

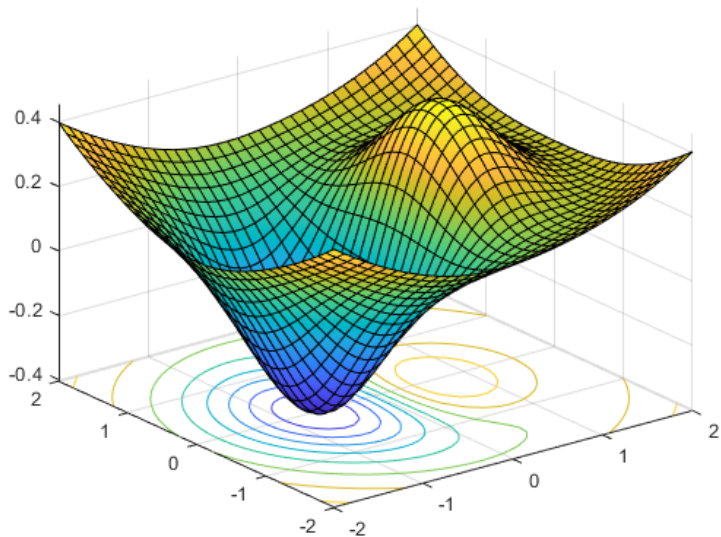
Introduction to data science & artificial intelligence (INF7100)

Arthur Charpentier

#41 Optimization

été 2020

Optimization



Optimisation: continuous (differentiable) case

The problem is to solve $\min_{y \in \mathbb{R}} \{f(y)\}$

Note: $\min_{y \in \mathbb{R}} \{f(y)\} = \max_{y \in \mathbb{R}} \{-f(y)\}$

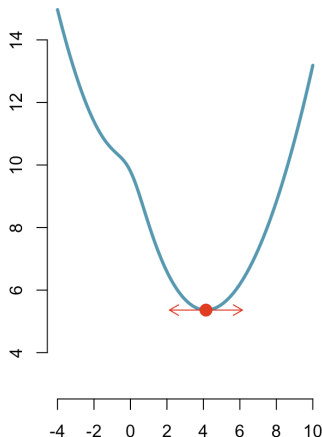
Note: $y^* \in \operatorname{argmin}_{y \in \mathbb{R}} \{f(y)\}$

and $\min_{y \in \mathbb{R}} \{f(y)\} = f(y^*)$.

First order condition

$$f'(y^*) = \left. \frac{\partial f(y)}{\partial y} \right|_{y=y^*} = 0$$

(necessary condition)



Optimisation: continuous (differentiable) case

First order condition

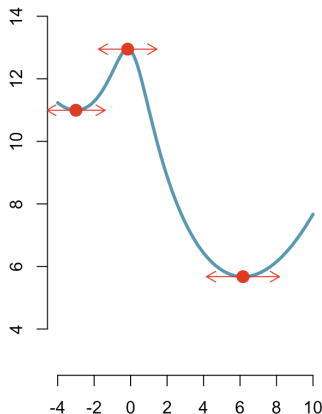
$$f'(y^*) = \left. \frac{\partial f(y)}{\partial y} \right|_{y=y^*} = 0$$

might be not sufficient

$$f''(y^*) = \left. \frac{\partial^2 f}{\partial y^2} \right|_{y=y^*} > 0 : \text{minimum}$$

$$f''(y^*) = \left. \frac{\partial^2 f}{\partial y^2} \right|_{y=y^*} < 0 : \text{maximum}$$

can be a local minimum...



Optimisation: continuous (differentiable) case

Example : $\{y_1, \dots, y_n\}$ in \mathbb{R} , let

$$f(y) = \sum_{i=1}^n (y_i - y)^2$$

$$\frac{\partial f(y)}{\partial y} = \frac{\partial}{\partial y} \sum_{i=1}^n (y_i - y)^2 = \sum_{i=1}^n \frac{\partial (y_i - y)^2}{\partial y} = \sum_{i=1}^n -2(y_i - y)$$

so

$$\left. \frac{\partial f(y)}{\partial y} \right|_{y=y^*} = 0 \text{ if and only if } \sum_{i=1}^n (y_i - y^*) = 0 \text{ or } \sum_{i=1}^n y_i = ny^*$$

