

## Devoir 2 - Partie Théorique

- Ce devoir doit être déposé sur Gradescope et peut-être être fait par équipes de maximum 3 étudiant.e.s. Vous pouvez discuter avec des étudiants d'autres groupes mais les réponses soumises par le groupe doivent être originales. A noter que nous utiliserons l'outil de détection de plagiat de Gradescope. Tous les cas suspectés de plagiat seront enregistrés et transmis à l'Université pour vérification.
- Seulement un étudiant doit soumettre les solutions et vous devez ajouter votre/vos membre(s) d'équipe sur la page de soumission de gradescope

### 1. Décomposition biais/variance [10 points]

Considérons les données générées de la manière suivante: une donnée  $x$  est échantillonnée à partir d'une distribution inconnue, et nous observons la mesure correspondante  $y$  générée d'après la formule

$$y = f(x) + \epsilon,$$

où  $f$  est une fonction déterministe inconnue et  $\epsilon \sim \mathcal{N}(0, \sigma^2)$ . Ceci définit une distribution sur les données  $x$  et mesures  $y$ , nous notons cette distribution  $p$ .

Étant donné un ensemble d'entraînement  $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$  échantillonné i.i.d. à partir de  $p$ , on définit l'hypothèse  $h_D$  qui minimise le risque empirique donné par la fonction de coût erreur quadratique. Plus précisément,

$$h_D = \arg \min_{h \in \mathcal{H}} \sum_{i=1}^n (y_i - h(x_i))^2$$

où  $\mathcal{H}$  est l'ensemble d'hypothèses (ou classe de fonction) dans lequel nous cherchons la meilleure fonction/hypothèse.

L'erreur espérée<sup>1</sup> de  $h_D$  sur un point donné  $(x', y')$  est notée  $\mathbb{E}[(h_D(x') - y')^2]$ . Deux termes importants qui peuvent être définis sont:

---

<sup>1</sup>Ici l'espérance porte sur le choix aléatoire d'un ensemble d'entraînement  $D$  de  $n$  points tirés à partir de la distribution inconnue  $p$ . Par exemple (et plus formellement) :  $\mathbb{E}[h_D(x')] = \mathbb{E}_{(x_1, y_1) \sim p} \dots \mathbb{E}_{(x_n, y_n) \sim p} \mathbb{E}[h_{\{(x_1, y_1), \dots, (x_n, y_n)\}}(x')]$ .

- Le *biais*, qui est la différence entre l'espérance de la valeur donnée par notre hypothèse en un point  $x'$  et la vraie valeur donnée par  $f(x')$ . Plus précisément,

$$biais = \mathbb{E}[h_D(x')] - f(x')$$

- La *variance*, est une mesure de la dispersion des hypothèse apprises sur des ensemble de données différents, autour de la moyenne  $\mathbb{E}[h_D(x')]$ . Plus précisément,

$$variance = \mathbb{E}[(h_D(x') - \mathbb{E}[h_D(x')])^2]$$

Montrez que l'erreur espérée pour un point donné  $(x', y')$  peut être décomposée en une somme de 3 termes:  $(biais)^2$ ,  $variance$ , et un terme de *bruit* qui implique  $\epsilon$ . Vous devez justifier toutes les étapes de dérivation.

## 2. Fonctions de transformation des données [10 points]

Dans cet exercice, vous allez concevoir des fonctions de transformation depuis l'espace de features original vers un espace où les données sont linéairement séparables. Pour les questions suivantes, si vous répondez '*oui*', écrivez l'expression de la transformation correspondante; et si votre réponse est '*non*', ajoutez une courte justification de votre réponse. Vous devez donner les formules explicites des transformations, et ces formules doivent utiliser uniquement des opérations mathématiques simples.

- (a) Soient les données 1-D suivantes (Figure 1). Pouvez-vous proposer une transformation 1-D (i.e. vers un espace de dimension 1) qui rend les points linéairement séparables?



Figure 1: Jeu de données 1D. Les points entre  $2k$  et  $2k + 1$  sont étiquetés par X. Les points entre  $2k + 1$  et  $2k + 2$  sont étiquetés par O.

- (b) Soient les données 2-D suivantes (Figure 2). Pouvez-vous proposer une transformation 1-D qui rend les points linéairement séparables?

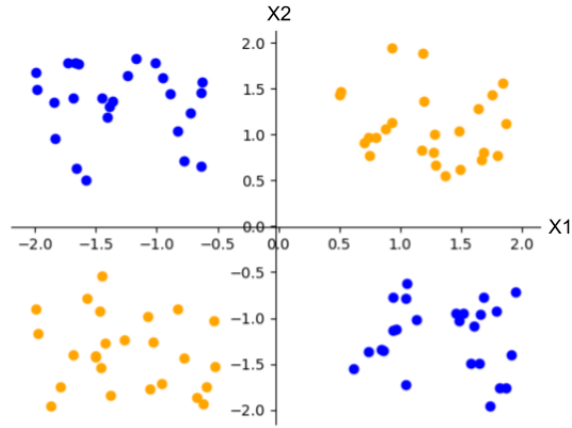


Figure 2: Jeu de données 2D.

