Cost function

Michel Donnet

January 19, 2024

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.

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:

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

et

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

Pour plus de simplicité, on va considérer que le biais est compris dans les poids: au lieu d'écrire $z = wX + b$, on écrit $z = \hat{X}\theta$ avec $\hat{X} = \begin{bmatrix} X \\ 1 \end{bmatrix}$ modifié ou on a ajouté une colonne avec que des 1 à la fin de la matrice X et $\theta = \begin{bmatrix} w & b \end{bmatrix}$ afin d'avoir une bonne cohérence avec le rapport et le code. (On a trouvé cela plus

facile d'avoir pour chaque labels les poids et bias sur une ligne, donc d'avoir θ_1 pour le label 1 etc...) Ainsi, on a: $\hat{X}\theta^T = X_1 \cdot w_1 + X_2 \cdot w_2 + \cdots + b$

Pour la suite, on va noter $X = \hat{X}$

Notre régression logistique binaire peut donc s'écrire comme:

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

et

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

Généralisation

On désire donc trouver une nouvelle distribution $\phi(z)$ tel que:

$$\phi(z) \in [0, 1] \ \forall z$$

est une généralisation de la fonction $\sigma(z)$

On veut donc que pour une régression logistique binaire, on ait $\sigma(z) = \phi(z)$.

On peut remarquer que:

$$\begin{aligned}
&P(Y = 1|X) \\
&= \frac{1}{1 + e^{-X\theta^T}} \\
&= \frac{1}{1 + e^{-X\theta^T}} \cdot \frac{e^{X\theta^T}}{e^{X\theta^T}} \\
&= \frac{e^{X\theta^T}}{e^{X\theta^T} + e^{X\theta^T - X\theta^T}} \\
&= \frac{e^{X\theta^T}}{e^{X\theta^T} + e^0} \\
&= \frac{e^{X\theta^T}}{e^{X\theta^T} + 1}
\end{aligned}$$

On peut considérer que nous avons un vecteur de poids pour chaque label.

Ainsi, on a $\theta_0 = \begin{bmatrix} w_0 & b_0 \end{bmatrix}$ pour le label 0 et $\theta_1 = \begin{bmatrix} w_0 & b_0 \end{bmatrix}$ pour le label 1.

Comme on a besoin seulement d'un vecteur de poids pour déterminer le label de nouveaux éléments avec leurs caractéristiques, on peut considérer que $\theta_0 = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}$.

Ainsi, la formule précédente nous donne:

$$
\begin{aligned}
& P(Y=1|X) \\
&= \frac{e^{X\theta_1^T}}{e^{X\theta_1^T}+1} \\
&= \frac{e^{X\theta_1^T}}{e^{X\theta_1^T}+e^0} \\
&= \frac{e^{X\theta_1^T}}{e^{X\theta_1^T}+e^{0\cdot X}} \\
&= \frac{e^{X\theta_1^T}}{e^{X\theta_1^T}+e^{X\theta_0^T}} \\
&= \frac{e^{X\theta_1^T}}{\sum_{i=0}^{1} e^{X\theta_i^T}}
\end{aligned}
$$

On peut donc généraliser cette formule pour K labels.

Cela nous donne:

$$P(Y=k|X) = \frac{e^{X\theta_k^T}}{\sum_{i=0}^{K} e^{X\theta_i^T}}$$

Comme la fonction exponentielle est toujours positive, on a bien que:

$$0 \leq e^{X\theta_k^T} \leq e^{X\theta_k^T} + \sum_{i\neq k}^{K} e^{X\theta_i^T}$$

$$\Leftrightarrow 0 \leq e^{X\theta_k^T} \leq \sum_{i}^{K} e^{X\theta_i^T}$$

$$\Leftrightarrow 0 \leq \frac{e^{X\theta_k^T}}{\sum_{i}^{K} e^{X\theta_i^T}} \leq 1$$

$$\Leftrightarrow 0 \leq \phi(z) \leq 1$$

De plus, on a que:

$$
\begin{aligned}
& \sum_{k}^{K} P(Y=k|X) \\
&= \sum_{k}^{K} \frac{e^{X\theta_k^T}}{\sum_{i}^{K} e^{X\theta_i^T}} \\
&= \frac{\sum_{k}^{K} e^{X\theta_k^T}}{\sum_{i}^{K} e^{X\theta_i^T}}
\end{aligned}
$$

$$= \frac{\sum_i^K e^{X\theta_i^T}}{\sum_i^K e^{X\theta_i^T}}$$
$$= 1$$

Donc la fonction $\phi(z)$ est bien une fonction de distribution de probabilité qui généralise la fonction sigmoïde pour des problèmes à plusieurs labels.

Cette fonction est courramment appelée fonction `softmax`.

