Least squares problems
How to state and solve them, then evaluate their solutions

Stéphane Mottelet

Université de Technologie de Compiègne

30 septembre 2016
Outline

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

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

Data: \((x_i, y_i)_{i=1..n}\)

Model: \(y = f(x; \theta)\)

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

Regression problem

Find \(\theta\) such that the model best explains the data,

i.e. \(y_i\) close to \(f(x_i, \theta), i = 1 \ldots n\).
Motivation

Regression problem, example

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

\[ y \rightarrow \text{find } \theta_1, \theta_2 \text{ such that the data fits the model } y = \theta_1 + \theta_2 x \]

How does one measure the fit/misfit?
Motivation

Least squares method

The least squares method measures the fit with the function

\[ S(\theta) = \sum_{i=1}^{n} (y_i - f(x_i; \theta))^2, \]

and aims to find \( \hat{\theta} \) such that

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

or equivalently

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

Important issues

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

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

\[ y_i = f(x_i; \theta) + \varepsilon_i, \]

where \(\varepsilon_i\) is a random variable with

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

The probability density of \(y_i\) is given by

\[ \phi_i : \mathbb{R} \times \mathbb{R}^p \longrightarrow \]

\[ (y, \theta) \longrightarrow \phi_i(y; \theta) = g(y - f(x_i; \theta)) \]

and \(E[y_i|\theta] = f(x_i; \theta)\)
Statistical framework

Example

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

$$
\phi_i(y; \theta) = (\sigma \sqrt{2\pi})^{-1} \exp\left(-\frac{1}{2\sigma^2} (y - f(x_i; \theta))^2\right)
$$
Statistical framework

Joint probability density and Likelihood function

- **Joint density**

  When \( \theta \) is given, as the \((y_i)\) are independent, the density of the whole vector \( y = (y_1, \ldots, y_n) \) is

  \[
  \phi(y; \theta) = \prod_{i=1}^{n} \phi_i(y_i; \theta).
  \]

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

  \[
  \text{Prob}[y \in D|\theta] = \int_{D} \phi(y; \theta) \, dy_1 \ldots dy_n
  \]

- **Likelihood function**

  When a sample of \( y \) is given, then \( L(\theta; y) = \phi(y; \theta) \) is called

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

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

Under the Gaussian hypothesis, then

$$L(\theta; y) = \prod_{i=1}^{n} (\sigma \sqrt{2\pi})^{-1} \exp \left( -\frac{1}{2\sigma^2} (y - f(x_i; \theta))^2 \right),$$

$$= (\sigma \sqrt{2\pi})^{-n} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y - f(x_i; \theta))^2 \right),$$

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

$$\arg\max_{\theta \in \mathbb{R}^p} L(\theta; y) = \arg\min_{\theta \in \mathbb{R}^p} S(\theta).$$
Least Absolute Deviation Regression: the misfit is measured by

\[ S_1(\theta) = \sum_{i=1}^{n} |y_i - f(x_i; \theta)|. \]

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

Yes, if \( \varepsilon_i \) 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
Densities of Gaussian vs. Laplacian random variables (with zero mean and unit variance):

Second issue: the two statistical hypothesis are very different!
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

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
Maths reminder
Matrix algebra

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

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

- 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

\[
I = \begin{pmatrix} 1 & \cdots \\ \vdots \\ 1 \end{pmatrix}
\]
Transposition, Inner product and norm:

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

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

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

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

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 $A^{-1}$)
  \[ \text{in Scilab/Matlab: } x = A \backslash y \]
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 

\[ f(a + h) = f(a) + J_f(a)h + \|h\|\varepsilon(h), \quad \lim_{h \to 0} \varepsilon(h) = 0 \]

- **Jacobian matrix, partial derivatives**:

  \[ [J_f(a)]_{ij} = \frac{\partial f_i}{\partial x_j}(a) \]

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

  \[ f(a + h) = f(a) + \nabla f(a)^\top h + \|h\|\varepsilon(h), \quad \lim_{h \to 0} \varepsilon(h) = 0 \]
When $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, 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 $x_k$

$$T(x) = f(x_k) + J_f(x_k)(x - x_k)$$

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

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

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 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 $\rho_k$, 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)

Motivation and statistical framework
Maths reminder
**Linear Least Squares (LLS)**
Non Linear Least Squares (NLLS)
Statistical evaluation of solutions
The model \( y = f(x; \theta) \) is linear w.r.t. \( \theta \), i.e.

\[
y = \sum_{k=1}^{p} \theta_k \phi_k(x), \quad \phi_k : \mathbb{R} \to \mathbb{R}
\]

Examples

- \( y = \sum_{k=1}^{p} \theta_k x^{k-1} \)
- \( y = \sum_{k=1}^{p} \theta_k \cos \frac{(k-1)x}{T} \), where \( T = x_n - x_1 \)
- ...
Linear Least Squares