$$\text{i.e. } y^* = \frac{1}{n} \sum_{i=1}^n y_i = \bar{y}.$$

Optimisation: continuous (differentiable) case

Solving $f'(y^*) = 0$ numerically

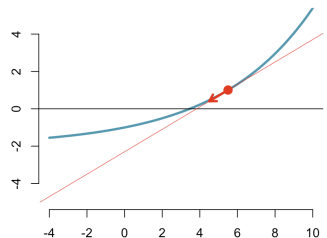
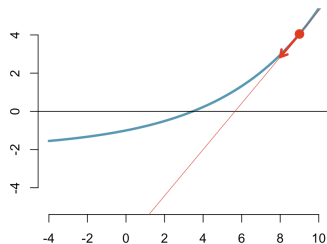
Newton's method: solve $g(y^*) = 0$

$$g(y) \simeq g(y_0) + g'(y_0)(y - y_0)$$

If $g(y) \simeq 0$, $g(y_0) + g'(y_0)(y - y_0) \simeq 0$

Start from y_0 , then

$$y_{k+1} = y_k - \frac{g(y_k)}{g'(y_k)}$$

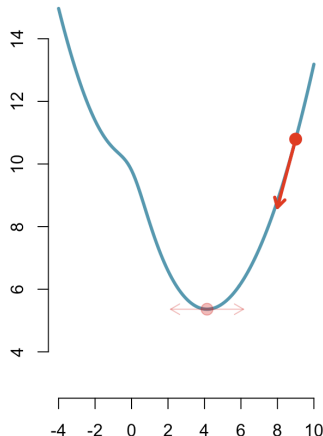


Optimisation: continuous (differentiable) case

To solve $f'(y^*) = 0$ numerically
Start from y_0 , then

$$y_{k+1} = y_k - \frac{f'(y_k)}{f''(y_k)}$$

$f'(y_k)$ gives the direction
 $f''(y_k)$ gives the speed of convergence
(close to the minimum $f''(y_k) > 0$)



Optimisation: continuous (differentiable) case

In python

```
1 > import statistics as stat
2 > v=[0.89367,-1.04729,1.97133,-0.38363,1.65414]
3 > stat.mean(v)
4 0.617644
5 > import numpy as np
6 > def f0(x):
7 ...     return np.sum((np.array(v)-x)**2)
8 > f = np.vectorize(f0)
9 > from scipy.optimize import fminbound
10 > fminbound(f, -1, 1)
11 0.6176439999999992
```

or

```
1 > from scipy.optimize import minimize
2 > minimize(f, 0, method='nelder-mead')
3     final_simplex: (array([[0.617625 ],
4         [0.6176875]]), array([6.75753495, 6.75753495]))
5         fun: 6.757534946524999
6         message: 'Optimization terminated successfully.'
```


Optimisation: continuous (differentiable) case

or

```
1 > def f(x):  
2 ...     s=[0]*len(v)  
3 ...     for i in range(len(v)):  
4 ...         s[i]=((v[i]-x)**2)  
5 ...     return sum(s)  
6 > fminbound(f, -1, 1)  
7 0.6176439999999992
```

and in R

```
1 > v = c(0.89367, -1.04729, 1.97133, -0.38363, 1.65414)  
2 > mean(v)  
3 [1] 0.617644  
4 > f = function(x) sum((v-x)^2)  
5 > optim(0, f)  
6 $par  
7 [1] 0.6175781  
8 $value  
9 [1] 6.757535
```

Optimisation: continuous (differentiable) case

The problem is $\min_{\mathbf{y} \in \mathbb{R}^p} \{f(\mathbf{y})\}$

or $\min_{(y_1, \dots, y_p) \in \mathbb{R}^p} \{f(y_1, \dots, y_p)\}$

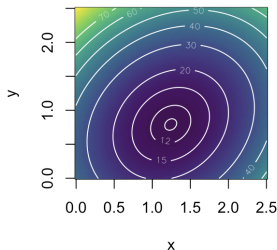
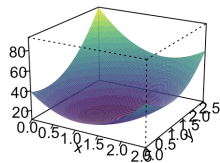
First order conditions: $\nabla f(\mathbf{y}^*) = \mathbf{0}$,

$$\left. \frac{\partial f(y_1, y_2, \dots, y_p)}{\partial y_1} \right|_{\mathbf{y}=\mathbf{y}^*} = 0$$

$$\left. \frac{\partial f(y_1, y_2, \dots, y_p)}{\partial y_2} \right|_{\mathbf{y}=\mathbf{y}^*} = 0$$

