Least squares problems

How to state and solve them, then evaluate their solutions

Stéphane Mottelet

Université de Technologie de Compiègne

April 28, 2020

Outline

Motivation and statistical framework

Motivation

Regression problem

- **Data** : $(x_i, y_i)_{i=1..n}$,

- **Model** : $y = f_\theta(x)$

  - $x \in \mathbb{R}$: independent variable
  - $y \in \mathbb{R}$: dependent variable (value found by observation)
  - $\theta \in \mathbb{R}^p$: parameters

- **Regression** problem

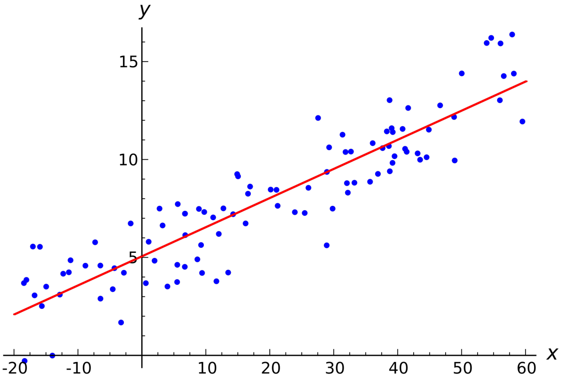
Find θ such that the model best explains the data,

i.e. y_i is close to $f_\theta(x_i)$, $i = 1 \ldots n$.

Motivation

Regression problem, example

Simple linear regression : $(x_i, y_i) \in \mathbb{R}^2$



$\longrightarrow$ find θ_1, θ_2 such that the data fits the model $y = \theta_1 + \theta_2 x$

How does one measure the fit/misfit ?

Motivation

Least squares method

The least squares method measures the fit with the Sum of Squared Residuals (SSR)

$$S(\theta) = \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2,$$

and aims to find $\hat{\theta}$ such that

$$\forall \theta \in \mathbb{R}^p, \quad S(\hat{\theta}) \leq S(\theta),$$

or equivalently

$$\hat{\theta} = \arg\min_{\theta \mathbb{R}^p} S(\theta).$$

Important issues

- statistical interpretation
- existence, uniqueness and practical determination of $\hat{\theta}$ (algorithms)

Statistical framework

Hypothesis

1. $(x_i)_{i=1\ldots n}$ are given
2. $(y_i)_{i=1\ldots n}$ are samples of random variables

$$y_i = f_\theta(x_i) + \varepsilon_i,\ i = 1\ldots n,$$

where ε_i, $i = 1\ldots n$ are independent and identically distributed (i.i.d.) and

$$E[\varepsilon_i] = 0,\ E[\varepsilon_i^2] = \sigma^2, \text{ density } \varepsilon \to g(\varepsilon)$$

The probability density of y_i is given by

$$\begin{aligned}\phi_\theta^i : \mathbb{R} &\longrightarrow \mathbb{R}\\ y &\longrightarrow \phi_\theta^i(y) = g(y - f_\theta(x_i))\end{aligned}$$

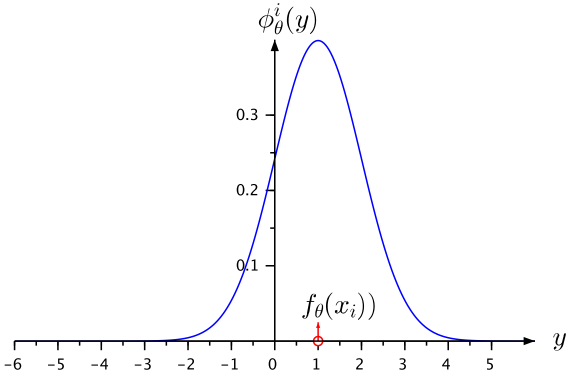
hence $E[y_i|\theta] = f_\theta(x_i)$.

Statistical framework

Example

If ε is normally distributed, i.e. $g(\varepsilon) = (\sigma\sqrt{2\pi})^{-1} \exp(-\frac{1}{2\sigma^2}\varepsilon^2)$, we have

$$\phi_\theta^i(y) = (\sigma\sqrt{2\pi})^{-1} \exp\left(-\frac{1}{2\sigma^2}\,(y - f_\theta(x_i))^2\right)$$



Statistical framework

Joint probability density and Likelihood function

- Joint density

When θ is given, as the (y_i) are independent, the density of the vector $\mathbf{y} = (y_1, \ldots, y_n)$ is

$$\phi_\theta(\mathbf{y}) = \prod_{i=1}^{n} \phi_\theta^i(y_i) = \phi_\theta^1(y_1)\phi_\theta^2(y_2)\ldots\phi_\theta^n(y_n).$$

Interpretation : for $D \subset \mathbb{R}^n$

$$\text{Prob}(\mathbf{y} \in D|\theta) = \int_D \phi_\theta(\mathbf{y})\, dy_1 \ldots dy_n$$

- Likelihood function

When a sample of $\mathbf{y}$ is given, then $L_\mathbf{y}(\theta) \stackrel{\text{def}}{=} \phi_\theta(\mathbf{y})$ is called

