Cost function

Michel Donnet

January 9, 2024

Contents

1 Fonction de coût pour la régression logistique

Afin d'entraîner les paramètres de la régression logistique, il faut pouvoir comparer les résultats obtenus par la régression avec les résultats attendus.

Pour cela, on pourrait penser utiliser quelque chose comme la `Mean Squared Error (MSE)`, qui est une moyenne du carré de la différence entre le résultat obtenu par la régression (donné par z) et la valeur estimée y.

La MSE nous donne une estimation de l'erreur moyenne faite entre la fonction approximative f et la valeur attendue y.

L'objectif est donc de minimiser la MSE afin de minimiser l'erreur entre les valeurs estimées et les valeurs attendues.

Ce qui nous donnerait

$$MSE = \frac{1}{n}\sum_{i}^{n}(\sigma(z_i) - y_i)^2$$

avec σ la fonction sigmoïde utilisée pour la régression logistique, définie comme suit:

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

.

Donc notre MSE nous donnerait:

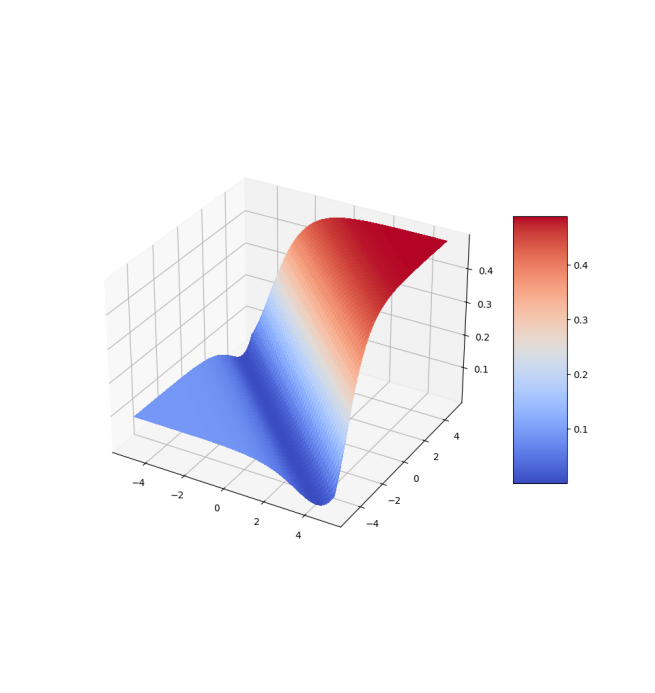
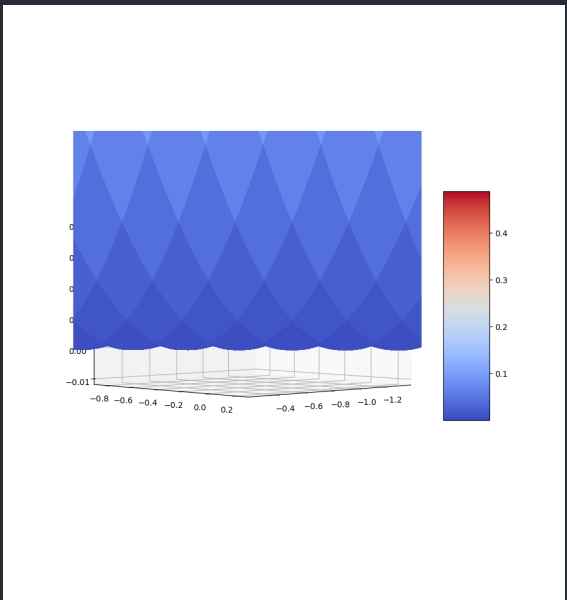
$$MSE = \frac{1}{n}\sum_{i}^{n}(\frac{1}{1 + e^{-z_i}} - y_i)^2$$

Afin de visualiser la fonction MSE obtenu, nous avons créé un graphe de la fonction

$$MSE = (\frac{1}{1 + e^{-(x*w+b)}} - y)^2$$

et nous avons choisi $x = 1$ et $y = 0.3$.

Nous obtenons alors les graphes suivants:

Global vision	Zoomed vision
	

Nous pouvons remarquer sur la figure ou l'on a zoomé, que la fonction MSE admet plusieurs minimum locaux. La fonction MSE n'est donc pas convexe... Ceci est problématique pour la descente en gradient, car celle-ci risquera de se retrouver coïncé dans un minimum local et ne trouvera jamais le minimum global.

