Introduction to Statistical Supervised Machine Learning

Devoir Maison : Correction

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.

   Pour montrer que ce problème est convexe, on va étudier la dérivée seconde. Plus précisément, on cherchera à montrer que la hessienne de ce problème est définie positive.

   Le gradient de la fonction objective est donné par

   $$\nabla_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 = 2\mathbf{X}^T + 2\lambda \mathbf{I}_{d+1} - 2\mathbf{X}^T\mathbf{y}$$

   La dérivée seconde est ainsi donnée par

   $$\nabla^2_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 = 2\mathbf{X}^T + 2\lambda \mathbf{I}_{d+1}.$$

   Or pour tout $\lambda \geq 0$, la matrice $2\lambda \mathbf{I}_{d+1}$ est semi-définie positive. Il nous reste donc à montrer que la matrice $\mathbf{X}^T\mathbf{X}$ est aussi sem-définie positive, ainsi la hessienne sera semi-définie positive comme somme de deux matrices semi-définie positive.

   Or pour tout vecteur $\mathbf{u} \in \mathbb{R}^{d+1}$, nous avons

   $$\mathbf{u}^T\mathbf{X}^T\mathbf{X}\mathbf{u} = (\mathbf{X}\mathbf{u})^T(\mathbf{X}\mathbf{u}) = \|\mathbf{X}\mathbf{u}\|_2^2 \geq 0.$$

Ce qui achève notre démonstration.

2. En déduire l'expression du paramètre optimal $\boldsymbol{\theta}$.

   Repartons du fait que

$$\nabla_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 = 2\mathbf{X}^T + 2\lambda \mathbf{I}_{d+1} - 2\mathbf{X}^T\mathbf{y}.$$

   La fonction étant convexe, la solution est donnée par la valeur de $\boldsymbol{\theta}$ qui annule le gradient précédent, *i.e.*, par la valeur de $\boldsymbol{\theta}$ telle que

$$\nabla_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 = 0.$$

   Soit

$$2\mathbf{X}^T + 2\lambda \mathbf{I}_{d+1} - 2\mathbf{X}^T\mathbf{y} = 0 \iff \boldsymbol{\theta} = \left(\mathbf{X}^T\mathbf{X} + 2\lambda \mathbf{I}_{d+1}\right)^{-1} \mathbf{X}^T\mathbf{y}.$$

3. Donner la valeur de $\boldsymbol{\theta}$ à l'aide des données de l'énoncé.

   Reprenons données de l'énoncé. Nous avons alors

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & 0 \\ 1 & 3 & 2 \\ 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & -4 & 0 \\ 1 & 3 & 0 \end{pmatrix} \quad \text{et} \quad \mathbf{y} = \begin{pmatrix} 1 \\ 4 \\ 2 \\ 4.5 \\ 2.2 \\ 4.2 \end{pmatrix}.$$

   Nous avons alors

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 8 \end{pmatrix} \quad \text{et} \quad \mathbf{X}^T\mathbf{y} = \begin{pmatrix} 17.9 \\ 16.3 \\ -0.4 \end{pmatrix}.$$

   La matrice $\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_3$ est donc inversible et son inverse est donnée par

$$\begin{pmatrix} \frac{1}{6+\lambda} & 0 & 0 \\ 0 & \frac{1}{40+\lambda} & 0 \\ 0 & 0 & \frac{1}{8+\lambda} \end{pmatrix}$$

Donc pour tout $\lambda \geq 0$, nous avons

$$\boldsymbol{\theta} = \begin{pmatrix} \frac{17.9}{6+\lambda} \\ \frac{16.3}{40+\lambda} \\ \frac{-0.4}{8+\lambda} \end{pmatrix}$$

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.

   Le premier problème d'optimisation consiste à minimiser la loss quadratique pénalisée par un terme ridge. Il y a donc un compromis à réaliser entre la minimisation du risque empirique et la complexité du modèle appris.

