

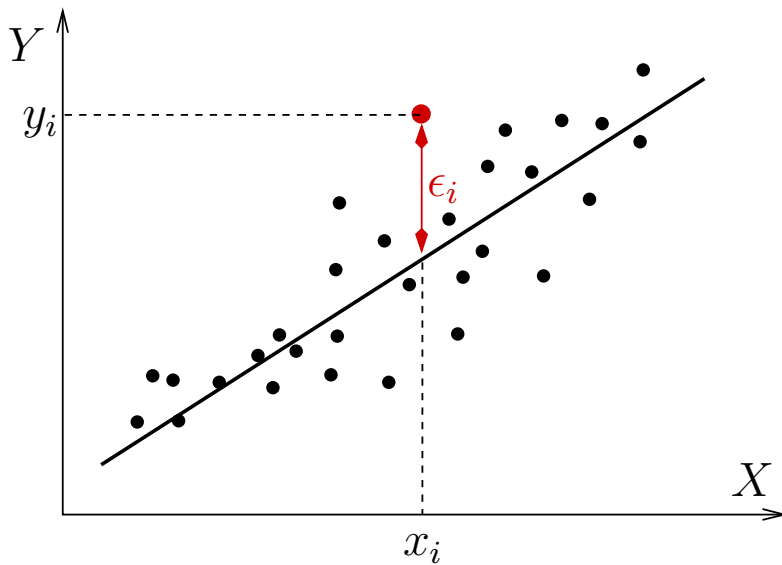
DESCENTE DE GRADIENT (ET RÉGRESSION)

Vincent Guigue
vincent.guigue@agroparistech.fr

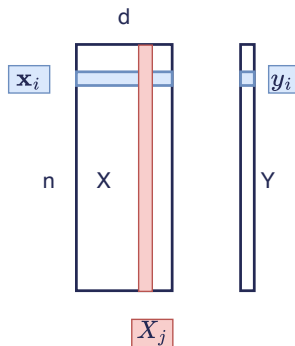
RÉGRESSION



Régression linéaire



Régression linéaire au sens des moindres carrés



- $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

- Modèle linéaire : $\hat{y}_i = \sum_{j=1}^d x_{ij} w_j + b$

- Simplification : $\hat{y}_i = \sum_{j=1}^{d+1} x_{ij} w_j$ avec $\forall i, x_{i,d+1} = 1$

- Critère d'optimisation : $C = \sum_{i=1}^n (\hat{y}_i - y_i)^2$

- Problème d'apprentissage :

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_{i=1}^n \left(\sum_{j=1}^{d+1} x_{ij} w_j - y_i \right)^2$$



Ecriture matricielle & résolution analytique

$$C = \sum_{i=1}^n (\hat{y}_i - y_i)^2 = \sum_{i=1}^n \left(\sum_{j=1}^{d+1} x_{ij} w_j - y_i \right)^2 = \|X\mathbf{w} - \mathbf{Y}\|^2$$

Résolution :

$$\nabla_{\mathbf{w}} C = \begin{pmatrix} \frac{\partial C}{\partial w_1} \\ \vdots \\ \frac{\partial C}{\partial w_d} \end{pmatrix} = 2X^T(X\mathbf{w} - \mathbf{Y}), \quad \frac{\partial C}{\partial w_j} = \sum_{i=1}^n 2x_{ij} \left(\sum_{j=1}^{d+1} x_{ij} w_j - y_i \right)$$

$$\mathbf{w}^* \Leftrightarrow \nabla_{\mathbf{w}} C = 0 \Leftrightarrow 2X^T(X\mathbf{w} - \mathbf{Y}) = 0 \Leftrightarrow X^T X \mathbf{w} = X^T \mathbf{Y}$$

Cette dernière forme correspond à un système d'équations linéaires :

$$\begin{pmatrix} X_1^T X_1 & X_1^T X_2 & \dots & X_1^T X_d \\ \vdots & \vdots & \vdots & \vdots \\ X_d^T X_1 & X_d^T X_2 & \dots & X_d^T X_d \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix} = \begin{pmatrix} X_1^T \mathbf{Y} \\ \vdots \\ X_d^T \mathbf{Y} \end{pmatrix}$$



Résolution rapide & efficace

Problème de la forme :

$$A\mathbf{w} = B, \quad A = X^T X \in \mathbb{R}^{d \times d}, \quad B = X^T Y \in \mathbb{R}^d$$