Notre objectif est donc de trouver une fonction de coût pour pouvoir entraîner les paramètres de la régression multinomiale. On cherche à maximiser la vraisemblance des données. Donc pour un label Y donné, on veut maximiser:

$$\sum_k^K f(Y,k)P(Y=k|X)$$

avec $f(Y,k)$ la fonction qui vaut 1 si $Y=k$ et 0 sinon.

Comme on a plusieurs couples de données (X_i, Y_i), on peut écrire la fonction précédente comme:

$$\sum_i^n \sum_k^K f(Y_i,k)P(Y_i=k|X_i)$$

En maximisant cette fonction, on fait en sorte que le paramètre θ_k permette d'obtenir la prédiction que le label soit égal à k avec la somme des probabilités où $Y_i = k$ est la plus grande possible.

Afin de pouvoir utiliser un algorithme comme la descente en gradient, il faut non pas maximiser une fonction, mais minimiser une fonction.

Tout d'abord, comme on travaille avec des exponentielles, on a intérêt à prendre un logarithme pour éviter d'avoir à travailler avec de trop grandes valeurs. Cette modification n'aura pas d'impact sur la convexité car la fonction logarithme est une fonction strictement croissante.

Enfin, comme on cherche une fonction à minimiser et non pas à maximiser pour pouvoir utiliser la descente en gradient, on va prendre l'inverse de la fonction.

Cela s'appelle courrament le `negative logarithm likelihood`.

Cela nous donne une fonction de coût comme suit:

$$\sum_i^n \sum_k^K f(Y_i,k)\log(\frac{1}{P(Y_i=k|X_i)})$$
$$\sum_i^n \sum_k^K f(Y_i,k)(\log(1) - \log(P(Y_i=k|X_i)))$$

$$-\sum_{i}^{n}\sum_{k}^{K} f(Y,k)\log(P(Y_i = k|X_i))$$

On peut minimiser cette fonction de coût grâce à une descente en gradient.

Dérivée

On va calculer la dérivée de la fonction de coût.

On a:

$$\begin{aligned} & log(P(Y = k|X)) \\ & = log\left(\frac{e^{X\theta_k^T}}{\sum_i^K e^{X\theta_i^T}}\right) \\ & = X\theta_k^T - log\left(\sum_i^K e^{X\theta_i^T}\right) \end{aligned}$$

Donc:

$$\frac{\partial}{\partial\theta_j}\sum_i^K f(Y,i)log(P(Y = i|X))$$

(NB: On considère que Y = k, tous les autres termes étant annulés car f = 0)

$$\begin{aligned} & = \frac{\partial}{\partial\theta_j} f(Y,k)log(P(Y = k|X)) \\ & = \frac{\partial}{\partial\theta_j}\left(X\theta_k^T - log\left(\sum_i^K e^{X\theta_i^T}\right)\right) \end{aligned}$$

Supposons que j = k.

$$\begin{aligned} & = X - \frac{\partial}{\partial\theta_j} log\left(\sum_i^K e^{X\theta_i^T}\right) \\ & = X - \frac{1}{\sum_i^K e^{X\theta_i^T}}\frac{\partial}{\partial\theta_j}\sum_i^K e^{X\theta_i^T} \\ & = X - \frac{1}{\sum_i^K e^{X\theta_i^T}}\frac{\partial}{\partial\theta_j} e^{X\theta_j^T} \\ & = X - \frac{Xe^{X\theta_j^T}}{\sum_i^K e^{X\theta_i^T}} \\ & = X - XP(Y = j|X) \\ & = X(1 - P(Y = j|X)) \end{aligned}$$

Supposons que $j \neq k$.

$$\begin{aligned}
&\frac{\partial}{\partial \theta_j}\left(X\theta_k^T - log\left(\sum_i^K e^{X\theta_i^T}\right)\right) \\
&= -\frac{\partial}{\partial \theta_j} log\left(\sum_i^K e^{X\theta_i^T}\right) \\
&= -\frac{1}{\sum_i^K e^{X\theta_i^T}} \frac{\partial}{\partial \theta_j} \sum_i^K e^{X\theta_i^T} \\
&= -\frac{1}{\sum_i^K e^{X\theta_i^T}} \frac{\partial}{\partial \theta_j} e^{X\theta_j^T} \\
&= -\frac{X e^{X\theta_j^T}}{\sum_i^K e^{X\theta_i^T}} \\
&= -XP(Y=j|X)
\end{aligned}$$

On a donc:

$$\frac{\partial}{\partial \theta_j} \sum_i^K f(Y,i) log(P(Y=i|X)) = X(f(Y,j) - P(Y=j|X))$$

car $f(Y,k)$ est égal à 1 si $Y=k$ et 0 sinon.

Donc pour n données, cela nous donne:

$$\frac{\partial}{\partial \theta_j} \sum_m^n \sum_i^K f(Y_m,i) log(P(Y_m=i|X_m)) = \sum_m^n X_m(f(Y_m,j) - P(Y_m=j|X_m))$$