STA 212 **(à rendre le : 04/05/2022 à 23h59)**

Examen TP

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 absolument contenir les résultats d'exécution du code implémenté. Vous pouvez travailler au plus en binôme (rendre une seule copie `prenom1_nom1_prenom2_nom2.pdf`).

Exercice 1 : Sélection de modèle pour l'ajustement de courbes (5 points points)

On s'intéresse à l'ajustement de courbes à des données bruitées où on suppose l'existence d'un lien polynomial entre une variable réponse $y \in \mathbb{R}$ et une variable explicative $x \in \mathbb{R}$, donné par

$$y = \sum_{k=1}^{d} \beta_k x^k + \varepsilon,$$

où $\varepsilon \sim \mathcal{N}(0,1)$ est un bruit gaussien. Une question de sélection de modèle que nous explorons dans ce problème, est celle du choix du degré d approprié de cet ajustement polynomial.

(a) Le Fichier XYmodel.rda contient un échantillon $\{x_i, y_i\}_{i=1}^n$ avec $n = 100$. Pour $d = 1, 2, \ldots, 10$, ajuster le modèle précédent en minimisant la perte des moindres carrés

$$L(\beta) = \frac{1}{2n} \sum_{i=1}^{n} \left[y_i - \sum_{k=1}^{d} \beta_k (x_i)^k \right]^2 .$$

Quelle est la valeur de d qui minimise la perte ? Tracer sur la même figure, le jeu de données sous forme d'un nuage de points ainsi que les modèles ajustés pour $d = 1, 2, 3, 4$.

(b) Choisir la valeur optimale de $\widehat{d}$ à l'aide du critère AIC revient à minimiser $L\big(\widehat{\beta}[d]\big) + \frac{d}{n}$, où $\widehat{\beta}[d]$ est le vecteur de paramètres ajustés au modèle polynomial de degré d. Calculer le critère AIC pour les différentes valeurs de $d \in \{1, 2, \ldots, 10\}$. Quelle est la valeur optimale $\widehat{d}$?

(c) On note $\widehat{\beta} = \widehat{\beta}\big[\widehat{d}\big]$ le vecteur des paramètres du modèle optimal obtenu en **(b)** et x une nouvelle observation de la variable explicative. On peut calculer la réponse prédite

$$\widehat{y} = \sum_{k=1}^{\widehat{d}} \widehat{\beta}_k x^k.$$

Le fichier XYnew.rda contient un échantillon de test $\{x_i, y_i\}_{i=1}^m$ où $m = 500$. Utiliser ce nouveau jeu de données pour générer les prédictions $\widehat{y}_i, i = 1, 2, \ldots m$ et calculer l'erreur

$$\sum_{i=1}^{m} \left(\widehat{y}_i - y_i\right)^2.$$

(d) Répéter **(c)** pour le modèle complet de paramètres $\widehat{\beta}\big[10\big]$. L'erreur de prédiction du modèle complet est-elle plus grande que l'erreur du modèle optimal ?

Exercice 2 : Implémentation d'un perceptron (aux origines des SVM) (5 points points)

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

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

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

1. Initialiser $\theta^0 = 0$.

2. Pour $t = 0, 1, 2, \dots$, 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) $R = \max\limits_{i=1,\dots,n} \|x_i\|_2$.

(ii) $\delta = \min\limits_{i=1,\dots,n} \left(\frac{y_i \langle \theta^*, x_i \rangle}{\|\theta^*\|_2} \right) > 0$ 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).