STA 212 **(à rendre le : 30/04/2023)**

Devoir

Enseignant: Mohammed Sedki *pageweb:* masedki.github.io

Instructions : le rendu est à envoyer par mail sous forme d'un pdf (`prenom_nom.pdf`) généré par un fichier Rmarkdown. Le pdf doit impérativement inclure les sorties (résultats d'exécution) du code implémenté. Vous pouvez travailler au plus en binôme (rendre une seule copie `prenom1_nom1_prenom2_nom2.pdf`).

Exercice 1 : Modélisation probabiliste

(4 points points)

Nous avons vu en cours un résultat d'optimalité de la règle de classification de Bayes en classification binaire pour la perte binaire. Rappelons que la règle de Bayes est donnée par

$$g^*(x) = \begin{cases} 1 & \text{si} \quad \eta(x) \geq \frac{1}{2} \\ 0 & \text{sinon,} \end{cases}$$

où $\eta(x) = \mathbb{P}(Y = 1|X = x)$. La dépendance de la loi conditionnelle de Y sachant $X = x$ rend cette règle inaccessible. Nous avons proposé en cours de mimer la règle de Bayes en approchant η par un estimateur $\widehat{\eta}$ à partir des données (régression logistique, arbre de classification, k-plus-proches-voisins etc). La règle de classification estimée est donnée par

$$\widehat{g}_n(x) = \begin{cases} 1 & \text{si} \quad \widehat{\eta}(x) \geq \frac{1}{2} \\ 0 & \text{sinon,} \end{cases}$$

En régression logistique, l'estimateur $\widehat{\eta}$ de η est obtenue en remplaçant les paramètres de régression par l'estimateur de maximum de vraisemblance dans la fonction de lien logistique. Notons $\mathcal{R}(g)$ le risque associé à la fonction de perte binaire de la règle de classification g donné par

$$\mathcal{R}(g) = \mathbb{E}\Big[\mathbb{1}\left\{g(X) \neq Y\right\}\Big] = \mathbb{P}\big(g(X) \neq Y\big)$$

(a) Montrer que

$$\mathcal{R}\big(\widehat{g}_n\big) - \mathcal{R}\big(g^*\big) \leq 2\mathbb{E}\Big[\big|\widehat{\eta}(X) - \eta(X)\big|\Big].$$

(b) Interpréter le résultat précédent.

Exercice 2 : Classification multi-classes

(6 points points)

Supposons maintenant que la variable réponse Y est à valeurs dans un ensemble fini dénombrable $\mathcal{Y} = \{1, 2, \ldots, K\}$ où K est le nombre de classes. On note g une règle de classification à valeurs dans $\mathcal{Y}$(une application de $\mathcal{X} \mapsto \mathcal{Y}$).

(a) Pour la perte binaire $\ell\big(g(x), y\big) = \mathbb{1}\{g(x) \neq y\}$, montrer que la règle de classification de Bayes est donnée par

$$g^*(x) = \underset{j \in \mathcal{Y}}{\operatorname{argmax}}\, \eta_j(x), \quad \text{où} \quad \eta_j(x) = \mathbb{P}\big(Y = j \mid X = x\big).$$

(b) Pour toute règle de classification $g : \mapsto \mathcal{Y}$, vérifier que

$$\mathcal{R}\big(g\big) - \mathcal{R}\big(g^*\big) = \mathbb{E}\Big[\max_{j \in \mathcal{Y}} \eta_j(X) - \eta_{g(X)}(X)\Big].$$

(c) Soit $\widehat{\eta}_j$ un estimateur de η_j à partir d'un jeu de données (n réalisations i.i.d du couple (X, Y)). On définit $\widehat{g}_n$ la règle de classification obtenue par substitution, *i.e.* $\widehat{g}_n(x) = \underset{j \in \mathcal{Y}}{\operatorname{argmax}}\, \widehat{\eta}_j(x)$. Montrer que

$$\mathcal{R}\big(\widehat{g}_n\big) - \mathcal{R}\big(g^*\big) \leq 2\mathbb{E}\Big[\max_{j\in\mathcal{Y}} \big|\widehat{\eta}_j(X) - \eta_j(X)\big|\Big].$$

Exercice 3 : Implémentation d'un perceptron (aux origines des SVM) (10 points points)

L'algorithme dit *perceptron* permet de construire des fonctions de seuil linéaires

$$\mathcal{F} = \Big\{\operatorname{sign}\big[f_\theta(\cdot)\big] \mid f_\theta(x) = \langle\theta, x\rangle, \theta \in \mathbb{R}^d\Big\}.$$

Soient un échantillon $\mathcal{D}_n = \big\{(x_i, y_i) \in \mathcal{X} \times \{-1, +1\}, i = 1, \ldots, n\big\}$ et un vecteur de poids $\theta \in \mathbb{R}^d$, on définit l'ensemble des erreurs $m(\theta) = \big\{i \in \{1, \ldots, n\} : y_i\langle\theta, x_i\rangle < 0\big\}$, l'algorithme *perceptron* génère une suite de vecteur $\theta^0, \theta^1, \theta^3, \ldots$

1. Initialiser $\theta^0 = 0$.
2. Pour $t = 0, 1, 2, \ldots$, tant que $m\big(\theta^t\big) \neq \emptyset$, choisir au hasard un élément $i \in m\big(\theta^t\big)$ et mettre à jour $\theta^{t+1} = \theta^t + y_i x_i$.

Définition Un jeu de données est linéairement séparable s'il existe $\theta \in \mathbb{R}^d$ tel que $m\big(\theta\big) = \emptyset$, *i.e.* il existe une droite qui sépare les deux classes sans erreurs de classement.

(a) Montrer que pour tout jeu de données linéairement séparable $\mathcal{D}_n$, l'algorithme du *perceptron* nécessite au maximum $T = \frac{R^2}{\delta^2}$ itérations pour achever la recherche de la droite séparatrice où

(i) $R = \max\limits_{i=1,\ldots,n} \|x_i\|_2$.

(ii) $\delta = \min\limits_{i=1,\ldots,n} \Big(\frac{y_i\langle\theta^*, x_i\rangle}{\|\theta^*\|_2}\Big) > 0$ pour un certain θ^*, $m\big(\theta^*\big) = \emptyset$ où δ est appelé la marge (la distance entre la droite θ^* et le point le plus proche du jeu de données).

(b) Implémentez sous R l'algorithme du perceptron linéaire décrit ci-dessus, en utilisant l'initialisation $\theta^0 = 0$. Appliquez-le au jeu de données `X_y.rda`. Comparez le nombre d'itérations requises à la borne théorique précédente. (Vous pouvez utiliser le point fixe θ^* que vous obtenez pour estimer la marge δ).