Apprentissage, session № 1

Master 2 SDS, Université Paris-Saclay 23/01/2024

Classification : compréhension

1. Considérons l'exécution de l'algorithme AdaBoost avec un arbre à 2 feuilles comme règle faible. Après M itérations de l'algorithme AdaBoost, quel est le nombre de paramètres à conserver (à stocker dans la mémoire) pour prédire ? Expliquer précisément la provenance de ces paramètres.

2. Dans un AdaBoost, arrêteriez-vous l'itération dans les cas suivants ? Justifiez votre réponse en deux phrases maximum par proposition.

   a. Le taux de mauvais classement de la règle de classification combinée sur les données d'apprentissage originales est de 0.

   b. Le taux de mauvais classement de la règle faible actuelle sur les données d'apprentissage pondérées est de 0.

3. Soit une matrice X représentant un jeu de données composé de n vecteurs lignes de dimension n. On suppose que les colonnes sont linéairement indépendantes. Montrer que pour tout étiquetage binaire des lignes de la matrice X, on pourra toujours construire une de classification linéaire avec un vecteur de poids w qui sépare les points. Supposons que le classificateur ait la forme sign$(w \cdot x)$. **Noter qu'une matrice carrée composée de lignes ou colonnes linéairement indépendantes est inversible.**

4. Construire un jeu de données de classification unidimensionnelle (x est de dimension 1) pour lequel l'erreur de validation croisée *Leave-One-Out* de l'algorithme des 1-nn est toujours 1. En d'autres termes, la règle de classification par 1-nn ne prédit jamais correctement le point retenu.

5. De manière générique, une règle de classification peut être écrite comme $H(x) = \text{sign}\big(F(x)\big)$, où $H(x) : \mathbb{R}^d \to \{-1, +1\}$ (car $y \in \{-1, +1\}$) et $F(x) : \mathbb{R}^d \to \mathbb{R}$. Pour optimiser les paramètres de la fonction F, nous avons besoin de minimiser une perte $\sum_{i=1}^n L\Big(y_i F(x_i)\Big)$ calculée sur le jeu de données d'apprentissage. Ici, L est une fonction du produit $yF(x)$. Par exemple, pour une règle de classification linéaire, nous avons $F(x) = w_0 + \sum_{j=1}^d w_j x_j$ et $yF(x) = y\big(w_0 + \sum_{j=1}^d w_j x_j\big)$. Parmi les fonctions illustrer sur la figure 1, quelles fonctions de perte sont appropriées pour la classification ? Pour celles qui ne sont pas appropriées, expliquez pourquoi. En général, quelles conditions doit-elle satisfaire pour être une fonction de perte appropriée ?

6. Parmi les fonctions de perte appropriées à la classification sur la figure 1, quelle est la fonction la plus robuste aux valeurs aberrantes ? Justifier.

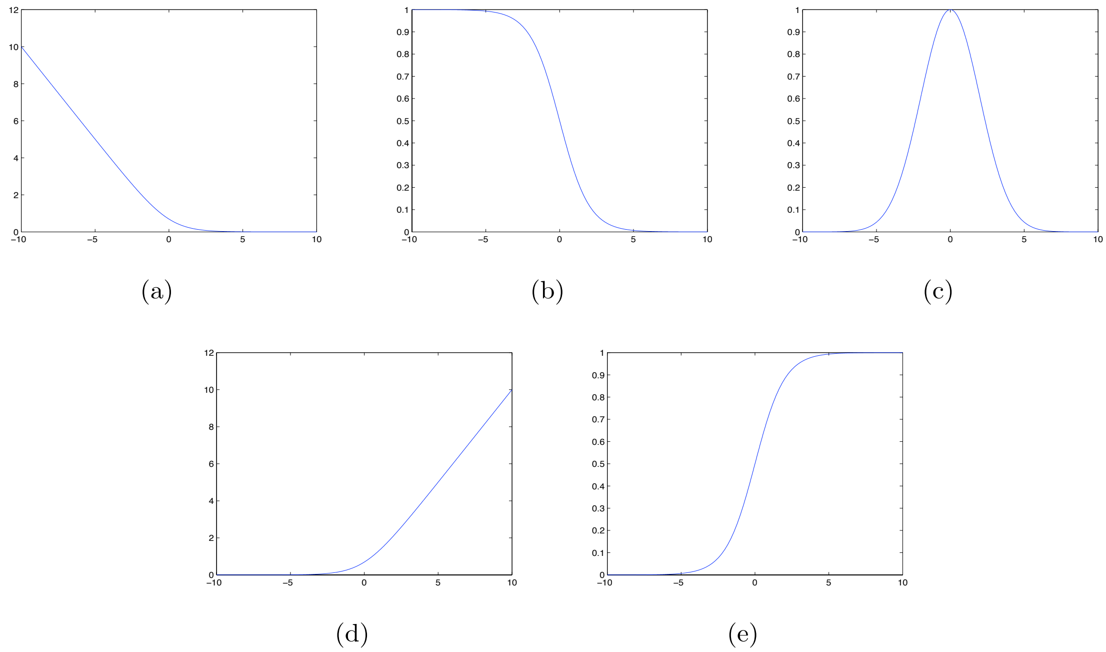

Figure 1: L'axe des x représente $yF(x)$, et l'axe des y représente $L(yF(x))$.