Likelihood of the parameters θ

Statistical framework

Maximum Likelihood Estimation

The Maximum Likelihood Estimate of θ is the vector $\hat{\theta}$ defined by

$$\hat{\theta} = \arg\max_{\theta \in \mathbb{R}^p} L_{\mathbf{y}}(\theta).$$

Under the Gaussian hypothesis, then

$$\begin{aligned} L_{\mathbf{y}}(\theta) &= \prod_{i=1}^{n} (\sigma\sqrt{2\pi})^{-1} \exp\left(-\frac{1}{2\sigma^2}\left(y_i - f_\theta(x_i)\right)^2\right), \\ &= (\sigma\sqrt{2\pi})^{-n} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - f_\theta(x_i)\right)^2\right), \end{aligned}$$

hence, we recover the least squares solution, i.e.

$$\arg\max_{\theta \in \mathbb{R}^p} L_{\mathbf{y}}(\theta) = \arg\min_{\theta \in \mathbb{R}^p} S(\theta).$$

- Least Absolute Deviation Regression : the misfit is measured by

$$S_1(\theta) = \sum_{i=1}^{n} |y_i - f_\theta(x_i)|.$$

Is $\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^p} S_1(\theta)$ is a maximum likelihood estimate ?

Yes, if ε_i has a Laplace distribution

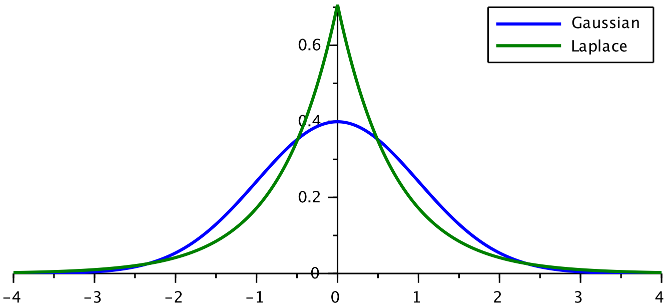
$$g(\varepsilon) = (\sigma\sqrt{2})^{-1} \exp\left(-\frac{\sqrt{2}}{\sigma}|\varepsilon|\right)$$

First issue : S_1 is not differentiable

Statistical framework

Alternatives : Least Absolute Deviation Regression

Densities of Gaussian vs. Laplacian random variables (with zero mean and unit variance) :



Second issue : the two statistical hypothesis are very different !

Statistical framework

Take home message

Take home message #1 :

Doing Least Squares Regression means that you assume that the model error is Gaussian.

However, if you have no idea about the model error :

1. the nice theoretical and computational framework you will get is worth doing this assumption. . .
2. *a posteriori* goodness of fit tests can be used to assess the normality of errors.

Maths reminder

Maths reminder

Matrix algebra

- Notation : $A \in \mathcal{M}_{n,m}(\mathbb{R})$, $x \in \mathbb{R}^n$,

$$A = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

- Product : for $B \in \mathcal{M}_{m,p}(\mathbb{R})$, $C = AB \in \mathcal{M}_{n,p}(\mathbb{R})$,

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}$$

- Identity matrix

$$\mathrm{I} = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$$

- Transposition, Inner product and norm :

$$A^\top \in \mathcal{M}_{m,n}(\mathbb{R}) \quad , \quad \left[A^\top\right]_{ij} = a_{ji}$$

For $x \in \mathbb{R}^n, y \in \mathbb{R}^n$,

$$\langle x, y\rangle = x^\top y = \sum_{i=1}^{n} x_i y_i, \quad \|x\|^2 = x^\top x$$

Maths reminder

Matrix algebra

- Linear dependance / independence :

  a set $\{x_1, \ldots, x_m\}$ of vectors in $\mathbb{R}^n$ is dependent if a vector x_j can be written as

  $$x_j = \sum_{k=1, k \neq i}^{m} \alpha_k x_k$$

  - a set of vectors which is not dependent is called independent
  - a set of $m > n$ vectors is necessarily dependent
  - a set of n independent vectors in $\mathbb{R}^n$ is called a basis

- The rank of a $A \in \mathcal{M}_{nm}$ is the number of its linearly independent columns

  $$\text{rank}(A) = m \iff \{Ax = 0 \Rightarrow x = 0\}$$

Maths reminder

Linear system of equations

When A is square

$$\mathrm{rank}(A) = n \iff \text{there exists } A^{-1} \text{ s.t. } A^{-1}A = AA^{-1} = \mathrm{I}$$

When the above property holds :

- For all $y \in \mathbb{R}^n$, the system of equations
$$Ax = y,$$
has a unique solution $x = A^{-1}y$.

- Computation : Gauss elimination algorithm (no computation of A^{-1})

in Scilab/Matlab : $x = A\backslash y$

Maths reminder

Differentiability

- Definition : let $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$,

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix}, \quad f_i : \mathbb{R}^n \longrightarrow \mathbb{R},$$

  f is differentiable at $a \in \mathbb{R}^n$ if

$$f(a+h) = f(a) + f'(a)h + \|h\|\varepsilon(h), \quad \lim_{h \to 0} \varepsilon(h) = 0$$

- Jacobian matrix, partial derivatives :

$$\left[f'(a)\right]_{ij} = \frac{\partial f_i}{\partial x_j}(a)$$

- Gradient : if $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, is differentiable at a,

$$f(a+h) = f(a) + \nabla f(a)^\top h + \|h\|\varepsilon(h), \quad \lim_{h \to 0} \varepsilon(h) = 0$$

Maths reminder

Nonlinear system of equations

When $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, a solution $\hat{x}$ to the system of equations

$$f(\hat{x}) = 0$$

can be found (or not) by the Newton's method : given x_0, for each k

1. consider the affine approximation of f at x_k

$$T(x) = f(x_k) + f'(x_k)(x - x_k)$$

2. take x_{k+1} such that $T(x_{k+1}) = 0$,

$$x_{k+1} = x_k - f'(x_k)^{-1} f(x_k)$$

Newton's method can be very fast... if x_0 is not too far from $\hat{x}$!