Simple linear regression

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

Residual vector \( r(\theta) \)

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

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

Optimality conditions

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

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

- Necessary optimality condition

\[
\nabla S(\hat{\theta}) = 0
\]

Compute the gradient by expanding $S(\theta)$
\[ S(\theta + h) = \| A(\theta + h) - y \|^2 = \| A\theta - y + Ah \|^2 \]
\[ = (A\theta - y + Ah)^\top (A\theta - y + Ah) \]
\[ = (A\theta - y)^\top (A\theta - y) + (A\theta - y)^\top Ah + (Ah)^\top (A\theta - y) + (Ah)^\top Ah \]
\[ = \| A\theta - y \|^2 + 2(A\theta - y)^\top Ah + \| Ah \|^2 \]
\[ = S(\theta) + \nabla S(\theta)^\top h + \| Ah \|^2 \]
\[ \nabla S(\theta) = 2A^\top (A\theta - y) \]
**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**:

$$S(\theta) = S(\hat{\theta} + \theta - \hat{\theta}) = S(\hat{\theta}) + \nabla S(\hat{\theta})^\top (\theta - \hat{\theta}) + \|A(\theta - \hat{\theta})\|^2,$$

$$= S(\hat{\theta}) + \|A(\theta - \hat{\theta})\|^2,$$

$$\geq S(\hat{\theta})$$

**Uniqueness**:

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

$$\iff A(\theta - \hat{\theta}) = 0$$

$$\iff \theta = \hat{\theta}.$$
Linear Least Squares
Simple linear regression

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

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

\[ A^\top A = \left[ \begin{array}{cc}
 n & S_x \\
 S_x & S_{xx}
\end{array} \right], \quad A^\top y = \left[ \begin{array}{c} S_y \\
 S_{xy} \end{array} \right] \]

\[ \theta_1 = \frac{S_y S_{xx} - S_x S_{xy}}{nS_{xx} - S_x^2}, \quad \theta_2 = \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x^2} \]
Linear Least Squares
Practical resolution, Scilab or Matlab

\[ A = [\text{ones}(x), x, x^2] \]

\[ \theta = A \backslash y \]

\[ \text{plot}(x, y, "o", x, A \ast \theta, x, "r") \]
Linear Least Squares
Practical resolution, Scilab or Matlab

\[ A = [\text{ones}(x), x, x.\ ^2] \]

\[ \theta = A \backslash y \]

\[ \text{plot}(x, y, "o", x, A*\theta, x, "r") \]
Linear Least Squares
An interesting example

- Find a circle which best fits \((x_i, y_i)_{i=1}^{n}\) in the plane
- 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 is non-linear w.r.t. \((a, b, R)\) but

\[
r_i = R^2 - a^2 - b^2 + 2ax_i + 2by_i - (x_i^2 + y_i^2),
\]

\[
= [2x_i, 2y_i, 1] \begin{bmatrix} a \\ b \\ R^2 - a^2 - b^2 \end{bmatrix} - (x_i^2 + y_i^2)
\]

is linear w.r.t. \((a, b, R^2 - a^2 - b^2) = \theta\).
Linear Least Squares

An interesting example

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

- **In Scilab**

```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))
```

Stéphane Mottelet (UTC)
Linear Least Squares

An interesting example

- **Standard form**

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

- **In Scilab**

```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)

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
Non Linear Least Squares (NLLS)

Example

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

\[ y = f(x; \theta) = \exp(\theta_1 + \theta_2 x), \]

The “log trick” leads some people to minimize

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

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

if \(y_i - f(x_i; \theta)\) is Gaussian then \(\log y_i - \log f(x_i; \theta)\) is not!
Non Linear Least Squares (NLLS)

Possibles angles of attack

Remind that

\[ S(\theta) = \| r(\theta) \|^2, \quad r_i(\theta) = f(x_i; \theta) - y_i. \]

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

- Compute a solution of the non linear systems of equations

\[ \nabla S(\theta) = 2J_r(\theta)^\top r(\theta) = 0, \]

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
Use the spirit of Newton’s method as follows: start with $\theta_0$ and for each $k$

- consider the development of the residual vector at $\theta_k$ with $h = \theta - \theta_k$

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

and take $\theta_{k+1}$ as the vector minimizing

$$S_k(\theta) = \|r(\theta_k) + J_r(\theta_k)(\theta - \theta_k)\|^2$$

finding $\theta_{k+1}$ is a LLS problem!
Non Linear Least Squares (NLLS)

Gauss Newton method

- **Original formulation**

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

\[
= \theta_k - \frac{1}{2} [J_r(\theta_k) J_r(\theta_k)]^{-1} \nabla S(\theta_k)
\]

What can you do when the rank of \( J_r(\theta_k) \) is deficient? Pick up a \( \lambda > 0 \) and take \( \theta_{k+1} \) as the vector minimizing

\[
\| r(\theta_k) + J_r(\theta_k)(\theta - \theta_k) \|^2 + \lambda \| (\theta - \theta_k) \|^2
\]

- **Levenberg-Marquardt method**

\[
\theta_{k+1} = \theta_k - \frac{1}{2} [J_r(\theta_k) J_r(\theta_k) + \lambda I]^{-1} \nabla S(\theta_k)
\]

\( \lambda \) allows to balance between speed \( (\lambda = 0) \) and robustness \( (\lambda \to \infty) \)
Consider again data \((x_i, y_i)\) to be fitted by the non linear model

\[
f(x; \theta) = \exp(\theta_1 + \theta_2 x),
\]

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

\[
J_r(\theta) = \begin{bmatrix}
\exp(\theta_1 + \theta_2 x_1) & x_1 \exp(\theta_1 + \theta_2 x_1) \\
\vdots & \vdots \\
\exp(\theta_1 + \theta_2 x_n) & x_n \exp(\theta_1 + \theta_2 x_1)
\end{bmatrix}
\]
Non Linear Least Squares (NLLS)

Example 1

- In Scilab, use the *lsqrsolve* or *leastsq* function

\[ \hat{\theta} = (1.001, -1.969) \]