   Dans le second problème d'optimisation, seule la minimisation du risque empirique impliquant la loss quadratique est présente. C'est donc le vecteur solution associé à ce deuxième problème d'optimisation qui se focalise sur la minimisation du risque empirique.

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 $\|\cdot\|_1$.

   Lorsque l'on remplace un terme de pénalité ℓ_2, dit *ridge*, par un terme de pénalité ℓ_1, dit *lasso*, notre modèle tend à devenir *parcimonieux* et à effectuer de la sélection de variables,*i.e.*, seules les variables les plus importantes auront un rôle à jouer dans la tâche de prédiction.

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

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

En repartant du système d'équations précédentes, on peut écrire, de façon équivalente :

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

En sommant l'ensemble de ces équations, on a

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

En utilisant le fait que $\sum_{k=1}^{q-1} Pr(Y=k \mid \mathbf{x}) = 1 - Pr(Y=q \mid \mathbf{x})$, on trouve alors le résultat attendu.

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

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

C'est une conséquence directe de la question précédente. En effet, pour tout $k \in [\![1, q-1]\!]$, nous avons

$$Pr(Y = k \mid \mathbf{x}) = \exp\left(\langle \boldsymbol{\theta}^{(k)}, \mathbf{x} \rangle\right) Pr(Y = q \mid \mathbf{x})$$

En remplacant $Pr(Y = q \mid \mathbf{x})$ par l'expression obtenue précédemment, on trouve immédiatement le résultat énoncé.

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

$$k^\star = \underset{k \in [\![1,q]\!]}{\arg\max}\, Pr(Y = k \mid \mathbf{x}).$$

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 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.

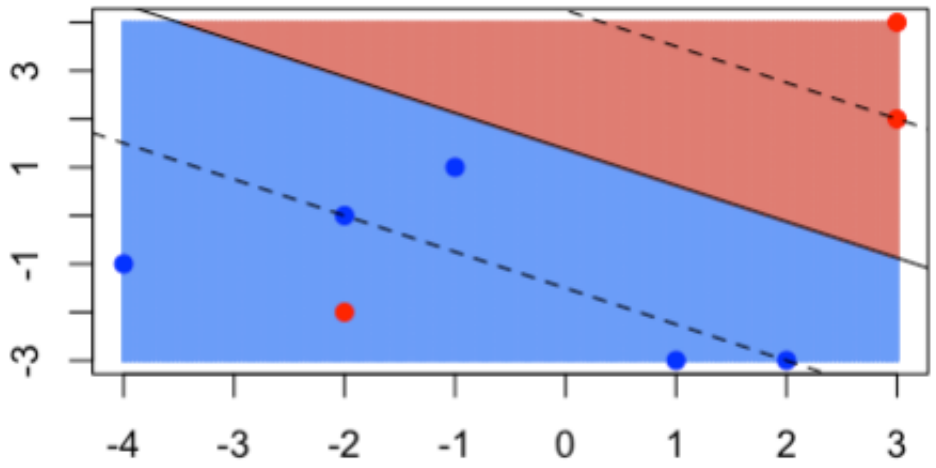

Figure 1: Classifieur SVM linéaire. La zone bleue représente la zone de prédiction négative, *i.e.* $y = -1$ et la zone rouge représente la zone de prédiction positive, *i.e.* $y = +1$.

Soit $h : \mathbb{R}^d \to \mathbb{R}$, une hypothèse qui renvoie ici un nombre réel. Considérons un exemple $(\mathbf{x}, y)$, où $\mathbf{x}$ représente le vecteur des descripteurs de l'exemple et y son étiquette. Alors la *hinge loss* est définie par

$$\ell(h(\mathbf{x}), y) = \max(0, 1 - yh(\mathbf{x})).$$

Il ne reste qu'à montrer que celle loss est convexe. Plus précisément, on va montrer que la fonction $t \mapsto \max(0, 1 - t)$ est convexe.

Les fonctions $t \mapsto 0$ et $t \mapsto 1-t$ sont deux fonctions affines, elles sont donc convexes. Montrons alors que le maximum de deux fonctions convexes est convexe et nous avons terminé.