- (c) En utilisant les idées que vous avez utilisées pour les deux questions précédentes, pouvez-vous proposer une transformation des données suivantes (Figure 3) qui les rendent linéairement séparables? Si votre réponse est ‘oui’, donnez l’expression du noyau qui correspond à la transformation proposée. Souvenez-vous que  $K(x, y) = \langle \phi(x), \phi(y) \rangle$ , donc trouvez  $\phi$  et faites le produit scalaire pour obtenir le noyau.

### 3. Dérivées et gradients [15 points]

Dans cette question, nous examinerons les dérivées et les gradients. Le Chapitre 5 de “Mathematics for Machine Learning” peut être utilisé comme référence pour cette question.

- (a) Calculer la dérivée  $f'(x)$  pour:

$$f(x) = -5 \log(x^5) \sin(x^2)$$

- (b) Calculer la dérivée  $f'(x)$  pour:

$$f(x) = 3 \exp\left(\frac{-5}{3\sigma}(x - \mu)^2\right)$$

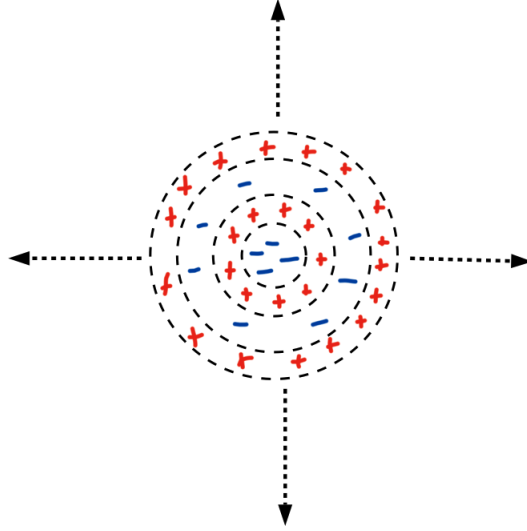


Figure 3: Un autre jeu de données 2D. Les points entre les aires de rayon  $2k$  et  $2k + 1$  sont étiquetés par  $-$ . Les points entre les aires de rayon  $2k + 1$  et  $2k + 2$  sont étiquetés par  $+$ .

où  $\sigma, \mu \in \mathbb{R}$ .

(c) Considérez les fonctions suivantes:

$$f_1(x) = \sin(2x_1) \cos(3x_2)$$

où  $x \in \mathbb{R}^2$ , et

$$f_2(x, y) = 3x^T y$$

$$f_3(x) = -4xx^T$$

où  $x, y \in \mathbb{R}^n$ .

- i. Quelles sont les dimensions de  $\frac{\partial f_1}{\partial x}$ ,  $\frac{\partial f_2}{\partial x}$  et  $\frac{\partial f_3}{\partial x}$ ?
  - ii. Calculer la matrice Jacobienne pour chacune de ces fonctions.
- (d) Calculer les dérivées  $\frac{df}{dx}$  des fonctions suivantes:
- i. Utilisez la règle de dérivée en chaîne. Donnez les dimensions de chaque dérivée.

$$f(z) = 2\exp\left(-\frac{1}{2}z^2\right)$$

$$z = g(y) = y^T S^{-1} y$$

$$y = h(x) = x - \mu$$

où  $x, \mu \in \mathbb{R}^D, S \in \mathbb{R}^{D \times D}$ .

ii.

$$f(x) = \text{tr}(xx^T + \sigma I)$$

où  $x \in \mathbb{R}^D$  and  $\text{tr}(A)$  est la trace de  $A$ .

iii. Utilisez la règle de la chaîne pour calculer les dérivées et fournir également les dimensions de la dérivée partielle pour

$$f = \tanh^2(z), \quad z = Ax + b$$

où  $f \in \mathbb{R}^M, x \in \mathbb{R}^N, A \in \mathbb{R}^{M \times N}, b \in \mathbb{R}^M$ .

(a) **Optimisation [15 points]** Soit la régression logistique à une dimension suivante:

$$\sigma(wx) = \frac{1}{1 + e^{-wx}}$$

où  $x, w \in \mathbb{R}$ , et la fonction de perte associée:

$$L(w) = -y \log \sigma(wx) - (1 - y) \log (1 - \sigma(wx))$$

(a) Montrer que la fonction de perte associée avec la régression logistique est convexe en utilisant une des définitions de convexité suivante

- $L(w)$  est convexe si et seulement si

$$\forall w_1, w_2, t \in [0, 1] : L(tw_1 + (1 - t)w_2) \leq tL(w_1) + (1 - t)L(w_2)$$

- $L(w)$  est convexe si et seulement si  $\frac{d^2 L}{dw^2}(w) \geq 0$  for all  $w$

