	4	-3

qui a conduit à l'obtention du SVM linéaire représenté en Figure 1 et dont les paramètres sont approximativement les suivants :

$$\boldsymbol{\theta} = (-0.261, -0.348) \quad \text{et} \quad b = 0.478$$

Notre classifieur est donc de la forme $\text{sign}(\langle \boldsymbol{\theta}, \mathbf{x} \rangle + b)$

On rappelle que le SVM linéaire repose sur l'usage de la hinge loss.

1. Après avoir rappelé la définition de la hinge loss, montrer que cette dernière est convexe.
2. Sur la Figure 1, identifier :
   (a) l'hyperplan séparateur
   (b) les marges du SVM
   (c) les vecteurs (ou points) supports
3. On considère les individus

$$\mathbf{x}'_1 = (1, 0),\ \mathbf{x}'_2 = (2, 3) \quad \text{et} \quad \mathbf{x}'_3 = (-3, 2)$$

qui vérifient $y'_1 = 1,\ y'_2 = -1$ et $y'_3 = 1$. Pour ces différents points

(a) donner l'étiquette prédite par le modèle,

(b) évaluer la valeur de la loss.

Exercice 3 : Arbres de décisions

On repart du jeu de données présenté dans l'exercice précédent que l'on rappelle ci-dessous:

y	-1	1	-1	-1	1	1	1	-1
x_1	-2	3	-1	1	-4	-2	3	2
x_2	0	2	1	-3	-1	-2	4	-3

On s'intéresse ici à une tâche de classification binaire, on cherche donc à prédire la valeur de $y \in \{-1, 1\}$ en fonction des caractéristiques $\mathbf{x}$.

1. Evaluer la valeur de l'indice de Gini à la racine de l'arbre.
2. Evaluer la valeur de l'entropie à la racine de l'arbre.

On souhaite maintenant étudier deux modèles induit par deux splits différents : (i) pour le premier modèle, le premier split se fait avec la règle $x_1 < 2.5$; (ii) pour le deuxième modèle le split se fait à l'aide du critère $x_2 \geq 0$.

3. Pour les différents splits :
   (a) représenter les arbres obtenus : on indiquera quels sont les exemples que l'on trouvera dans chaque feuille
   (b) évaluer la valeur de l'indice de Gini dans les différentes feuilles
   (c) déterminer les performances de classification de ces deux arbres
   (d) calculer le gain de gini pour les deux splits. Quel est le split réellement effectué par l'arbre lors de l'apprentissage ?

Exercice 4 : Boosting

Considérons que l'on mette en place l'algorithme Adaboost avec un séparateur linéaire de la forme $h(\mathbf{x}) = \langle \boldsymbol{\theta}, \mathbf{x} \rangle + b$ de sorte à ce que ce dernier renvoie une valeur dans l'ensemble

$\{-1, 1\}$. Les hypothèses sont apprises sur le jeu de données suivant

y	-1	1	-1	-1	-1	1	1	-1
x_1	-2	-1	-1	2	-4	-2	3	1
x_2	6	2	4	-3	-1	-2	-2	1

Au cours de la première itération de notre algorithme, les paramètres du modèle sont donnés par $\boldsymbol{\theta} = (1, 1)$ et $b = -1$ et chaque exemple a un poids égal à $\frac{1}{m}$ où m désigne le nombre d'exemples.

1. Evaluer l'erreur global de votre modèle
2. En déduire le poids du modèle nouvellement appris
3. Déterminer la pondération des exemples pour la prochaine itération de l'algorithme adaboost.

On considère le modèle suivant :

$$H(\mathbf{x}) = \text{sign}(\alpha_1 h_1(\mathbf{x}) + \alpha_2 h_2(\mathbf{x}) + \alpha_3 h_3(\mathbf{x})),$$

où $(\alpha_1, \alpha_2, \alpha_3) = (2, 3, 1.5)$ et $h_1(\mathbf{x}) = \text{sign}(3x_1 + 2), h_2(\mathbf{x}) = \text{sign}(2x_1 + 3x_2 - 4)$ et $h_3(\mathbf{x}) = \text{sign}(-x_1 + 2x_2 - 1)$.