Maths reminder

Find a local minimum - gradient algorithm

When $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is differentiable, a vector $\hat{x}$ satisfying $\nabla f(\hat{x}) = 0$ and

$$\forall x \in \mathbb{R}^n, f(\hat{x}) \leq f(x)$$

can be found by the descent algorithm : given x_0, for each k :

1. select a direction d_k such that $\nabla f(x_k)^\top d_k < 0$
2. select a step ρ_k, such that

$$x_{k+1} = x_k + \rho_k d_k,$$

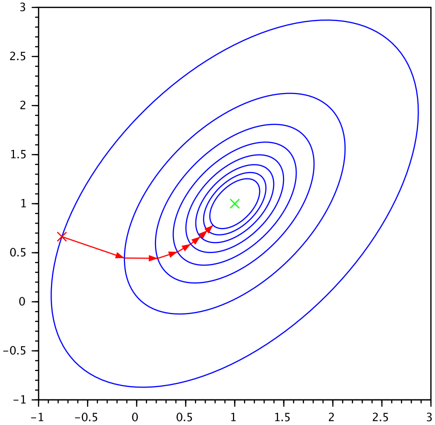
   satisfies (among other conditions)

$$f(x_{k+1}) < f(x_k)$$

The choice $d_k = -\nabla f(x_k)$ leads to the gradient algorithm

Maths reminder

Find a local minimum - gradient algorithm



$$x_{k+1} = x_k - \rho_k \nabla f(x_k),$$

Linear Least Squares (LLS)

Linear Least Squares

Linear models

- The model $y = f_\theta(x)$ is linear w.r.t. θ, i.e.

$$y = \sum_{j=1}^{p} \theta_j \phi_j(x), \quad \phi_k : \mathbb{R} \rightarrow \mathbb{R}$$

- Examples

  - $y = \sum_{j=1}^{p} \theta_j x^{j-1}$
  - $y = \sum_{j=1}^{p} \theta_j \cos \frac{(j-1)x}{T}$, where $T = x_n - x_1$
  - ...

Linear Least Squares

The residual for simple linear regression

- Simple linear regression

$$S(\theta) = \sum_{i=1}^{n} (\theta_1 + \theta_2 x_i - y_i)^2 = \|r(\theta)\|^2,$$

  Residual vector $r(\theta)$

$$r_i(\theta) = [1, x_i] \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} - y_i$$

- For the whole residual vector

$$r(\theta) = A\theta - y, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad A = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

Linear Least Squares

The residual for a general linear model

- General linear model $f_\theta(x) = \sum_{j=1}^{p} \theta_j \phi_j(x)$

$$
\begin{aligned}
S(\theta) &= \sum_{i=1}^{n} (f_\theta(x_i) - y_i)^2 = \|r(\theta)\|^2, \\
&= \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \theta_j \phi_j(x_i) - y_i \right)^2 = \|r(\theta)\|^2,
\end{aligned}
$$

Residual vector $r(\theta)$

$$
r_i(\theta) = [\phi_1(x_i), \ldots, \phi_p(x_i)] \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_2 \end{bmatrix} - y_i
$$

- For the whole residual vector $r(\theta) = A\theta - y$ where A has size $n \times p$ and

$$
a_{ij} = \phi_j(x_i).
$$

Linear Least Squares

Optimality conditions

- Linear Least Squares problem : find $\hat{\theta}$

$$\hat{\theta} = \arg\min_{\theta\mathbb{R}^p} S(\hat{\theta}) = \|A\theta - y\|^2$$

- Necessary optimality condition

$$\nabla S(\hat{\theta}) = 0$$

  Compute the gradient by expanding $S(\theta)$

Linear Least Squares

Optimality conditions

$$\begin{aligned}
S(\theta + h) &= \|A(\theta + h) - y\|^2 = \|A\theta - y + Ah\|^2 \\
&= (A\theta - y + Ah)^\top (A\theta - y + Ah) \\
&= (A\theta - y)^\top (A\theta - y) + (A\theta - y)^\top Ah + (Ah)^\top (A\theta - y) + (Ah)^\top Ah \\
&= \|A\theta - y\|^2 + 2(A\theta - y)^\top Ah + \|Ah\|^2 \\
&= S(\theta) + \nabla S(\theta)^\top h + \|Ah\|^2
\end{aligned}$$

$$\nabla S(\theta) = 2A^\top (A\theta - y),$$

hence $\nabla S(\hat{\theta}) = 0$ implies

$$A^\top A \hat{\theta} = A^\top y.$$

Theorem : a solution of the LLS problem is given by $\hat\theta$, solution of the “normal equations”

$$A^\top A\hat\theta = A^\top y,$$

moreover, if rank $A = p$ then $\hat\theta$ is unique.

