R4.04 Méthodes d'optimisation

Tom Ferragut, Thibault Godin & Lucie Naert
IUT de Vannes Informatique

Optimisation, minimum et dérivées

retour au lycée :

Un industriel cherche à optimiser la quantité de métal utilisée pour la fabrication d'une boite de conserve

$S = 2\pi r^2 + 2\pi r.h$

$V = \pi r^2 h$

Si on fixe le volume à $1L = 1000cm^3$ on obtient $h = \frac{1000}{\pi r^2}$,

Donc $S(r) = 2\pi r^2 + \frac{2V}{r}$

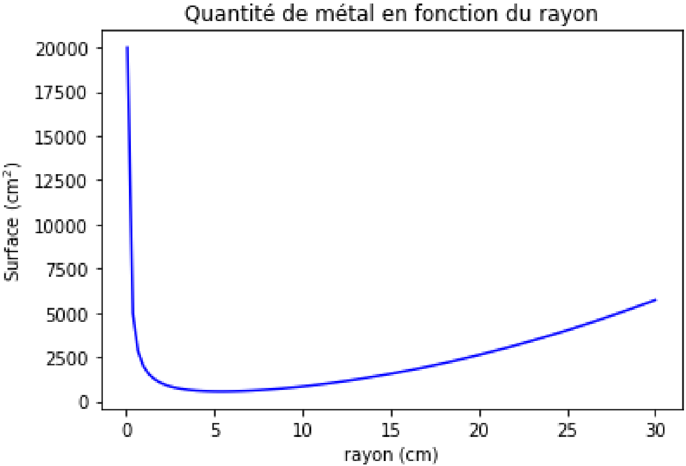
Le minimum s'obtient en dérivant S : $S'(r) = \frac{4\pi r^3 - 2V}{r^2}$

On cherche donc $4\pi r^3 - 2V = 0 \rightsquigarrow r = \sqrt[3]{\frac{2V}{4\pi}} \approx 5,42cm$

D'où $S_{min} \approx 554cm^2$ (et $h \approx 10,84cm$)

Caractérisation du minimum

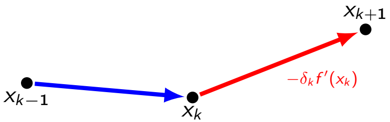
Soit $f : \mathbb{R} \to \mathbb{R}$ dérivable. Si f admet un minimum ou un maximum local en x^* alors f' s'annule en x^*



Un premier algorithme

Algorithme de la descente de gradient 1D
Soit une fonction $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto f(x)$ dont on sait calculer la dérivée $f'(x)$.
Données.

- Un point initial $x_0 \in \mathbb{R}$.
- Un niveau d'erreur $\varepsilon > 0$.



Itération. On calcule une suite de points $x_1, x_2, \ldots \in \mathbb{R}$ par récurrence de la façon suivante. Supposons que l'on ait déjà obtenu le point x_k :

- on calcule $f'(x_k)$,
- on choisit un pas δ_k et on calcule

$$x_{k+1} = x_k - \delta_k f'(x_k).$$

Arrêt. On s'arrête lorsque $\|f'(x_k)\| \leq \varepsilon$ (ou après un nombre prédéterminé de pas).

Descente de gradient, intuition

$f : \mathbb{R} \to \mathbb{R} : f(a) = a^2 + 1.$

a^* ?

$\rightsquigarrow$ gradient

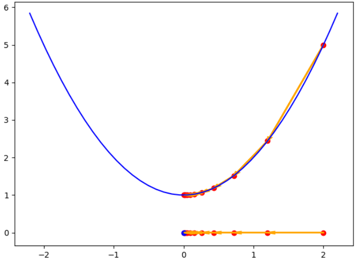
$$a_{k+1} = a_k - \delta \nabla f(a)$$

où δ est le pas et

$\nabla f(a) = f'(a) = 2a.$

$$a_{k+1} = a_k - 2\delta a_k.$$

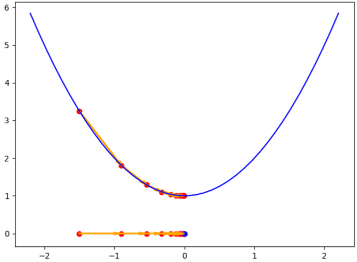
$\delta = 0.2 \quad a_0 = 2$



$\delta = 0.2$ $a_0 = 2.$

k	a_k	$f'(a_k) = \nabla f(a_k)$	$f(a_k)$
0	2	4	5
1	1.2	2.4	2.44
2	0.72	1.44	1.5184
3	0.43	0.86	1.1866
4	0.25	0.5184	1.0671
5	0.15	0.31	1.0241
6	0.093	0.186	1.0087
7	0.055	0.111	1.0031
8	0.033	0.067	1.0011
9	0.020	0.040	1.0004
10	0.012	0.024	1.0001

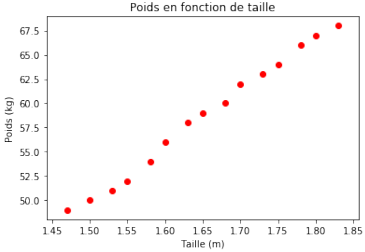
$\delta = 0.2 \quad a_0 = -1.5$



Application : régression linéaire simple

taille	147	150	153	155	158	160	163	165	168	170	173	175	178	180	183
poids	49	50	51	52	54	56	58	59	60	62	63	64	66	67	68

On a un vecteur $\boldsymbol{x} = (x_i)_i$ de tailles et un autre $\boldsymbol{y} = (y_i)_i$ de poids.