Si on pouvait faire des plots pour la fonction avec plus de paramètres, on verrait mieux et plus de minimum locaux.

Cela est dû au fait que la fonction $\sigma(z)$ n'est pas linéaire. En effet, on a $\sigma(z) = \frac{1}{1+e^{-z}}$ et si on prend par exemple $\sigma(2) = \frac{1}{1+e^{-2}} = 0.8$ et $\sigma(-2) = \frac{1}{1+e^{-(-2)}} = \frac{1}{1+e^{2}} = 0.1$, on n'a pas que $\sigma(2) = -\sigma(-2)$, ce qui signifie que la fonction $\sigma(z)$ n'est pas linéaire. Cela entraîne que la fonction MSE n'est pas convexe.

La descente en gradient ne pourra donc pas fonctionnner correctement.

C'est pourquoi, on utilise plutôt la `log loss` fonction, qui s'appelle également `cross-entropy loss` fonction ou encore `cross-entropy`.

La formule de la `cross-entropy` est donnée par:

$$H(p,q) = -\sum_{x} p(x) log(q(x))$$

On peut considérer que $p(x) = y$ et que $q(x) = z$.

1.1 Nouveau test

On souhaite définir une fonction à minimiser permettant de trouver les paramètres optimaux de la régression logistique.

Notre classification se base sur la fonction sigmoïde $\sigma(z) = \frac{1}{1+e^{-z}}$.

Comme la fonction exponnentielle est toujours positive, on a bien que $\sigma(z) \in [0, 1]$.

La fonction sigmoïde nous donne la probabilité que l'élément donné appartienne à un label.

Autrement dit, la fonction sigmoïde est la fonction de répartition de la régression logistique.

Soit $Y \in \{0, 1\}$ les différents labels que peut prendre l'élément que l'on considère et soit X l'ensemble des caractéristiques connues de l'élément, dont on cherche à déterminer dans quelle classe le mettre, donc quel label on doit lui attribuer. Soit θ le vecteur des poids des covariables, indiquant à quel point les covariables influencent sur la décision du label. On a donc:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(X_1 \times w_1 + X_2 \times w_2 + \cdots + b)}}$$

et

$$P(Y = 0|X) = 1 - \frac{1}{1 + e^{-(X_1 \times w_1 + \cdots + b)}}$$

Pour plus de simplicité, on va considérer que le biais est compris dans les poids: au lieu d'écrire $z = w^T X + b$, on écrit $z = \theta^T \hat{X}$ avec $\hat{X} = \begin{bmatrix} 1 \\ X \end{bmatrix}$ modifié ou on a ajouté un 1 au début et $\theta = \begin{bmatrix} b \\ w \end{bmatrix}$. Ainsi, on a:

$$\theta^T \hat{X} = \theta_0 + X_1 \times \theta_1 + \cdots + X_n \times \theta_n = b + X_1 \times w_1 + \cdots + X_n \times w_n$$

Notre régression logistique binaire peut donc s'écrire comme:

$$P(Y = 1|X) = \frac{1}{1 + e^{\theta^T X}} = \sigma(\theta^T X)$$

et

$$P(Y = 0|X) = 1 - \sigma(\theta^T X)$$

Dans notre cas, on a plus que 2 labels. Il s'agit donc d'une régression multinomiale. Ainsi, on va modifier notre vecteur θ pour obtenir une régression multinomiale: l'idée est que pour chaque label $Y \in \{0, 1, \ldots, k\}$, on va calculer la probabilité que l'élément (avec ses caractéristiques X) appartienne à ce label, et la probabilité la plus grande va nous donner la classe dans laquelle on doit placer l'élément.

Ainsi, on aura pour k labels une matrice de poids donnée par:

$$\theta = \begin{bmatrix} \theta_{11} & \ldots & \theta_{1n} \\ \ldots & \ddots & \ldots \\ \theta_{k1} & \ldots & \theta_{kn} \end{bmatrix}$$

Et cela nous donne:

$$P(Y = i|X) = \sigma(\theta_i^T X)$$

Comme il s'agit de probabilités et que pour des caractéristiques X données, l'élément X appartient forcément à une classe, on a que:

$$\sum_i^k P(Y = i|X) = 1$$

1.1.0.1 Rapport de vraissemblance

Le rapport de vraissemblance est plus communément appelé odds ratio.