Proof :

$$\begin{aligned} S(\theta) = S(\hat\theta + \theta - \hat\theta) &= S(\hat\theta) + \nabla S(\hat\theta)^\top(\theta - \hat\theta) + \|A(\theta - \hat\theta)\|^2, \\ &= S(\hat\theta) + \|A(\theta - \hat\theta)\|^2, \\ &\geq S(\hat\theta) \end{aligned}$$

Uniqueness :

$$\begin{aligned} S(\hat\theta) = S(\theta) &\iff \|A(\theta - \hat\theta)\|^2 = 0, \\ &\iff A(\theta - \hat\theta) = 0 \\ &\iff \theta = \hat\theta, \end{aligned}$$

Linear Least Squares

Simple linear regression

- $\operatorname{rank} A = 2$ if there exists $i \neq j$ such that $x_i \neq x_j$
- Computations :

$$S_x = \sum_{i=1}^{n} x_i,\ S_y = \sum_{i=1}^{n} y_i,\ S_{xy} = \sum_{i=1}^{n} x_i y_i,\ S_{xx} = \sum_{i=1}^{n} x_i^2$$

$$A^\top A = \begin{bmatrix} n & S_x \\ S_x & S_{xx} \end{bmatrix}, \quad A^\top y = \begin{bmatrix} S_y \\ S_{xy} \end{bmatrix}$$

$$\theta_1 = \frac{S_y S_{xx} - S_x S_{xy}}{n S_{xx} - S_x^2}, \quad \theta_2 = \frac{n S_{xy} - S_x S_y}{n S_{xx} - S_x^2}$$

Linear Least Squares

Practical resolution with Scilab

- When `A` is square and invertible, the Scilab command

  ```
  x=A\y
  ```

  computes `x`, the unique solution of `A*x=y`.

- When `A` is not square and has full (column) rank, then the command

  ```
  x=A\y
  ```

  computes `x`, the unique least squares solution. i.e. such that `norm(A*x-y)` is minimal.

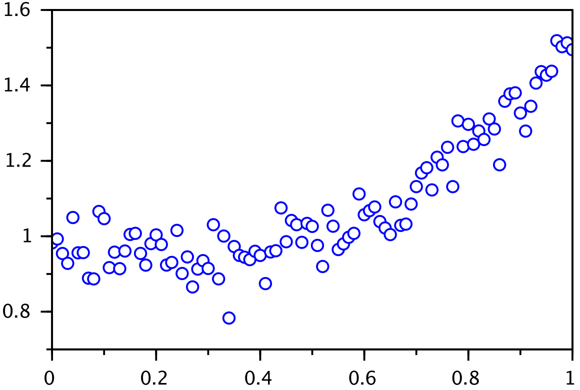
  - Although mathematically equivalent to

    ```
    x=(A'*A)\(A'*y)
    ```

    the command `x=A\y` is numerically more stable, precise and efficient

Linear Least Squares

Practical resolution with Scilab

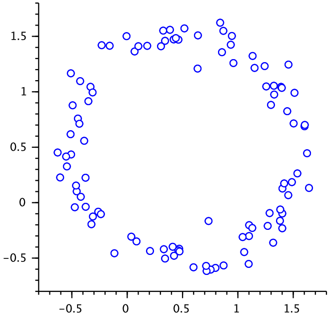


Fit $(x_i, y_i)_{i=1\ldots n}$ with a polynomial of degree 2 with Scilab

Linear Least Squares

An interesting example

Find a circle wich best fits $(x_i, y_i)_{i=1\ldots n}$ in the plane



- Minimize the algebraic distance

$$d(a,b,R) = \sum_{i=1}^{n} \left((x_i - a)^2 + (y_i - b)^2 - R^2\right)^2 = \|r\|^2$$

Linear Least Squares

An interesting example

- Algebraic distance

$$d(a,b,R)=\sum_{i=1}^{n}\left((x_i-a)^2+(y_i-b)^2-R^2\right)^2=\|r\|^2$$

The residual vector is non-linear w.r.t. (a,b,R) but we have

$$\begin{aligned} r_i &= R^2-a^2-b^2+2ax_i+2by_i-(x_i^2+y_i^2),\\ &= [2x_i\,,2y_i,\ 1]\begin{bmatrix} a \\ b \\ R^2-a^2-b^2 \end{bmatrix}-(x_i^2+y_i^2)\end{aligned}$$

hence residual is linear w.r.t. $\theta=(a,b,R^2-a^2-b^2)$.

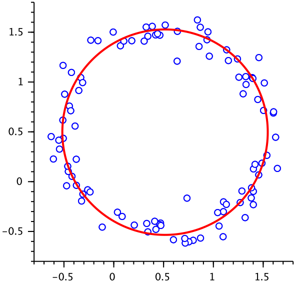
Linear Least Squares

An interesting example

- Standard form, the unknown is $\theta = (a, b, R^2 - a^2 - b^2)$

$$A = \begin{bmatrix} 2x_1 & 2y_1 & 1 \\ \vdots & \vdots & \vdots \\ 2x_n & 2y_n & 1 \end{bmatrix}, \quad z = \begin{bmatrix} x_1^2 + y_1^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix}, \quad d(a, b, R) = \|A\theta - z\|^2$$

- In Scilab




```
A=[2*x,2*y,ones(x)]
z=x.^2+y.^2
theta=A\z
a=theta(1)
b=theta(2)
R=sqrt(theta(3)+a^2+b^2)
t=linspace(0,2*%pi,100)
plot(x,y,"o",a+R*cos(t),b+R*sin(t))
```

Take home message #2 :

Solving linear least squares problem is just a matter of linear algebra

Non Linear Least Squares (NLLS)

Non Linear Least Squares (NLLS)

Example

- Consider data (x_i, y_i) to be fitted by the non linear model

$$y = f_\theta(x) = \exp(\theta_1 + \theta_2 x),$$

The "log trick" leads some people to minimize

$$S_{log}(\theta) = \sum_{i=1}^{n} (\log y_i - (\theta_1 + \theta_2 x_i))^2\,,$$

i.e. do simple linear regression of $(\log y_i)$ against (x_i), but this violates a fundamental hypothesis because

if $y_i - f_\theta(x_i)$ is normally distributed then $\log y_i - \log f_\theta(x_i)$ is not !

