TP 9 – Régression linéaire
Année 2022-2023

L'objectif de ce TP est de manipuler les concepts abordés en cours sur la régression linéaire.

1 Quelques rappels mathématiques

1.1 Dérivées partielles

Pour les fonctions données ci-après, donnez les expressions des dérivées partielles par rapport à x, y et z (quand z apparait).

1. $$f(x,y) = 2 + 3x + 5y + 4xy + x^2y \tag{1}$$

2. $$g(x,y,z) = x(y^2+z^2) + y(x^2+y^2+z^2) - \frac{xyz}{\sqrt{x^2+y^2+z^2}} \tag{2}$$

3. $$h(x,y,z) = \sqrt{1 + x^2y^2 + xyz^2} \tag{3}$$

4. $$s(x,y) = (x^2+y)^2 + cos(xy) - sin(2\pi x^3) \tag{4}$$

1.2 Autres dérivées partielles

Pour les fonction $J(\boldsymbol{\theta})$ suivantes, donnez les expressions des dérivées partielles par rapport aux paramètres θ_i, $\boldsymbol{\theta}$ étant le vecteur des paramètres θ_i.

1. $$J(\boldsymbol{\theta}) = \theta_0 + 2\theta_1 x + \theta_0(x^2+y^2) + \theta_1 xy + \theta_0\theta_1 x^2 y \tag{5}$$

2. $$J(\boldsymbol{\theta}) = 1/2 + cos(\theta_0 x) - sin(\theta_1 x + \theta_0) \tag{6}$$

3. $$J(\boldsymbol{\theta}) = \theta_1 x^3 + \theta_0 x^2 y^2 + \theta_1 x^3 y^2 + \theta_0\theta_1 x^2 y^3 + \theta_1\sqrt{\theta_0 x.y} \tag{7}$$

4. $$J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{k=1}^{k=m}(\theta_0 + \theta_1 x^{(k)} - y^{(k)})^2 \tag{8}$$

2 Régression linéaire par méthode analytique des moindres carrés

On considère un ensemble de m données d'apprentissage (x_k, y_k) qui relient l'entrée x_k à la sortie y_k pour chaque exemple k. La régression linéaire consiste à chercher les paramètres a et b définissant la droite $y = ax + b$ qui passe au plus près de cet ensemble de points. Les paramètres a et b sont déterminés par une méthode qui minimise le coût :

$$J(a, b) = \frac{1}{2m} \sum_{k=1}^{k=m} (\hat{y}_k - y_k)^2 \tag{9}$$

$\hat{y}_k$ étant la valeur de sortie estimée, c'est-à-dire obtenue en projetant le point (x_k, y_k) verticalement sur la droite. J(a,b) peut aussi s'écrire :

$$J(a, b) = \frac{1}{2m} \sum_{k=1}^{k=m} (ax_k + b - y_k)^2 \tag{10}$$

Dans cet exercice on cherche à déterminer les paramètres a et b par une méthode analytique qui s'appelle méthode des moindres carrés décrite ci-après :

1. La méthode des moindres carrés consiste à calculer de manière explicite les paramètres a et b qui minimisent $J(a, b)$. On doit pour cela résoudre les deux équations :

   $\frac{\partial J(a,b)}{\partial a} = 0$ et $\frac{\partial J(a,b)}{\partial b} = 0$

2. Donnez les deux expressions qui résultent de ces deux équations aux dérivées partielles.

Solutions : les solutions de ce système d'équations sont données par :

$$a = \frac{m \sum_{k=1}^{k=m} x_k y_k - \sum_{k=1}^{m} x_k \sum_{k=1}^{m} y_k}{m \sum_{k=1}^{m} x_k^2 - \left(\sum_{k=1}^{m} x_k\right)^2} \tag{11}$$

$$b = \frac{\sum_{k=1}^{m} x_k^2 \sum_{k=1}^{m} y_k - \sum_{k=1}^{m} x_k \sum_{k=1}^{m} x_k y_k}{m \sum_{k=1}^{m} x_k^2 - \left(\sum_{k=1}^{m} x_k\right)^2} \tag{12}$$

3. En vous appuyant sur les équations précédentes, écrivez l'algorithme qui permet de calculer a et b. Vous pourrez utiliser les fonctions de `numpy` : lire les données du fichier et créer deux listes **x** et **y**, les transformer en `array`, utiliser les fonctions `sum` et produit de deux vecteurs, etc.
   Exemple si x = np.array([1,2,3]) et y = np.array([2,2,2])
   x.sum() renvoie 6
   La somme des carrés s'écrie : (x**2).sum()
   Le produit des vecteurs x et y s'écrie : x*y et la somme des éléments (x*y).sum()

4. Testez la méthode précédente qui permet de calculer a et b avec les données du fichier `dataset.csv`

3 Régression linéaire par descente de gradient

La méthode de descente de gradient est celle vue en cours. Elle consiste à écrire un algorithme qui minimise une fonction coût pour calculer les paramètres a et b. Ce calcul est réalisé de manière incrémentale, jusqu'à ce qu'il y ait convergence de l'algorithme. On peut mesurer la convergence avec la fonction coût qui tend vers zéro. L'algorithme qui calcule les paramètres a et b pendant un certain nombre d'itérations est donné par :

$$a = a - \alpha \frac{\partial J(a,b)}{\partial a} \tag{13}$$

$$b = b - \alpha \frac{\partial J(a,b)}{\partial b} \tag{14}$$

La fonction coût est donnée par l'équation (2).

1. Écrivez la fonction *compute_ partial_ derivates* qui prend en paramètres les anciennes valeurs de a et b et qui retourne les dérivées partielles de la fonction coût par rapport à chacun de ces paramètres.

2. Écrivez la fonction *compute_ new_ parameters* qui prend en paramètres les anciennes valeurs de a et b et qui calcule les nouvelles valeurs mises à jour par l'algorithme correspondant aux équations (13) et (14). Vous utiliserez les dérivées partielles calculées par la fonction *compute_ partial_ derivates.*

3. Écrivez la fonction *gradient_ descent* qui prend en paramètres le nombre d'itérations, le taux α, et les valeurs initiales de a et b.

4. L'algorithme Gradient Descent tente de réduire, à chaque itération, le coût global d'erreur en minimisant la fonction *J(a,b)*. Dans cette question on souhaite regarder comment évoluent les valeurs de *J(a,b)* au cours des itérations. Écrivez la fonction *compute_ cost_function* qui prend en paramètres les valeurs de a et b et retourne la valeur du coût global.

5. Écrivez la fonction qui affiche l'évolution du coût précédent en fonction du nombre d'itérations.

6. Écrivez le programme principal qui lance la descente de gradient sur le jeu de données du fichier *dataset.csv* fourni sur Moodle.