IFT-3395 Fundamentals of Machine Learning
Prof. Ioannis Mitliagkas

Devoir 2 - Partie Théorique

- Ce devoir doit être déposé sur Gradescope et peut-être être fait en groupe jusqu'à 3 étudiants. 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.
- Un seul étudiant doit soumettre les solutions et vous devez ajouter votre membre d'équipe sur la page de soumission de Gradescope.

1. **Décomposition biais/variance** [7 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 $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ 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[1] 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:

   - 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,

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

   - 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,

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

[1] 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) : $\mathbb{E}[(h_D(x')] = \mathbb{E}_{(x_1,y_1)\sim p} \cdots \mathbb{E}_{(x_n,y_n)\sim p} \mathbb{E}[(h_{\{(x_1,y_1),\ldots,(x_n,y_n)\}}(x')]$.

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 contient ϵ. Vous devez justifier toutes les étapes de dérivation.

2. **Dérivation du gradient pour la régression logistique.** [11 points]

   Étant donné un ensemble de données (x_i, y_i) pour $i = 1, 2, \ldots, n$, où $x_i \in \mathbb{R}^d$ sont les caractéristiques d'entrée et $y_i \in \{0, 1\}$ sont les étiquettes binaires correspondantes, la fonction de perte logistique pour un seul point de données est définie comme :

$$L(w; (x_i, y_i)) = -\left[y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))\right]$$

   où :

   - $z_i = w^\top x_i$ est la combinaison linéaire des poids et des caractéristiques,
   - $\sigma(z_i) = \frac{1}{1+e^{-z_i}}$ est la fonction sigmoïde.

   La perte logistique totale sur tous les échantillons, avec régularisation L2, est donnée par :

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} \left[-y_i \log(\sigma(z_i)) - (1 - y_i) \log(1 - \sigma(z_i))\right] + \frac{\lambda}{2} \|w\|^2$$

   où λ est le paramètre de régularisation.

   (a) **Dérivation du gradient pour la régression logistique.** [5 points]

      i. Dériver le gradient de la perte logistique $L(w; (x_i, y_i))$ par rapport au vecteur de poids w.

      ii. Dériver le gradient de la fonction objectif totale $J(w)$ par rapport à w, en incluant le terme de régularisation L2. Montrer toutes les étapes de la dérivation.

   (b) **Règle de mise à jour pour la descente de gradient :** [4 points]

      i. Écrire l'expression finale pour le gradient de $J(w)$ en termes de la fonction sigmoïde $\sigma(z_i)$. En utilisant les gradients dérivés, écrire les règles de mise à jour pour effectuer la descente de gradient sur w.

   (c) **Interprétation** [2 points]

      i. Explique le rôle du paramètre de régularisation λ dans le contexte la régression logistique.

3. **Risque de Bayes** [16 points]

   Dans cet exercice, nous montrerons que le classificateur de Bayes (en supposant que nous utilisions la vraie distribution cible sous-jacente) minimise le risque réel sur tous les classificateurs possibles.

   Rappelons que le but de la classification binaire est d'apprendre une fonction f de l'espace d'entrée, $\mathcal{X}$, vers l'espace des classes, $\mathcal{Y} = \{0, 1\}$. On peut mesurer la qualité d'un classificateur f en utilisant la fonction de coût $0 - 1$; i.e.,

$$\ell(\hat{y}, y) = \mathbb{1}_{\{\hat{y} \neq y\}} = \begin{cases} 1, & \text{if } \hat{y} \neq y \\ 0, & \text{otherwise} \end{cases}$$

Rappelons que le vrai risque de f est défini par

$$R(f) = \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(f(x), y)\right]$$

où $\mathcal{P}$ est la distribution cible sous-jacente.

Habituellement, nous supposons que $\mathcal{P}$ est inconnu et nous déduisons f à partir d'un ensemble de données tirées de $\mathcal{P}$. Pour cet exercice, nous considérerons le classifieur de Bayes construit en utilisant la distribution cible $\mathcal{P}$, qui est définie par

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

où $\eta(x) \equiv P(Y = 1|X = x)$.