Non Linear Least Squares (NLLS)

Possibles angles of attack

Remember that

$$S(\theta) = \|r(\theta)\|^2, \quad r_i(\theta) = f_\theta(x_i) - y_i.$$

A local minimum of S can be found by different methods :

- Find a solution of the non linear systems of equations

$$\nabla S(\theta) = 2r'(\theta)^\top r(\theta) = 0,$$

  with the Newton's method :

  - needs to compute the Jacobian of the gradient itself (do you really want to compute second derivatives ?),
  - does not guarantee convergence towards a minimum.

Non Linear Least Squares (NLLS)

Possibles angles of attack

Use the spirit of Newton's method as follows : start with θ_0 and for each k

- consider the Taylor development of the residual vector at θ_k

$$r(\theta) = r(\theta_k) + r'(\theta_k)(\theta - \theta_k) + \|\theta - \theta_k\|\varepsilon(\theta - \theta_k)$$

  and take θ_{k+1} such that the squared norm of the affine approximation

$$\|r(\theta_k) + r'(\theta_k)(\theta_{k+1} - \theta_k)\|^2$$

  is minimal.

finding $\theta_{k+1} - \theta_k$ is a LLS problem !

Non Linear Least Squares (NLLS)

Gauss-Newton method

- Original formulation of the Gauss-Newton method

$$\theta_{k+1} = \theta_k - \left[r'(\theta_k)^\top r'(\theta_k)\right]^{-1} r'(\theta_k)^\top r(\theta_k),$$

- Equivalent Scilab implementation using backslash \ operator

$$\theta_{k+1} = \theta_k - r'(\theta_k)\backslash r(\theta_k)$$

Problem: what can you do when $r'(\theta_k)$ has not full column rank ?

Non Linear Least Squares (NLLS)

Levenberg-Marquardt method

- Modify the Gauss-Newton iteration: pick up a $\lambda > 0$ and take θ_{k+1} such that

$$S_\lambda(\theta_{k+1} - \theta_k) = \|r(\theta_k) + r'(\theta_k)(\theta_{k+1} - \theta_k)\|^2 + \lambda\|(\theta_{k+1} - \theta_k)\|^2$$

  is minimal.

- After rewriting $S_\lambda(\theta_{k+1} - \theta_k)$ using block matrix notation as

$$S_\lambda(\theta_{k+1} - \theta_k) = \left\| \begin{pmatrix} r'(\theta_k) \\ \lambda^{\frac{1}{2}} \mathrm{I} \end{pmatrix} (\theta_{k+1} - \theta_k) + \begin{pmatrix} r(\theta_k) \\ \mathbf{0} \end{pmatrix} \right\|^2$$

finding $\theta_{k+1} - \theta_k$ is a LLS problem and for any $\lambda > 0$ a unique solution exists !

Non Linear Least Squares (NLLS)

Levenberg-Marquardt method

- Since the residual vector reads

$$\begin{pmatrix} r'(\theta_k) \\ \lambda^{\frac{1}{2}}\mathrm{I} \end{pmatrix} (\theta_{k+1} - \theta_k) + \begin{pmatrix} r(\theta_k) \\ \mathbf{0} \end{pmatrix}$$

the normal equations of the LLS are given by

$$\begin{pmatrix} r'(\theta_k) \\ \lambda^{\frac{1}{2}}\mathrm{I} \end{pmatrix}^\top \begin{pmatrix} r'(\theta_k) \\ \lambda^{\frac{1}{2}}\mathrm{I} \end{pmatrix} (\theta_{k+1} - \theta_k) = -\begin{pmatrix} r'(\theta_k) \\ \lambda^{\frac{1}{2}}\mathrm{I} \end{pmatrix}^\top \begin{pmatrix} r(\theta_k) \\ \mathbf{0} \end{pmatrix}$$

$$\iff \quad \left(r'(\theta_k)^\top, \lambda^{\frac{1}{2}}\mathrm{I}\right) \begin{pmatrix} r'(\theta_k) \\ \lambda^{\frac{1}{2}}\mathrm{I} \end{pmatrix} (\theta_{k+1} - \theta_k) = -\left(r'(\theta_k)^\top, \lambda^{\frac{1}{2}}\mathrm{I}\right) \begin{pmatrix} r(\theta_k) \\ \mathbf{0} \end{pmatrix}$$

$$\iff \quad \left(r'(\theta_k)^\top r'(\theta_k) + \lambda\mathrm{I}\right)(\theta_{k+1} - \theta_k) = -r'(\theta_k)^\top r(\theta_k)$$

Non Linear Least Squares (NLLS)

Levenberg-Marquardt method

- Hence, the mathematical formulation of Levenberg-Marquardt method is

$$\theta_{k+1} = \theta_k - [r'(\theta_k)^\top r'(\theta_k) + \lambda \mathrm{I}]^{-1} r'(\theta_k)^\top r(\theta_k)$$

but practical Scilab implementation should use the backslash \ operator

$$\theta_{k+1} = \theta_k - \begin{pmatrix} r'(\theta_k) \\ \lambda^{\frac{1}{2}} \mathrm{I} \end{pmatrix} \backslash \begin{pmatrix} r(\theta_k) \\ \mathbf{0} \end{pmatrix}$$

Non Linear Least Squares (NLLS)

Levenberg-Marquardt method

Where is the insight in Levenberg-Marquardt method ?