7. Soit $F(x) = w_0 + \sum_{j=1}^{d} w_j x_j$ et $L\big(yF(x)\big) = \frac{1}{1+\exp\big(yF(x)\big)}$. Supposons qu'on souhaite mettre en place une décente du gradient pour calculer les valeurs optimales de w_0 et des w_j. Décrire les équations de mise à jour des paramètres à chaque itération de la décente du gradient avec un pas η.

Arbres de décision et agrégation d'arbres

A. Proposer un arbre de profondeur minimale de classer parfaitement le jeu de données de la figure 2.

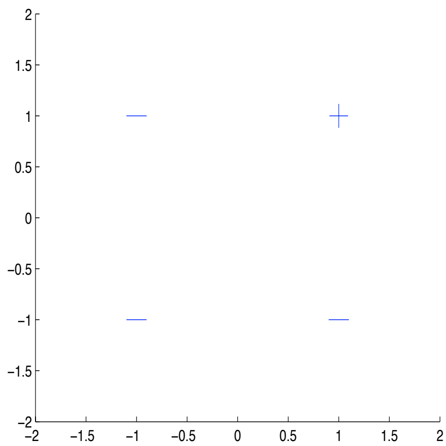

Figure 2: Un exemple de jeu de données avec deux variables x_1 et x_2.

B. Une règle de classification combinée $H_M(x)$ (obtenue par AdaBoost par exemple) est donnée par

$$H_M(x) = \text{sign}\Big(\sum_{m=1}^{M} \alpha_m h_m(x)\Big),$$

où h_m est une règle de classification faible ajustée à l'itération m. Décrire une règle $H_2(x)$ avec 2 règles faibles qui classe parfaitement les données de la figure 2. Les deux règles faibles doivent être des arbres à deux feuilles.

C. Proposer un arbre de profondeur minimale pour classer parfaitement le jeu de données de la figure 3.

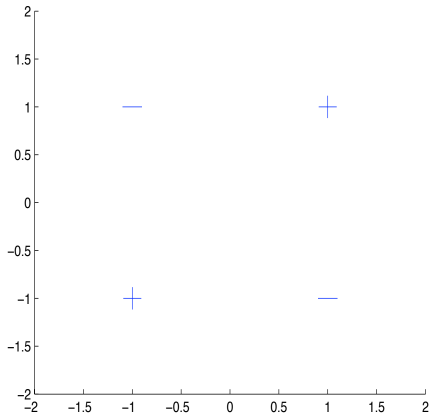

Figure 3: Un exemple de jeu de données avec deux variables x_1 et x_2.

D. Supposons que nous souhaitons utiliser les 4 règles faibles indiquées (la flèche de chaque règle faible indique la région correspondante à la classe $+$) sur la figure 4 pour construire une règle de classification combinée $H_4(x)$. Montrer qu'il n'y a pas de poids $\alpha_1, \alpha_2, \alpha_3$ et α_4 qui permettent à $H_4(x)$ de classer parfaitement les données.

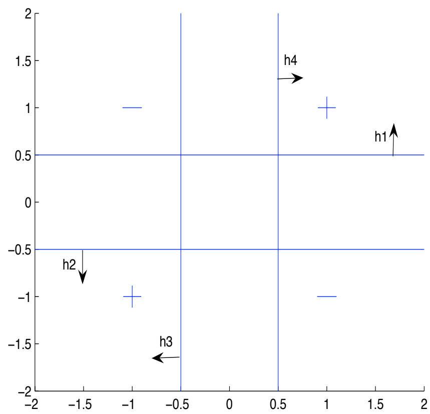

Figure 4: Un exemple de jeu de données avec deux variables x_1 et x_2. Les règles faibles $h_1, \ldots, h_4$ sont des arbres à deux feuilles .

E. Supposons que le jeu de données est composé d'un vecteur de variables explicatives binaires $x \in \{0,1\}^d$ et chaque étiquette $y \in \{-1,+1\}$ (le vrai label) correspond au vote majoritaire des variables explicatives *i.e* $y = \text{sign}\Big(\sum_{j=1}^{d}(2x_j - 1)\Big)$ où x_j est la j ième composante du vecteur de variables explicatives x. Décrire un arbre de décision de profondeur minimale pour classer parfaitement les données. Combien de feuilles a-t-il ?

F. Proposer une règle de classification combinée avec un nombre minimal de règles de classification faibles pour classer le jeu de données de la question précédente.