Vous montrerez que pour n'importe quelle fonction $g : \mathcal{X} \to \mathcal{Y}$ on a $R(g) \geq R(f^*)$

(a) Tout d'abord, montrez que $R(f) = P_{(x,y)\sim\mathcal{P}}(f(x) \neq y)$.

(b) Montrez que, pour tout $g : \mathcal{X} \to \mathcal{Y}$,

$$P(g(x) \neq y) = 1 - \left[\mathbb{1}_{\{g(x)=1\}}\eta(x) + \mathbb{1}_{\{g(x)=0\}}\left(1 - \eta(x)\right)\right]$$

(c) En utilisant la réponse à la question précédente et le fait que $\mathbb{1}_{\{g(x)=0\}} = 1 - \mathbb{1}_{\{g(x)=1\}}$, montrez que, pour toute fonction $g : \mathcal{X} \to \mathcal{Y}$,

$$P\left(g(x) \neq Y|X = x\right) - P\left(f^*(x) \neq Y|X = x\right) = (2\eta(x) - 1)\left(\mathbb{1}_{\{f^*(x)=1\}} - \mathbb{1}_{\{g(x)=1\}}\right)$$

(d) Enfin, montrez que, pour tout $g : \mathcal{X} \to \mathcal{Y}$,

$$(2\eta(x) - 1)\left(\mathbb{1}_{\{f^*(x)=1\}} - \mathbb{1}_{\{g(x)=1\}}\right) \geq 0$$

(e) Conclure.

4. **Validation croisée "leave-one-out"** [16 points]

Soit l'ensemble de données $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ échantillonné i.i.d. à partir d'une distribution inconnue p. Nous étudions la validation croisée "leave-one-out", qu'on pourrait traduire par "garder un exemple de côté", par la suite nous utiliserons la notation VCLOO. Pour rappel, la VCLOO sur un ensemble de données de taille n consiste à réaliser k validations croisées dans le cas particulier où $k = n - 1$. Pour estimer le risque (c'est-à-dire l'erreur de test) d'un algorithme d'apprentissage en utilisant les données D, VCLOO consiste à comparer chaque sortie y_i avec la prédiction effectuée à l'aide du modèle obtenu en entraînant sur toutes les données sauf l'exemple (x_i, y_i).

Plus précisément, si on note $h_{D\setminus i}$ l'hypothèse obtenue par l'algorithme d'apprentissage entraîné sur les données $D \setminus \{(x_i, y_i)\}$, l'erreur leave-one-out est donnée par:

$$\text{erreur}_{LOO} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(h_{D\setminus i}(x_i), y_i)$$

où $\mathcal{L}$ est la fonction de perte.

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

Leave-one-out est non biaisé

(a) Rappelez la définition du risque d'une hypothèse h pour un problème de régression avec la fonction de coût erreur quadratique

(b) En utilisant D' pour dénoter un ensemble de données de taille $n-1$, montrez que

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

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 erreur_{LOO} est un estimateur (presque) non-biaisé du risque de h_D.

Complexité de leave-one-out Nous étudions maintenant LOO pour la régression linéaire où les données d'entrées $\mathbf{x}_1, \ldots, \mathbf{x}_n$ 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) En considérant que la complexité en temps pour inverser une matrice de taille $m \times m$ est en $\mathcal{O}\left(m^3\right)$, quelle sera la complexité du calcul de la solution de la régression linéaire sur l'ensemble de données D?

(d) En notant $\mathbf{X}_{-i} \in \mathbb{R}^{(n-1)\times d}$ et $\mathbf{y}_{-i} \in \mathbb{R}^{(n-1)}$ la matrice des données d'entrées et le vecteurs des sorties obtenus en supprimant la ligne i de $\mathbf{X}$ et la composante i de $\mathbf{y}$, écrivez l'expression de l'erreur VCLOO pour la régression linéaire. Quelle est la complexité algorithmique du calcul de cette formule?

(e) Dans le cas particulier de la régression linéaire, l'erreur leave-one-out peut être calculée de manière plus efficace. Montrez que dans le cas de la régression linéaire, on a:

$$\text{erreur}_{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \mathbf{w}^{*\top}\mathbf{x}_i}{1 - \mathbf{x}_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i}\right)^2$$

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?