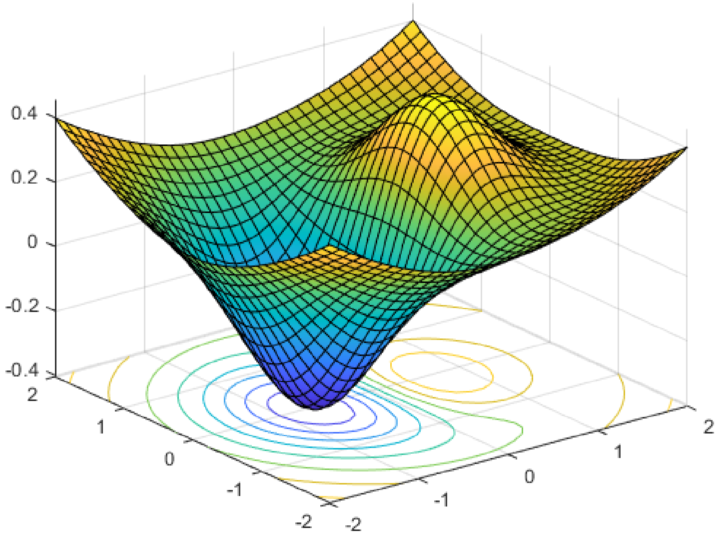
Introduction to data science & artificial intelligence (INF7100)

Arthur Charpentier

#41 Optimization

été 2020

Optimization



Optimisation: continuous (differentiable) case

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

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

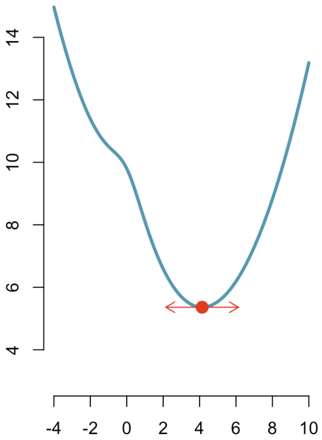
Note: $y^\star \in \text{argmin}_{y\in\mathbb{R}}\{f(y)\}$

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

First order condition

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

(necessary condition)



Optimisation: continuous (differentiable) case

First order condition

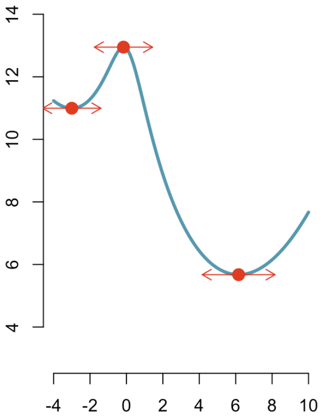
$$f'(y^\star) = \left.\frac{\partial f(y)}{\partial y}\right|_{y=y^\star} = 0$$

might be not sufficient

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

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

can be a local minimum...



Optimisation: continuous (differentiable) case

Example : $\{y_1, \cdots, y_n\}$ 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

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

i.e. $y^\star = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} y_i = \overline{y}$.

Optimisation: continuous (differentiable) case

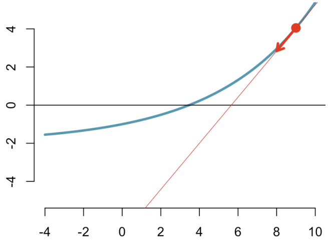
Solving $f'(y^\star) = 0$ numerically
Newton's method: solve $g(y^\star) = 0$

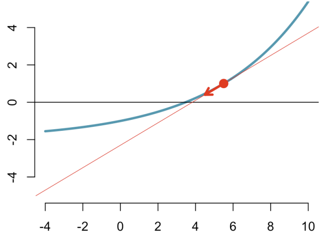
$$g(y) \simeq g(y_0) + g'(y_0)(y - y_0)$$