Si on veut essayer de définir le rapport de vraissemblance formellement, cela nous donne:

Soit $P(Y = 1|X) = p$ la probabilité que l'élément (possédant ses caractéristiques X) respecte une contrainte Y. On a donc $1-p$ la probabilité que l'élément ne respecte pas la contrainte Y. Le rapport de vraissemblance se définit de la sorte:

$$O(Y = 1|X) = \frac{p}{1-p}$$

On remarque:

$$\begin{array}{ll} \text{Si } p = 1-p \Rightarrow & O(Y = 1|X) = O(Y = 0|X) = 1 \\ \text{Si } p > 1-p \Rightarrow & O(Y = 1|X) > 1 \\ \text{Si } p < 1-p \Rightarrow & 0 \leq O(Y = 1|X) < 1 \text{ (car p, étant une probabilité, } \in [0, 1]) \end{array}$$

Ce qui signifie que si l'élément respecte bien la contrainte Y, alors le rapport de vraissemblance est supérieur à 1 et si l'élément ne respecte pas bien la contrainte Y, alors le rapport de vraissemblance est inférieur à 1.

Soit i le ième label et s l'ensemble de tous les labels moins le ième label.

On a donc:

$$P(Y = i|X) = p = \sigma(\theta_i^T X)$$

et

$$P(Y = s|X) = 1 - p = 1 - \sigma(\theta_i^T X)$$

Le rapport de vraissemblance s'écrit alors:

$$\begin{aligned} & O(Y = i|X) \\ &= \frac{P(Y = i|X)}{P(Y = s|X)} \\ &= \frac{\sigma(\theta_i^T X)}{1 - \sigma(\theta_i^T X)} \\ &= \frac{1}{1 + e^{-\theta_i^T X}} \times \frac{1}{1 - \frac{1}{1+e^{-\theta_i^T X}}} \\ &= \frac{1}{1 + e^{-\theta_i^T X}} \times \frac{1}{\frac{1+e^{-\theta_i^T X}-1}{1+e^{-\theta_i^T X}}} \\ &= \frac{1}{1 + e^{-\theta_i^T X}} \times \frac{1 + e^{-\theta_i^T X}}{e^{-\theta_i^T X}} \\ &= \frac{1}{e^{-\theta_i^T X}} \\ &= \boxed{e^{\theta_i^T X}} \end{aligned}$$

Le rapport de vraisemblance nous donne alors une indication sur le respect du label par notre élément. On va alors vouloir mettre l'élément dans la classe dont le label donne le plus grand rapport de vraissemblance.

Comme la fonction logarithme est une fonction strictement croissante, on peut appliquer la fonction sur le rapport de vraissemblance sans perdre l'information donnée par celui-ci. En effet, si le résultat est plus grand que $log(1) = 0$ alors l'élément respecte bien la contrainte et si le résultat est plus petit que $log(1) = 0$, alors l'élément ne respecte pas bien la contrainte.

Ainsi, cela nous donne:

$$log(O(Y = i|X)) = log(e^{\theta_i^T X}) = \theta_i^T X$$

On veut déterminer l'expression de $P(Y = i|X)$

On a:

$$\frac{P(Y = i|X)}{P(Y = s|X)} = e^{\theta_i^T X}$$

$$\Leftrightarrow P(Y = i|X) = e^{\theta_i^T X} P(Y = s|X)$$

Or

$$P(Y = s|X)$$

$$= \sum_{j \neq i}^{k} P(Y = j|X)$$

$$= \sum_{j \neq i}^{k} \sigma(\theta_j^T X)$$

$$= \sum_{j \neq i}^{k} \frac{1}{1 + e^{-\theta_j^T X}}$$

$$= \frac{1}{1 + \sum_{j \neq i}^{k} e^{-\theta_j^T X}}$$

Donc

$$P(Y = i|X) = \frac{e^{\theta_i^T X}}{1 + \sum_{j \neq i}^{k} e^{-\theta_j^T X}}$$

L'objectif étant de trouver le maximum de $P(Y = i|X)$, soit de trouver le minimum de $-log(P(Y = i|X)) = -\theta_i^T X + \sum_{j \neq i}^{k} log(1 + e^{-\theta_j^T X})$