- Remember that $\nabla S(\theta) = 2r'(\theta)^\top r(\theta)$, hence LM iteration reads

$$\begin{aligned}\theta_{k+1} &= \theta_k - \tfrac{1}{2}\left(r'(\theta_k)^\top r'(\theta_k) + \lambda \mathrm{I}\right)^{-1} \nabla S(\theta_k),\\ &= \theta_k - \tfrac{1}{2\lambda}\left(\tfrac{1}{\lambda} r'(\theta_k)^\top r'(\theta_k) + \mathrm{I}\right)^{-1} \nabla S(\theta_k)\end{aligned}$$

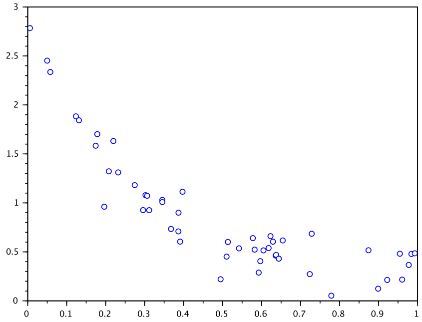
  - When λ is small, LM methods behaves more like the Gauss-Newton method.
  - When λ is large, LM methods behaves more like the gradient method.

λ allows to balance between speed ($\lambda = 0$) and robustness ($\lambda \to \infty$)

Non Linear Least Squares (NLLS)

Example 1

Consider data (x_i, y_i) to be fitted by the non linear model $f_\theta(x) = \exp(\theta_1 + \theta_2 x)$:



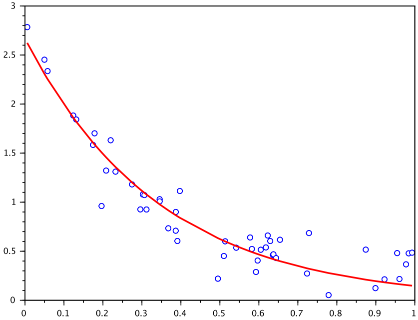
The Jacobian of $r(\theta)$ is given by

$$r'(\theta) = \begin{bmatrix} \exp(\theta_1 + \theta_2 x_1) & x_1 \exp(\theta_1 + \theta_2 x_1) \\ \vdots & \vdots \\ \exp(\theta_1 + \theta_2 x_n) & x_n \exp(\theta_1 + \theta_2 x_1) \end{bmatrix}$$

Non Linear Least Squares (NLLS)

Example 1

In Scilab, use the lsqrsolve or leastsq function:



$\hat{\theta}$ = (0.981, -2.905)

```
function r=resid(theta,n)
  r=exp(theta(1)+theta(2)*x)-y;
endfunction

function j=jac(theta,n)
  e=exp(theta(1)+theta(2)*x);
  j=[e x.*e];
endfunction

load data_exp.dat
theta0=[0;0];
theta=lsqrsolve(theta0,resid,length(x),jac);

plot(x,y,"ob", x,exp(theta(1)+theta(2)*x),"r")
```

Non Linear Least Squares (NLLS)

Example 2

- Enzymatic kinetics

$$s'(t) = \theta_2 \frac{s(t)}{s(t) + \theta_3},\ t > 0,$$
$$s(0) = \theta_1,$$

y_i = measurement of s at time t_i

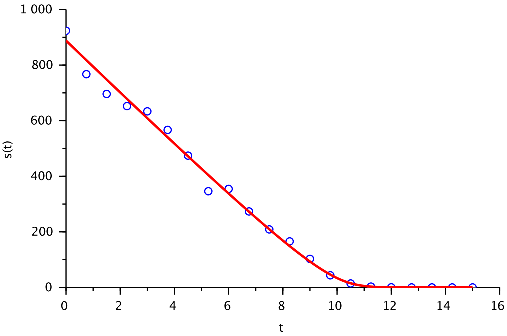
$$S(\theta) = \|r(\theta)\|^2, \quad r_i(\theta) = \frac{y_i - s(t_i)}{\sigma_i}$$

Individual weights σ_i allow to take into account different standard deviations of measurements

Non Linear Least Squares (NLLS)

Example 2

In Scilab, use the lsqrsolve or leastsq function



$\hat{\theta} = (887.9, 37.6, 97.7)$

```
function dsdt=michaelis(t,s,theta)
  dsdt=theta(2)*s/(s+theta(3))
endfunction

function r=resid(theta,n)
  s=ode(theta(1),0,t,michaelis)
  r=(s-y)./sigma
endfunction

load michaelis_data.dat
theta0=[y(1);20;80];
theta=lsqrsolve(theta0,resid,n)
```

If not provided, the Jacobian $r'(\theta)$ is approximated by finite differences (but true Jacobian always speed up convergence).