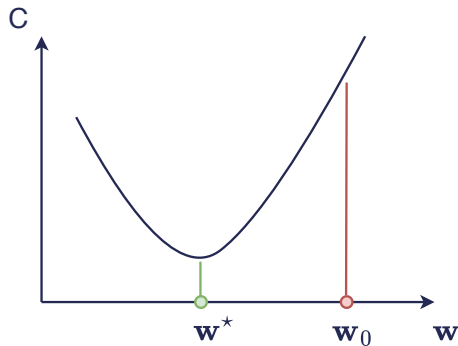
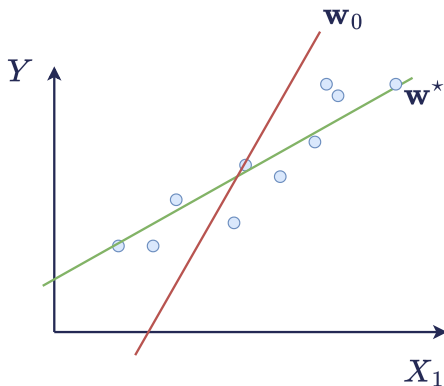
Résolution `w=np.linalg.solve(A,B)`

Un cadre idéal pour étudier la descente de gradient :

- 1 descente de gradient = solution approchée
- 2 descente de gradient = le 4x4 de l'optimisation :
 - on fait tout... Et parfois n'importe quoi

GRADIENT

Espace de description vs espace des paramètres



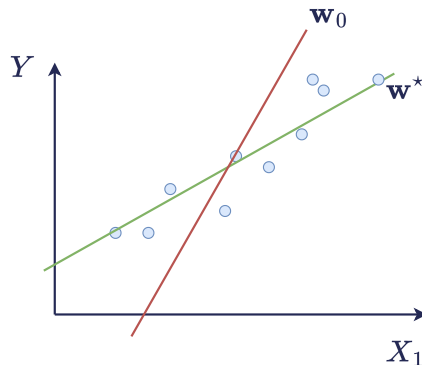
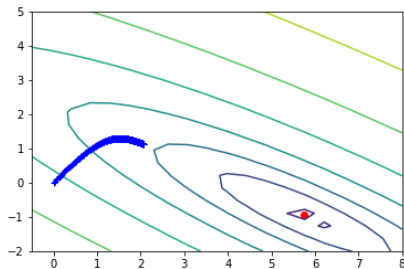
Algorithme itératif de la descente de gradient :

- 1 Initialiser w_0
- 2 En boucle (avec mise à jour du gradient) :

$$w^{t+1} = w^t - \varepsilon \nabla_w C$$

But du TP associé

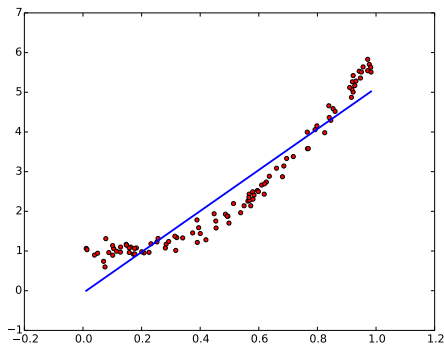
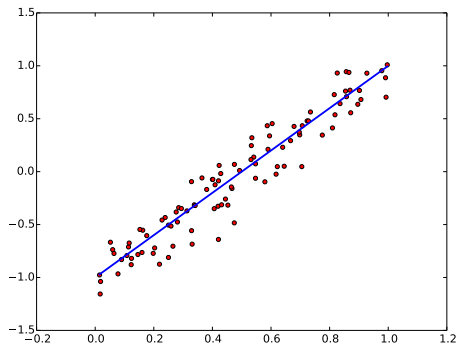
- 1 Rappeler les bases de la régression
- 2 Générer des données jouet
- 3 Implémenter une stratégie d'apprentissage à base de gradient :
 - Initialiser aléatoirement un régresseur
 - Améliorer itérativement le classifieur
- 4 Comprendre et illustrer le processus
 - Dans l'espace des points & dans l'espace des paramètres
 - Analyser empiriquement le processus itératif par rapport à la solution analytique



EXTENSION



Passer à la régression non linéaire



Bonne nouvelle : le cadre précédent va nous permettre de passer facilement au cas non linéaire.



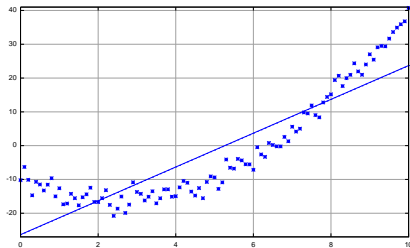
Transition facile

1 Transformer les données

$$X_{init} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \Rightarrow X = \begin{pmatrix} x_0^2 & x_0 & 1 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 \end{pmatrix}$$

2 Tout est fait !

$$\hat{Y} = X \cdot w, \quad X \in \mathbb{R}^{n \times 3}, \quad w = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \in \mathbb{R}^2, \quad \forall i, \hat{y}_i = ax_i^2 + bx_i + c$$

 \Rightarrow 