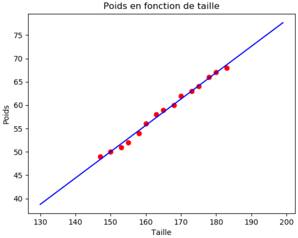


On cherche $a, b \in \mathbb{R}$ tels que $\boldsymbol{y} \approx a\boldsymbol{x} + b$

Application : régression linéaire simple

- On va créer une fonction mesurant la qualité de l'approximation
- On va chercher les paramètres donnant la meilleur estimation (selon cette mesure)

La fonction de coût usuelle est $E(a,b) = \sum_i (y_i - (ax_i + b))^2$ (c'est la somme des (carrés des) distances entre les points et la droite) $\rightsquigarrow$ on va minimiser cette fonction



Problème : on a 2 paramètres
$E(\mathrm{a},\mathrm{b}) = \sum_i (y_i - (\mathrm{a}x_i + \mathrm{b}))^2$

Optimisation, minimum et dérivées

$$\frac{\partial f}{\partial x}(x_0, y_0) = \lim_{h \to 0} \frac{f(x_0 + h, y_0) - f(x_0, y_0)}{h} \quad \frac{\partial f}{\partial y}(x_0, y_0) = \lim_{h \to 0} \frac{f(x_0, y_0 + h) - f(x_0, y_0)}{h}.$$

on fait "comme si" f était une fonction d'une variable et on dérive par rapport à celle-ci.

On vectorise souvent : $grad(f)(x_0, y_0) = \nabla f(x_0, y_0) = \left(\frac{\partial f}{\partial x}(x_0, y_0), \frac{\partial f}{\partial y}(x_0, y_0)\right)$

Soit $f : \mathbb{R}^2 \to \mathbb{R}$. Si f admet un minimum ou un maximum local en (x^*, y^*) alors [a] le gradient est le vecteur nul en ce point, autrement dit :

$$\frac{\partial f}{\partial x}(x^*, y^*) = 0 \quad \text{et} \quad \frac{\partial f}{\partial y}(x^*, y^*) = 0.$$

a. C'est même une équivalence si la dérivée de f est continue et que f est convexe

Même chose avec plus de dimensions/variables/paramètres

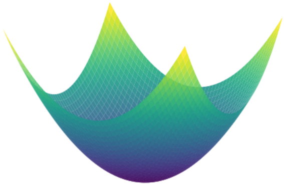
Exemple

$$\begin{array}{rccc} f: & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ & (a,b) & \longmapsto & a^2+b^2 \end{array}$$

$\nabla f(a,b) = (2a, 2b)$

$a^* = b^* = 0$

$\rightsquigarrow$ minimum en $(0,0)$

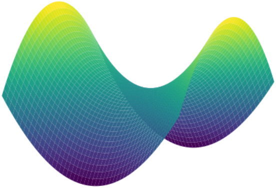


$$\begin{array}{rccc} f: & \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ & (a,b) & \longmapsto & -a^2+b^2 \end{array}$$

$\nabla f(a,b) = (-2a, 2b)$

$a^* = b^* = 0$

$\rightsquigarrow$ $(0,0)$ pas d'extremum (point selle)

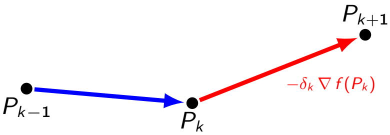


Descente de gradient

Algorithme de la descente de gradient.
Soit une fonction $f : \mathbb{R}^n \to \mathbb{R}$, $P \mapsto f(P)$ de plusieurs variables, avec $P = (a_1, \ldots, a_n)$, dont on sait calculer le gradient $\nabla f(P)$.

Données.

- Un point initial $P_0 \in \mathbb{R}^n$.
- Un niveau d'erreur $\varepsilon > 0$.



Itération. On calcule une suite de points $P_1, P_2, \ldots \in \mathbb{R}^n$ par récurrence de la façon suivante.
Supposons que l'on ait déjà obtenu le point P_k :

- on calcule $\nabla f(P_k)$,
- on choisit un pas δ_k et on calcule

$$P_{k+1} = P_k - \delta_k \nabla f(P_k).$$

Arrêt. On s'arrête lorsque $\| \nabla f(P_k) \| \leq \varepsilon$.