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

Start from y_0, then

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



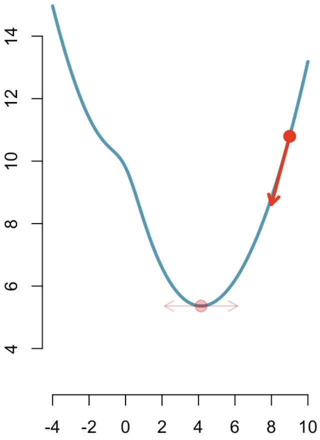


Optimisation: continuous (differentiable) case

To solve $f'(y^\star)=0$ 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

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

or

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

Optimisation: continuous (differentiable) case

or

```
> def f(x):
...   s=[0]*len(v)
...   for i in range(len(v)):
...     s[i]=((v[i]-x)**2)
...   return sum(s)
> fminbound(f, -1, 1)
0.61764399999999992
```

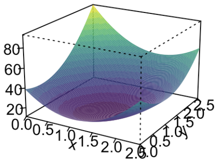
and in R

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

Optimisation: continuous (differentiable) case

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

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



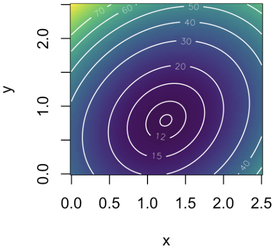
First order conditions: $\nabla f(\boldsymbol{y}^\star)=\boldsymbol{0}$,

$$\left.\frac{\partial f(y_1,y_2,\cdots,y_p)}{\partial y_1}\right|_{\boldsymbol{y}=\boldsymbol{y}^\star}=0$$

$$\left.\frac{\partial f(y_1,y_2,\cdots,y_p)}{\partial y_2}\right|_{\boldsymbol{y}=\boldsymbol{y}^\star}=0$$

$$\vdots$$

$$\left.\frac{\partial f(y_1,y_2,\cdots,y_p)}{\partial y_p}\right|_{\boldsymbol{y}=\boldsymbol{y}^\star}=0$$



Optimisation: continuous (differentiable) case

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

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

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

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

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

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

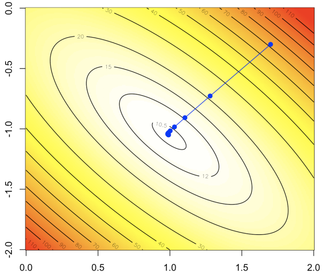
Optimisation: continuous (differentiable) case

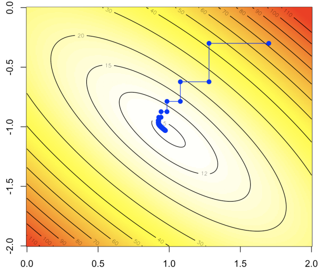
To solve $\nabla f(\boldsymbol{y}^\star) = \boldsymbol{0}$ numerically
Start from $\boldsymbol{y}_0$, then

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

$\nabla f(\boldsymbol{y}_k)$ gives the direction
$\boldsymbol{H}_k$ gives the speed of convergence
$\boldsymbol{H}_k$ is the inverse of the Hessian matrix

One con 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

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

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

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

Constrained Optimisation: Lagrangian

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

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

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

Constrained Optimisation: Lagrangian

Example : $\{(x_{1,1}, x_{2,1}, y_1), \cdots, (x_{1,n}, x_{2,n}, y_n)\}$ 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 $\displaystyle\min_{(b_1,b_2)\in\mathbb{R}^2} \{f(b_1, b_2)\}$ subject to $b_1^2 + b_2^2 \leq s$

or $\displaystyle\min_{\boldsymbol{b}\in\mathbb{R}^2} \{f(\boldsymbol{b})\}$ subject to $\|\boldsymbol{b}\|^2 \leq s$ (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,

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

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

The solution of $\min_{\boldsymbol{b}\in\mathbb{R}^2}\{f(\boldsymbol{b})\}$ subject to $\|\boldsymbol{b}\|^2 \leq s$
is $\boldsymbol{b}^\star = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$ 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 $f(x,y)=f^\star$

Wrap-up

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