\vdots

$$\left. \frac{\partial f(y_1, y_2, \dots, y_p)}{\partial y_p} \right|_{\mathbf{y}=\mathbf{y}^*} = 0$$



Optimisation: continuous (differentiable) case

Example : $\{(x_1, y_1), \dots, (x_n, y_n)\}$ in \mathbb{R}^2 , let

$$f(a, b) = \sum_{i=1}^n (y_i - [a + bx_i])^2$$

$$\frac{\partial f(a, b)}{\partial a} = -2 \sum_{i=1}^n (y_i - [a + bx_i]) = -2(n\bar{y} - [a + bn\bar{y}])$$

$$\frac{\partial f(a, b)}{\partial b} = -2 \sum_{i=1}^n (y_i - [a + bx_i])x_i$$

$$\left. \frac{\partial f(a, b)}{\partial a} \right|_{(a,b)=(a^*, b^*)} = 0 \text{ means that } \bar{y} = a^* + b^* \bar{x},$$

$$\left. \frac{\partial f(a, b)}{\partial b} \right|_{(a,b)=(a^*, b^*)} = 0 \text{ means that } \hat{\varepsilon} \perp \mathbf{x}, \hat{\varepsilon}_i = y_i - [a^* + b^* x_i],$$

Optimisation: continuous (differentiable) case

To solve $\nabla f(\mathbf{y}^*) = \mathbf{0}$ numerically
Start from \mathbf{y}_0 , then

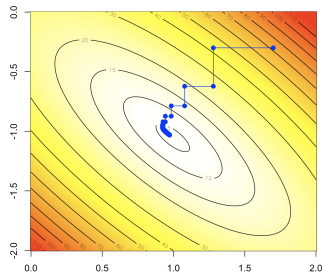
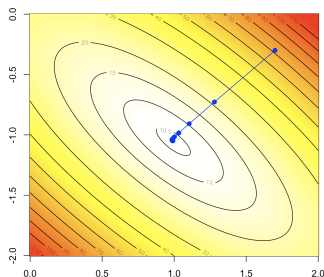
$$\mathbf{y}_{k+1} = \mathbf{y}_k - \mathbf{H}_k \nabla f(\mathbf{y}_k)$$

$\nabla f(\mathbf{y}_k)$ gives the direction

\mathbf{H}_k gives the speed of convergence

\mathbf{H}_k is the inverse of the Hessian matrix

One can also consider some numerical tricks, see **coordinate descent** where we iterate on the dimension (univariate optimisation problems)



Constrained Optimisation: continuous case

The problem is $\min_{(x,y) \in \mathbb{R}^2} \{f(x,y)\}$ subject to $g(x,y) \leq 0$,

or $\min_{(x,y) \in \mathbb{R}^2} \{f(x,y)\}$ subject to $g(x,y) = 0$.

$f(x,y)$ is the objective function

$g(x,y)$ is the constraint.

The trick is to consider the **Lagrangian**,

$$\mathcal{L}(x, y, \lambda) = f(x, y) + \lambda g(x, y)$$

Constrained Optimisation: Lagrangian

The optimization problem becomes

$$\min_{(x,y,\lambda)} \{\mathcal{L}(x, y, \lambda)\}$$

The first order conditions are now

$$\frac{\partial \mathcal{L}(x^*, y^*, \lambda^*)}{\partial x} = \frac{\partial f(x^*, y^*)}{\partial x} + \lambda^* \frac{\partial g(x^*, y^*)}{\partial x} = 0$$

$$\frac{\partial \mathcal{L}(x^*, y^*, \lambda^*)}{\partial y} = \frac{\partial f(x^*, y^*)}{\partial y} + \lambda^* \frac{\partial g(x^*, y^*)}{\partial y} = 0$$

$$\frac{\partial \mathcal{L}(x^*, y^*, \lambda^*)}{\partial \lambda} = g(x^*, y^*) = 0$$

