Régression linéaire multiple : comparaison de l'algorithme de descente de gradient avec un algorithme plus direct

Vincent Guillemot

La régression linéaire multiple permet de construire un modèle (linéaire) entre plusieurs variables explicatives (la matrice X) et une variable à expliquer (le vecteur y). La méthode permet l'estimation d'un vecteur de poids β^* tel que

$$\beta^* = \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$

> *Remarque : dans tout l'exercice, nous considérons que toutes les variables sont centrées. Il n'y a donc pas besoin d'introduire d'intercept dans le modèle.*

Solution analytique

1. Utilisez les notions de calcul matriciel présentées plus tôt pour calculer β^*.

Il est bien sûr très facile, à l'aide d'un logiciel comme R, d'utiliser des fonctions déjà implémentées pour calculer la solution de manière directe.

Le but de l'exercice est plutôt de comparer deux algorithmes itératifs qui sont composés tous les deux d'opérations matricielles simples :

- le premier est basé sur le calcul d'une inverse,
- le deuxième sera l'algorithme du gradient.

Solution directe

L'algorithme de Newton-Schulz (aussi appelé Hotelling-Bodewig) est un algorithme itératif qui permet de calculer l'inverse d'une matrice carrée A $(p \times p)$. Sa formulation est la suivante :

$$V_{k+1} = V_k(2I - AV_k),\ k = 0, 1, 2, \ldots$$

Avec la matrice V_0 qui est telle que toutes les valeurs propres de $I - AV_0$ sont inférieures à 1. Par exemple, si λ est la valeur propre maximale de A, alors $V_0 = \frac{1}{\lambda^2}A$ devrait permettre à l'algorithme de converger.

Cet algorithme peut donc être utilisé dès que l'on connait la valeur de λ. Or nous nous trouvons dans un cas particulier très intéressant : la matrice à inverser est une matrice définie positive. Cela nous permet d'utiliser l'algorithme de la puissance itérée. Sa formulation est la suivante :

$$u_{k+1} = \frac{1}{\|Au_k\|_2} Au_k,\ k = 0, 1, 2, \ldots$$

Avec u_0 qui peut être un vecteur quelconque (alétoire par exemple). A convergence, la valeur de λ est approximée par $u_\infty^\top A u_\infty$.

2. Implémentez l'algorithme de la puissance itérée pour calculer λ.
3. Implémentez l'algorithme de Newton-Shulz pour calculer A^{-1}.

La méthode du gradient

4. Implémentez la méthode du gradient pour trouver la solution du problème d'optimisation.
5. Comparez avec la méthode *directe*.