4. En reppartant de la description de l'algothime Adaboost, quel est le *weak learner* qui a l'erreur la plus faible ?
5. On considère les individus

$$\mathbf{x}'_1 = (1, 0),\ \mathbf{x}'_2 = (2, 3) \quad \text{et} \quad \mathbf{x}'_3 = (-3, 2)$$

Prédire l'étiquette de ces différents individus par votre méthode ensembliste.

Exercice 5 : Réseaux de neurones

On considère un dernier modèle se présentant sous la forme d'un réseau de neurones avec une couche cachée et deux neurones sur cette couche cachée, représenté en Figure 2.

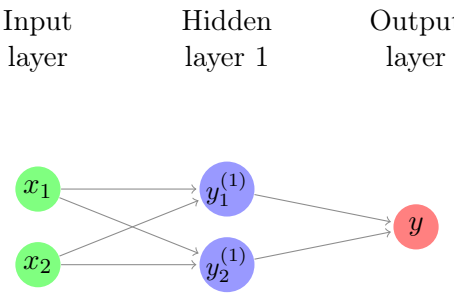

Figure 2: Représentation du neurone avec une couche cachée et deux neurones sur la couche cachée

Questions préliminaires

Dans cette partie là, on suppose que notre réseau dispose de H couches cachées comprenant chacune n_h neurones. La couche d'entrée est de taille d et la couche de sortie est de taille 1. On formule les hypothèses suivantes sur notre réseau :

- il est supposé *fully connected*, *i.e.*, les neurones d'un réseau à une couche h sont tous reliés à tous les neurones de la couche $h+1$ et réciproquement.
- on ne fera pas apparaître de terme de biais dans les différents modèles
- toutes les fonctions d'activations qui entrent en jeux sont des fonctions logistiques sauf sur la dernière couche, où il n'y a pas de fonctions d'activations.

1. Combien de paramètres se trouvent entre les couches h et $h+1$?
2. Quel est le nombre total de paramètres dans notre réseau ?

On reprend maintenant le réseau présenté en Figure 2. On s'intéresse à une tache de classification et le label sera donné par le signe de la sortie du réseau.

3. Pour différents jeux de données suivants, déterminer les poids du réseaux appropriés sur les différentes couches.

y	1	1	1	0
x_1	1	1	0	0
x_2	1	0	1	0

(a)

y	1	0	0	0
x_1	1	1	0	0
x_2	1	0	1	0

(b)

y	0	1	1	0
x_1	1	1	0	0
x_2	1	0	1	0

(c)

Mise à jour du réseau

On suppose toujours que

- il est supposé *fully connected*, *i.e.*, les neurones d'un réseau à une couche h sont tous reliés à tous les neurones de la couche $h+1$ et réciproquement.
- on ne fera pas apparaître de terme de biais dans les différents modèles
- toutes les fonctions d'activations qui entrent en jeux sont des fonctions logistiques sauf sur la dernière couche, où il n'y a pas de fonctions d'activations.

On se place dans un cadre de régression et la loss que l'on cherche à optimiser est la suivante :

$$\|y - h(\mathbf{x})\|_2^2,$$

où $h(\mathbf{x})$ désigne la sortie de notre réseau considéré et représenté en Figure 2. On suppose que tous les paramètres du modèle sont initialisés à 1 et on considère une donnée $\mathbf{x} = (1, 2)$.

1. Ecrire l'étape *forward* de la mise à jour des poids du réseau, *i.e.*, calculer la valeur des différents neurones de la couche cachée et la valeur prédite en sortie pour la donnée $\mathbf{x}$.
2. Ecrire l'étape *backward* de la mise à jour des poids du réseau, *i.e.*, les formules de mise à jour des différents poids du réseau et déterminer les nouveaux poids de ce réseau.