Non Linear Least Squares (NLLS)

Take home message

Take home message #3 :

Solving non linear least squares problems is not that difficult
with adequate software and good starting values

Statistical evaluation of solutions

1. Motivation and statistical framework
2. Maths reminder
3. Linear Least Squares (LLS)
4. Non Linear Least Squares (NLLS)
5. **Statistical evaluation of solutions**
6. Model selection

Statistical evaluation of solutions

Motivation

- Since the data $(y_i)_{i=1\dots n}$ is a sample of random variables, then $\hat{\theta}$ too !
- Confidence intervals for $\hat{\theta}$ can be easily obtained by at least two methods
    - Monte-Carlo method : allows to estimate the distribution of $\hat{\theta}$ but needs thousands of resamplings
    - Linearized statistics : very fast, but can be very approximate for high level of measurement error

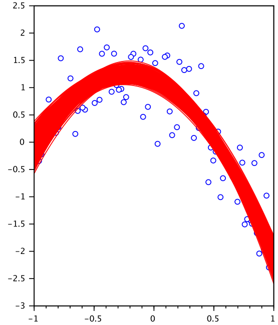
Statistical evaluation of solutions

Monte Carlo method

- The Monte Carlo method is a resampling method, i.e. works by generating new samples of synthetic measurement and redoing the estimation of $\hat{\theta}$. Here model is

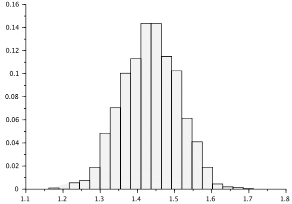
$$y = \theta_1 + \theta_2 x + \theta_3 x^2,$$

and data is corrupted by noise with $\sigma = \frac{1}{2}$

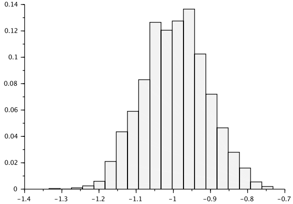


Statistical evaluation of solutions

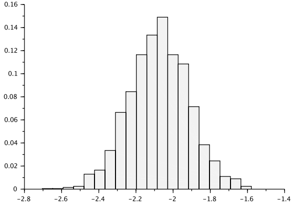
Monte Carlo method



θ_1



θ_2



θ_3

At confidence level=95%,

$$\hat{\theta}_1 \in [0.99, 1.29],$$
$$\hat{\theta}_2 \in [-1.20, -0.85],$$
$$\hat{\theta}_1 \in [-2.57, -1.91].$$

Statistical evaluation of solutions

Linearized Statistics

- Define the weighted residual $r(\theta)$ by

$$r_i(\theta) = \frac{y_i - f_\theta(x_i)}{\sigma_i},$$

  where σ_i is the standard deviation of y_i.
- The covariance matrix of $\hat{\theta}$ can be approximated by

$$V(\hat{\theta}) = F(\hat{\theta})^{-1}$$

  where $F(\hat{\theta})$ is the Fisher Information Matrix, given by

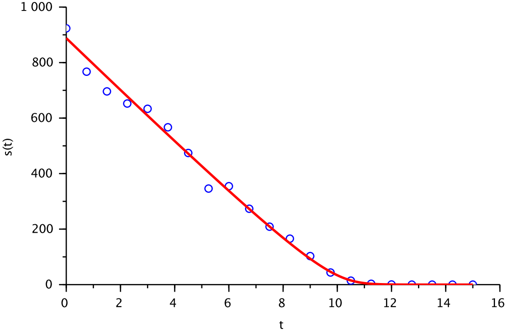
$$F(\theta) = r'(\theta)^\top r'(\theta)$$

- For example, when $\sigma_i = \sigma$ for all i, in LLS problems

$$V(\hat{\theta}) = \sigma^2 A^\top A$$

Statistical evaluation of solutions

Linearized Statistics



$\hat{\theta}$ = (887.9, 37.6, 97.7)

```
d=derivative(resid,theta)
V=inv(d'*d)
sigma_theta=sqrt(diag(V))

// 0.975 fractile Student dist.

t_alpha=cdft("T",m-3,0.975,0.025);

thetamin=theta-t_alpha*sigma_theta
thetamax=theta+t_alpha*sigma_theta
```