Et si le gradient est difficile à calculer ?

Approximation numérique du gradient :

$$\nabla f(x_0, y_0) = \left(\frac{\partial f}{\partial x}(x_0, y_0), \frac{\partial f}{\partial y}(x_0, y_0)\right)$$
$$\approx \left(\frac{f(x_0+\varepsilon, y_0) - f(x_0, y_0)}{\varepsilon}, \frac{f(x_0, y_0+\varepsilon) - f(x_0, y_0)}{\varepsilon}\right)$$

il y a de nombreuses sous-méthodes pour améliorer/adapter la descente de gradient :

Gradient stochastique, à pas optimal, conjugué ...

Application : régression linéaire simple

La fonction de coût usuelle est $E(a,b) = \sum_i (y_i - (ax_i + b))^2$

Estimation de b^* On considère $E(a,b) = E_a(b)$ comme une fonction de b.

$E(a,b) = \sum_i (y_i - (ax_i + b))^2$ est une fonction (de b) de la forme somme de u^2, donc la dérivée est une somme de $2u'(b)u(b)$

ici $u_i = (y_i - ax_i - b)$ donc $u' = -1$.
Alors $\frac{\partial E}{\partial b}(a,b) = \sum_i 2(-1)(y_i - ax_i - b)$.
b^* annulant l'équation $\rightsquigarrow$

$$-2\sum_i (y_i - ax_i - b^*) = 0$$

$$\sum_i (y_i - ax_i - b^*) = 0$$

$$n.b^* = \sum y_i - a\sum x_i$$

$$b^* = \frac{\sum_i y_i}{n} - a\frac{\sum_i x_i}{n}$$

$$b^* = \overline{\boldsymbol{y}} - a\overline{\boldsymbol{x}}$$

Application : régression linéaire simple

La fonction de coût usuelle est $E(a,b) = \sum_i (y_i - (ax_i + b))^2$

Estimation de a^* On considère $E(a,b) = E_b(a)$ comme une fonction de a.

On se place au point b^* optimal pour le second paramètre
$E(a,b^*) = \sum_i (y_i - (ax_i + (\overline{\mathbf{y}} - a\overline{\mathbf{x}})))^2 = \sum_i (y_i - \overline{\mathbf{y}} - ax_i + a\overline{\mathbf{x}})^2$

On développe l'identité remarquable

$$E(a,b^*) = \sum_i \left[a^2(x_i - \overline{\mathbf{x}})^2\right] + 2\sum_i \left[-a(x_i - \overline{\mathbf{x}})((y_i - \overline{\mathbf{y}}))\right] + \sum_i \left[(y_i - \overline{\mathbf{y}})^2\right]$$

on retrouve une fonction (de a) de la forme sommes de u^2, donc la dérivée est une somme de $2u'(a)u(a)$, une fonction somme de $v(a)$ et une constante. Donc

$$\frac{\partial E}{\partial a}(a,b^*) = \sum_i \left[2a(x_i - \overline{\mathbf{x}})^2\right] - 2\sum_i \left[(x_i - \overline{\mathbf{x}})((y_i - \overline{\mathbf{y}}))\right].$$

Application : régression linéaire simple

On cherche l'optimum

$$\sum_i \left[2a^*(x_i - \overline{\boldsymbol{x}})^2\right] - 2\sum_i \left[(x_i - \overline{\boldsymbol{x}})((y_i - \overline{\boldsymbol{y}}))\right] = 0$$

$$2a^* \sum_i \left[(x_i - \overline{\boldsymbol{x}})^2\right] - 2\sum_i \left[(x_i - \overline{\boldsymbol{x}})((y_i - \overline{\boldsymbol{y}}))\right] = 0$$

$$a^* = \frac{\sum_i \left[(x_i - \overline{\boldsymbol{x}})((y_i - \overline{\boldsymbol{y}}))\right]}{\sum_i \left[(x_i - \overline{\boldsymbol{x}})^2\right]}$$

$$a^* = \frac{Cov(\boldsymbol{x}, \boldsymbol{y})}{Var(\boldsymbol{x})}$$

Application : régression linéaire simple

taille	147	150	153	155	158	160	163	165	168	170	173	175	178	180	183
poids	49	50	51	52	54	56	58	59	60	62	63	64	66	67	68

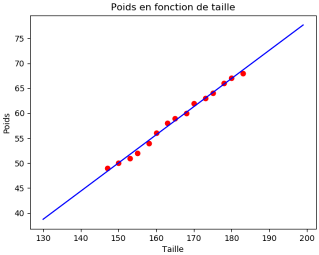
Ici on obtient :

$Var(\boldsymbol{x}) = 119.1$

$Cov(\boldsymbol{x}, \boldsymbol{y}) = 67.15$

d'où $a^* = 0.5638$ et

$b^* = -34.5423$



$\rightsquigarrow$ prédictions : le modèle prédit pour une personne de $177cm$ un poids de $177a^* + b^* \approx 65,3kg$