Vous pouvez aussi utiliser une autre définition, mais vous devez alors la donner explicitement

- (b) Donnez le gradient de  $\sigma(wx)$  au point  $w$ . Quelles sont ses dimensions?
- (c) Donnez une expression analytique pour tous les points stationnaires de  $L(w)$  w.r.t  $w$ , en justifiant votre réponse.
- (d) Donnez l'expression pour une étape de descente de gradient d'un point initial  $w_0$  à point  $w_1$  après, en utilisant le gradient du fonction de perte

4. **Validation croisée "k-fold"** [bonus 10 points]

Soit  $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$  un ensemble de données échantillonné i.i.d. à partir d'une distribution inconnue  $p$ . Pour estimer le risque (erreur de test) d'un algorithme d'apprentissage en utilisant  $D$ , la validation croisée "k-fold" utilise la  $i$ -ème portion des données  $D_i = \{(x_j, y_j) \mid j \in \text{ind}[i]\}$  (où  $\text{ind}[i]$  sont les indices des points de données dans la  $i$ -ème portion) pour estimer le risque de l'hypothèse retournée par un algorithme d'apprentissage entraîné sur toutes les données sauf la  $i$ -ème portion :  $D_{\setminus i} = \{(x_j, y_j) \mid j \notin \text{ind}[i]\}$ .

Plus précisément, si on note  $h_{D_{\setminus i}}$  l'hypothèse obtenue par l'algorithme d'apprentissage entraîné sur les données  $D_{\setminus i}$ , l'erreur de validation croisée k-fold est donnée par:

$$\text{error}_{k\text{-fold}} = \frac{1}{k} \sum_{i=1}^k \frac{1}{n/k} \sum_{j \in \text{ind}[i]} l(h_{D_{\setminus i}}(x_j), y_j)$$

où  $l$  est la fonction de perte.

Dans cet exercice, nous nous intéressons à certaines des propriétés de cet estimateur

**k-fold est non biaisé**

- (a) [bonus] Rappelez la définition du risque d'une hypothèse  $h$  pour un problème de régression avec la fonction de coût erreur quadratique
- (b) [bonus] En utilisant  $D'$  pour dénoter un ensemble de données de taille  $n - \frac{n}{k}$ , montrez que

$$\mathbb{E}_{D \sim p} [\text{error}_{k\text{-fold}}] = \mathbb{E}_{\substack{D' \sim p, \\ (x, y) \sim p}} [(y - h_{D'}(x))^2]$$

où la notation  $D \sim p$  signifie que  $D$  est échantillonné i.i.d. à partir de la distribution  $p$  et où  $h_D$  est l'hypothèse obtenue par l'algorithme d'apprentissage sur les données  $D$ . Expliquez en quoi cela montre que  $\text{error}_{k\text{-fold}}$  est un estimateur (presque) non-biaisé du risque de  $h_D$ .

**Complexité de k-fold** Nous étudions maintenant la validation croisée k-fold pour la régression linéaire où les données d'entrées  $\mathbf{x}_1, \dots, \mathbf{x}_n$  sont des vecteurs à  $d$  dimensions. Nous utilisons  $\mathbf{X} \in \mathbb{R}^{n \times d}$  et  $\mathbf{y} \in \mathbb{R}^n$  pour représenter la matrice des données d'entrée et le vecteur des sorties correspondantes.

- (c) **[bonus]** En considérant que la complexité en temps pour inverser une matrice de taille  $m \times m$  est en  $\mathcal{O}(m^3)$ , quelle sera la complexité du calcul de la solution de la régression linéaire sur l'ensemble de données  $D$ ?
- (d) **[bonus]** Soient  $\mathbf{X}_{-i} \in \mathbb{R}^{(n-\frac{n}{k}) \times d}$  and  $\mathbf{y}_{-i} \in \mathbb{R}^{(n-\frac{n}{k})}$  la matrice des données d'entrées et le vecteurs des sorties obtenus en supprimant les lignes de la  $i$ -ème portion de  $\mathbf{X}$ . En utilisant la formule pour  $error_{k-fold}$  mentionnée précédemment, écrivez l'expression de l'erreur de validation croisée "k-fold" pour la régression linéaire. Quelle est la complexité algorithmique du calcul de cette formule?
- (e) **[bonus]** Dans le cas particulier de la régression linéaire, l'erreur k-fold peut être calculée de manière plus efficace. Montrez que dans le cas de la régression linéaire, on a:

$$error_{k-fold} = \frac{1}{k} \sum_{i=1}^k \left( \frac{\mathbf{y}_i - \mathbf{X}_i \mathbf{w}^*}{\mathbf{I} - \mathbf{X}_i (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}_i^\top} \right)^2$$

où  $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$  est la solution de la régression linéaire calculée sur tout l'ensemble de données  $D$ . Quelle est la complexité du calcul de cette expression?