Constrained Optimisation: Lagrangian

Interpretation: the ratios of the partial derivatives are all equal, and equal to $-\lambda$,

$$-\lambda = \frac{\partial f(x^*, y^*) / \partial x}{\partial g(x^*, y^*) / \partial x} = \frac{\partial f(x^*, y^*) / \partial y}{\partial g(x^*, y^*) / \partial y}$$

(ratios of marginal benefit to marginal cost are all equals)

Constrained Optimisation: Lagrangian

Example : $\{(x_{1,1}, x_{2,1}, y_1), \dots, (x_{1,n}, x_{2,n}, y_n)\}$ in \mathbb{R}^3 , let

$$f(b_1, b_2) = \sum_{i=1}^n (y_i - [b_1 x_{1,i} + b_2 x_{2,i}])^2$$

Goal: find $\min_{(b_1, b_2) \in \mathbb{R}^2} \{f(b_1, b_2)\}$ subject to $b_1^2 + b_2^2 \leq s$

or $\min_{\mathbf{b} \in \mathbb{R}^2} \{f(\mathbf{b})\}$ subject to $\|\mathbf{b}\|^2 \leq s$ (see Ridge regression)

The Lagrangian is

$$\mathcal{L}(b_1, b_2, \lambda) = \sum_{i=1}^n (y_i - [b_1 x_{1,i} + b_2 x_{2,i}])^2 + \lambda(b_1^2 + b_2^2 - s)$$

$$\frac{\partial \mathcal{L}(b_1, b_2, \lambda)}{\partial b_j} = -2 \sum_{i=1}^n x_{j,i} (y_i - [b_1 x_{1,i} + b_2 x_{2,i}]) + 2\lambda b_j$$

$$\frac{\partial \mathcal{L}(b_1, b_2, \lambda)}{\partial \lambda} = (b_1^2 + b_2^2 - s)$$

Constrained Optimisation: Lagrangian

To go further... using matrix notations,

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \text{ and } \mathbf{X} = \begin{pmatrix} x_{1,1} & x_{2,1} \\ \vdots & \vdots \\ x_{1,n} & x_{2,n} \end{pmatrix}$$

The solution of $\min_{\mathbf{b} \in \mathbb{R}^2} \{f(\mathbf{b})\}$ with $f(\mathbf{b}) = (\mathbf{y} - \mathbf{X}\mathbf{b})^\top (\mathbf{y} - \mathbf{X}\mathbf{b})$
is $\mathbf{b}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$

The solution of $\min_{\mathbf{b} \in \mathbb{R}^2} \{f(\mathbf{b})\}$ subject to $\|\mathbf{b}\|^2 \leq s$
is $\mathbf{b}^* = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbb{I})^{-1} \mathbf{X}^\top \mathbf{y}$ for some $\lambda > 0$.

Constrained Optimisation & Duality

Primal problem, $\min_{(x,y) \in \mathbb{R}^2} \{f(x,y)\}$ subject to $g(x,y) = 0$

Dual problem, $\max_{(x,y) \in \mathbb{R}^2} \{g(x,y)\}$ subject to $f(x,y) = f^*$

Wrap-up

- ▶ to solve $y^* \in \operatorname{argmin}_{y \in \mathbb{R}} \{f(y)\}$, solve $f'(y^*) = \left. \frac{\partial f(y)}{\partial y} \right|_{y=y^*} = 0$
- ▶ to solve $\mathbf{y}^* \in \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^p} \{f(\mathbf{y})\}$, solve $\nabla f(\mathbf{y}^*) = \mathbf{0}$
- ▶ to solve $\mathbf{y}^* \in \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^p} \{f(\mathbf{y})\}$, subject to $g(\mathbf{y}) \leq 0$,
define the Lagrangian, $\mathcal{L}(\mathbf{y}, \lambda) = f(\mathbf{y}) + \lambda g(\mathbf{y})$
solve $\nabla \mathcal{L}(\mathbf{y}^*, \lambda^*) = \mathbf{0}$
- ▶ numerical algorithm, from \mathbf{y}_0 , update $\mathbf{y}_{k+1} = \mathbf{y}_k - \mathbf{H}_k \nabla f(\mathbf{y}_k)$
where $\nabla f(\mathbf{y}_k)$ gives the direction (gradient descent)
(\mathbf{H}_k gives the speed of convergence)