Pour cela, considérons f et g deux fonctions convexes et $\alpha \in [0, 1]$, alors pour tout $\mathbf{x}, \mathbf{x}'$, nous avons

$$f(\alpha\mathbf{x}+(1-\alpha)\mathbf{x}') \leq \alpha f(\mathbf{x})+(1-\alpha)f(\mathbf{x}') \leq \alpha \max(f(\mathbf{x}), g(\mathbf{x}))+(1-\alpha) \max(f(\mathbf{x}'), g(\mathbf{x}')),$$

et

$$g(\alpha\mathbf{x}+(1-\alpha)\mathbf{x}') \leq \alpha f(\mathbf{x})+(1-\alpha)g(\mathbf{x}') \leq \alpha \max(f(\mathbf{x}), g(\mathbf{x}))+(1-\alpha) \max(f(\mathbf{x}'), g(\mathbf{x}')).$$

Ainsi

$$\max(f(\alpha\mathbf{x}+(1-\alpha)\mathbf{x}'), g(\alpha\mathbf{x}+(1-\alpha)\mathbf{x}')) \leq \alpha \max(f(\mathbf{x}), g(\mathbf{x}))+(1-\alpha)\max(f(\mathbf{x}'), g(\mathbf{x}'))$$

2. Sur la Figure 1, identifier :

   (a) l'hyperplan séparateur

   L'hyperplan est représenté en trait plein sur le graphique.

   (b) les marges du SVM

   Les marges sont délimitées par les droites en traits pointillés sur le graphique.

   (c) les vecteurs (ou points) supports

   Les points supports correspondent à l'ensemble des points qui sont mals classés mais aussi aux points qui se trouvent dans la marge.

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,

```
theta = c(-0.261,-0.348)
b = 0.478

x1 = c(1,0)
x2 = c(2,3)
x3 = c(-3,2)

y1 = 1
y2 = -1
y3 = 1

# Calculs des prédictions
```

```
y1pred = sign(sum(x1*theta) + b)
y2pred = sign(sum(x2*theta) + b)
y3pred = sign(sum(x3*theta) + b)

# Les étiquettes prédites sont donc

y1pred

## [1] 1

y2pred

## [1] -1

y3pred

## [1] 1
```

(b) évaluer la valeur de la loss.

```
# Les différentes loss sont respectivement égales à

max(0,1-y1*(sum(x1*theta) + b))

## [1] 0.783

max(0,1-y2*(sum(x2*theta) + b))

## [1] 0

max(0,1-y3*(sum(x3*theta) + b))

## [1] 0.435
```

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.

   Notre jeu de données comporte autant d'exemples appartenant à la classe positive que d'exemples appartenant à la classe négative. La valeur de l'indice de Gini est donc maximale et égale à 1/2.

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

   Il est en de même pour l'entropie, nous avons deux classes équireprésentées, l'entropie est docn égale à 1.
   En effet

$$\begin{aligned} H_{root} = & -p_+ \log_2(p_+) - -p_- \log_2(p_-), \\ = & -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right), \\ = & -\ln(1/2)/\ln(2), \\ = & 1 \end{aligned}$$

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

   - **Arbre 1 :** Pour le premier arbre, on retrouvera les exemples $1, 3, 4, 5, 6$ et 8 dans la feuille de gauche (exemples pour lesquels $x_1 < 2.5$) et les exemples $2 et 7$ dans la feuille de droite.
   - **Arbre 2 :** Pour le premier arbre, on retrouvera les exemples $1, 2, 3$ et 7 dans la feuille de gauche (exemples pour lesquels $x_2 \geq 0$) et les exemples $4, 5, 6$ et 8 dans la feuille de droite.

   (b) évaluer la valeur de l'indice de Gini dans les différentes feuilles

   - **Arbre 1 :** Dans la feuille de gauche, nous avons 2 exemples de la classe positive et 4 exemples de la classe négative.
     L'indice de Gini est donc égal à $2 \times \frac{2}{6} \times \frac{4}{6} = \frac{4}{9}$. L'indice de Gini est égal à 0 dans la feuille de droite car celle dernière est pure.