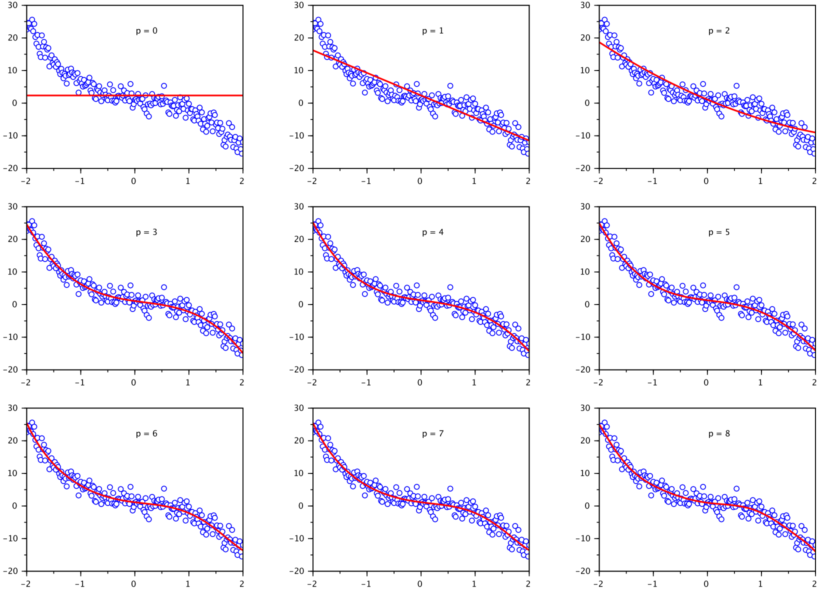
At 95% confidence level

$$\hat{\theta}_1 \in [856.68, 919.24], \quad \hat{\theta}_2 \in [34.13, 41.21], \quad \hat{\theta}_3 \in [93.37, 102.10].$$

Statistical evaluation of solutions

Model selection

Motivation : which model is the best ?



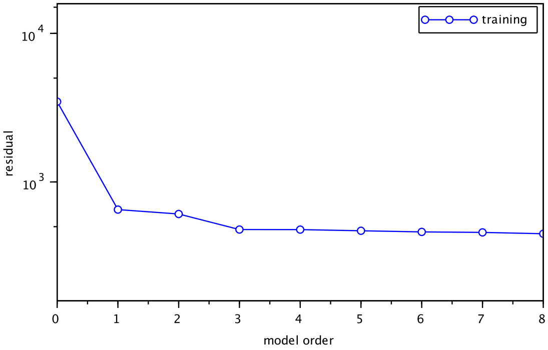
Model selection

Motivation : which model is the best ?

On the previous slide data has been fitted with the model

$$y = \sum_{k=0}^{p} \theta_k x^k, \quad p = 0 \dots 8,$$

Consider $S(\hat{\theta})$ as a function of model order p does not help much



p	$S(\hat{\theta})$
0	3470.32
1	651.44
2	608.53
3	478.23
4	477.78
5	469.20
6	461.00
7	457.52
8	448.10

Model selection

Validation

Validation is the key of model selection :

1. Define two sets of data
   - $T \subset \{1, \dots n\}$ for model training
   - $V = \{1, \dots n\} \setminus T$ for validation

2. For each value of model order p
   - Compute the optimal parameters with the training data

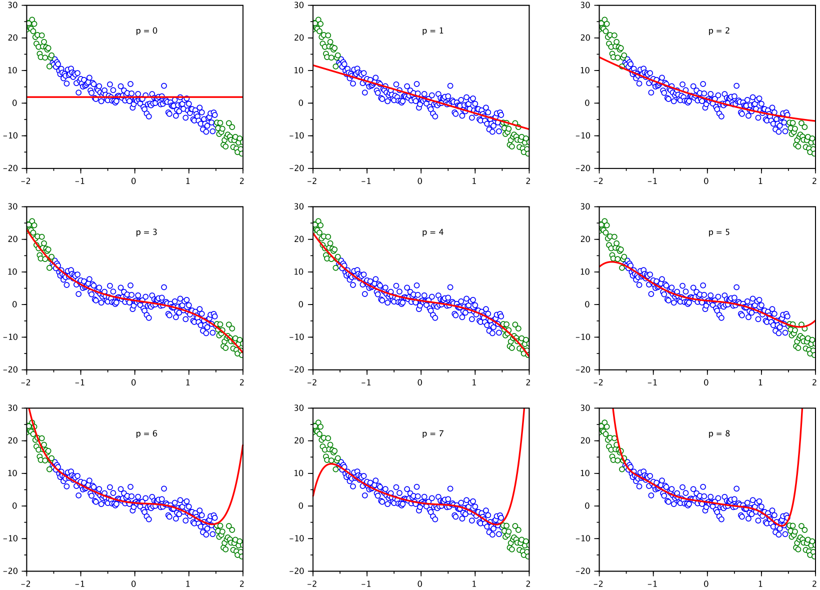
$$\hat{\theta}_p = \arg\min_{\theta \in \mathbb{R}^p} \sum_{i \in T} (y_i - f_\theta(x_i))^2$$

   - Compute the validation residual

$$S_V(\hat{\theta}_p) = \sum_{i \in V} (y_i - f_{\hat{\theta}_p}(x_i))^2$$

Model selection

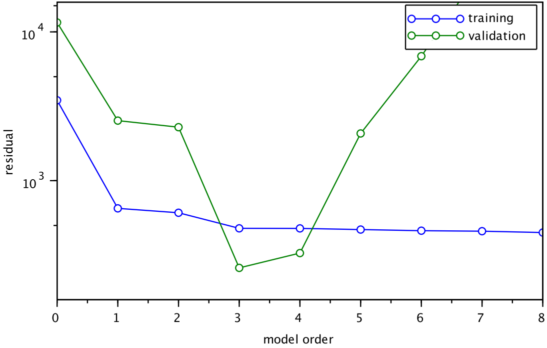
Training + Validation



Model selection

Training + Validation

Validation helps a lot: here the best model order is clearly $p = 3$!



p	$S_V(\hat{\theta}_p)$
0	11567.21
1	2533.41
2	2288.52
3	259.27
4	326.09
5	2077.03
6	6867.74
7	26595.40
8	195203.35

Take home message #4 :

Always evaluate your models by either computing confidence intervals for the parameters or by using validation.