AIC-4101C – Machine learning

1. Régression linéaire et descente de gradient

Kevin Zagalo & Marwan W. El Khazen

<kevin.zagalo@inria.fr>

<marwan.wehaiba-el-khazen@inria.fr>

Automne 2021

Exercice 1 *Descente de gradient*

Nous voulons minimiser la fonction $F : \mathbf{R}^d \to \mathbf{R}$, dérivable sur $\mathbf{R}^d$. Soit ∇F son gradient. La descente de gradient est un algorithme itératif :

— **Initialisation** $x_0 \in \mathbf{R}^d$

— **Itération** $x_{k+1} = x_k - \alpha \nabla F(x_k)$

Ceci est itéré jusqu'à convergence.

Question 1

Suggérer un critère d'arrêt pour l'algorithme.

Question 2

Implémenter l'algorithme en Python (nous définirons une fonction prenant en entrée la fonction F, son gradient ∇F et d'autres paramètres à déterminer).

Question 3

Que se passe-t-il si F n'est pas convexe ?

Question 4

Discuter l'importance du point initial x_0 dans le cas convexe, puis dans le cas non convexe.

■

Exercise 2 *Méthode des moindres carrés*

Une régression linéaire a comme entrée les observations $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots)$, les réponses associées $\mathbf{y} = (y_{\mathbf{x}_1}, y_{\mathbf{x}_2}, \ldots)$, et en sortie $\beta = (\beta_1, \ldots, \beta_d)$, et le biais β_0. Nous définissons la fonction $h_\beta(\mathbf{x}) = \mathbf{x} \cdot \beta + \beta_0$. Soient les erreurs locale et globale, respectivement :

$$e(\mathbf{x}; \beta) = \frac{1}{2}(y_\mathbf{x} - h_\beta(\mathbf{x}))^2 \ ; \ E(X; \beta) = \frac{1}{|X|} \sum_{\mathbf{x} \in X} e(\mathbf{x}; \beta) \tag{1}$$

Question 1

Nous avons traité le biais β_0 et β séparemment. Montrer que nous pouvons le considérer comme un poids. Comment adapter le problème de regression ?

Afin d'estimer les paramètres $\hat{\beta}$ optimaux, nous allons minimiser l'erreur globale. Nous allons implémenter l'algorithme suivant :

- **Entrée** X, les valeurs associées $\mathbf{y}$ et $\alpha > 0$.
- $\beta^{(0)} = \vec{0}$
- **Faire**
  - **Calculer** $L(\beta^{(t)}) = E(X; \beta^{(t)})$
  - **Mettre à jour** $\beta^{(t+1)} = \beta^{(t)} - \alpha \nabla L(\beta^{(t)})$
- **Jusqu'à** ce que toutes les données soient explorées et la *convergence* de la suite des paramètres de poids $\hat{\beta}$.

Question 2

Ecrire une fonction de mise à jour des paramètres de poids.

Question 3

Implémenter l'algorithme.

■

Exercice 3 *Question de cours*

Soient $f(\beta) = (y - X\beta)^\top(y - X\beta)$ et $\hat{\beta} = \operatorname{argmin}_{\beta \in \mathbf{R}^{d+1}} f(\beta)$. Montrer que $\hat{\beta} = (X^\top X)^{-1}X^\top \mathbf{y}$.

Rappel : (voir ici)

- Soient $v, a \in \mathbf{R}^k$. Alors $\frac{\partial v^\top a}{\partial v} = \frac{\partial a^\top v}{\partial v} = a$,
- Soient $v \in \mathbf{R}^k, M \in \mathbf{R}^{k \times k}$, $\frac{\partial v^\top M v}{\partial v} = (M + M^\top)v$.

$\blacksquare$