- **Arbre 2 :** Dans chacune des feuilles, nous avons deux exemples de la classe positive et deux exemples de la classe négative. Ainsi l'indice de Gini est égal à 1/2 dans chaque feuille.

(c) déterminer les performances de classification de ces deux arbres

- **Arbre 1 :** la feuille de gauche comporte une majorité d'exemples négatifs (4 sur 6), la prédiction est donc négative pour l'ensemble des exemples de cette feuille. La feuille de droite ne comporte que des exemples positifs, ce qui donne une prédiction positive pour les exemples qui tombent dans cette feuille.
  Ainsi, seuls deux exemples sont mal classés par cet arbre, ce qui donne une performance de 3/4.
- **Arbre 2 :** ce deuxième arbre n'est rien d'autre que la classifieur aléatoire, il a donc une performance égale à 1/2.

(d) calculer le gain de gini pour les deux splits. Quel est le split réellement effectué par l'arbre lors de l'apprentissage ?

- **Arbre 1 :** on rappelle que la gain Γ est donné par la relation

$$\Gamma = G_{root} - \left(\frac{N_L}{N_{root}} G_L + \frac{N_R}{N_{root}} G_R \right).$$

  D'où

$$\Gamma = \frac{1}{2} - \left(\frac{6}{8} \times \frac{4}{9} + \frac{2}{8} \times 0 \right) = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}.$$

- **Arbre 2 :** L'indice de Gini n'a pas évolué entre la racine et les deux feuilles, le gain est donc nul.

On va donc privilégier l'arbre 1 pour lequel le gain de Gini est maximal.

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

   Un exemple sera prédit comme appartenant à la classe positive si $h(\mathbf{x}) \geq 0$, et appartenant à la classe négative dans le cas contraire.

   Avec cette règle, on obtient une performance égale à 1/2.

2. En déduire le poids du modèle nouvellement appris

   Le poids du classifieur ainsi après, qui n'est rien d'autre que le classifieur aléatoire, est égale à

   $$\alpha = \frac{1}{2} \ln \left(\frac{1 - 0.5}{0.5} \right) = \frac{1}{2} \ln(1) = 0.$$

   Ce classifieur a donc un poids nul. Ce qui est normal vu que ses performances ne sont pas supérieures à celle du classifieur aléatoire.

3. Déterminer la pondération des exemples pour la prochaine itération de l'algorithme Adaboost.

   Comme notre classifieur a un poids nul, on en déduit directement que les exemples conservent le même poids que précédemment.

   Ce cas illustre pourquoi nous devons arrêter notre algorithme adaboost à partir du moment où la performance de l'hypothèse apprise est inférieure ou égale à 1/2. En effet, cette dernière n'implique aucune modification au niveau de la pondération des exemples. Nous serions donc amenés à apprendre toujours le même classifieur au cours des itérations.

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 repartant de la description de l'algorithme Adaboost, quel est le *weak learner* qui a l'erreur la plus faible ?

   On se content uniquement de regarder le poids α du classifieur. En effet, d'après la relation

$$\alpha = \frac{1}{2} \ln \left(\frac{1-\varepsilon}{\varepsilon} \right),$$

   le classifieur avec la pondération la plus importante est celui avec l'erreur de classification la plus faible.

   Dans le cas présent, il s'agit du classifieur h_2.

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.

   On résume tout cela dans la table ci-dessous

	$\mathbf{x}_1 = (1,0)$	$\mathbf{x}_1 = (2,3)$	$\mathbf{x}_1 = (-3,2)$
$h_1(\mathbf{x})$	$+1$	$+1$	-1
$h_2(\mathbf{x})$	-1	$+1$	-1
$h_3(\mathbf{x})$	-1	$+1$	$+1$
$H(\mathbf{x})$	-1	$+1$	-1