```scilab
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,30,jac);
plot(x,y,"ob",...
x,exp(theta(1)+theta(2)*x),"r")
```
Non Linear Least Squares (NLLS)

Example 2

- **Enzymatic kinetics**

  \[ s'(t) = \theta_2 \frac{s(t)}{s(t) + \theta_3}, \quad t > 0, \]

  \[ s(0) = \theta_1, \]

  \[ y_i = \text{measurement of } s \text{ at time } t_i \]

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

  Individual weighting allows to take into account different std. deviations of measurements
Non Linear Least Squares (NLLS)

Example 2

- In Scilab, use the `lsqrsolve` or `leastsq` function

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

```scilab
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 \(J_r(\theta)\) is approximated by finite differences (but analytical derivatives always speed up convergence)
Take home message #3:

Solving non linear least squares problems is not that difficult with adequate software and good starting values.
Motivation and statistical framework
Maths reminder
Linear Least Squares (LLS)
Non Linear Least Squares (NLLS)
Statistical evaluation of solutions
Model selection
Statistical evaluation of solutions

Motivation

- Since the data \((y_i)_{i=1...n}\) 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
The Monte Carlo 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

At confidence level=95%,

\[ \hat{\theta}_1 \in [0.99, 1.29], \]
\[ \hat{\theta}_2 \in [-1.20, -0.85], \]
\[ \hat{\theta}_3 \in [-2.57, -1.91]. \]
Define the weighted residual $r(\theta)$ by

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

where $\sigma_i$ 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

$$F(\theta) = J_r(\theta)^\top J_r(\theta)$$

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

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

\[ d = \text{derivative}(\text{resid}, \theta) \]
\[ V = \text{inv}(d' \times d) \]
\[ \sigma_{\theta} = \text{sqrt}(\text{diag}(V)) \]

// 0.975 fractile Student dist.
\[ t_{\alpha} = \text{cdft}("T", m-3, 0.975, 0.025); \]
\[ \text{thetamin} = \theta - t_{\alpha} \times \sigma_{\theta} \]
\[ \text{thetamax} = \theta + t_{\alpha} \times \sigma_{\theta} \]

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

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
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

\[ y = \sum_{k=0}^{p} \theta_k x^k, \quad p = 0 \ldots 8, \]

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

<table>
<thead>
<tr>
<th>( p )</th>
<th>( S(\hat{\theta}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>114.48939</td>
</tr>
<tr>
<td>1</td>
<td>68.882302</td>
</tr>
<tr>
<td>2</td>
<td>23.420028</td>
</tr>
<tr>
<td>3</td>
<td>22.879069</td>
</tr>
<tr>
<td>4</td>
<td>22.780545</td>
</tr>
<tr>
<td>5</td>
<td>22.77985</td>
</tr>
<tr>
<td>6</td>
<td>22.771418</td>
</tr>
<tr>
<td>7</td>
<td>22.440983</td>
</tr>
<tr>
<td>8</td>
<td>22.268065</td>
</tr>
</tbody>
</table>
Validation is the key of model selection:

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

2. For each value of model order \( p \)
   - Compute the optimal parameters with the training data
     \[
     \hat{\theta}_p = \arg\min_{\theta \in \mathbb{R}^p} \sum_{i \in T} (y_i - f(x_i; \theta))^2
     \]
   - Compute the validation residual
     \[
     S_V(\hat{\theta}_p) = \sum_{i \in V} (y_i - f(x_i; \hat{\theta}_p))^2
     \]
Validation helps a lot: here the best model order is clearly $p = 3$!

<table>
<thead>
<tr>
<th>$p$</th>
<th>$S_V(\hat{\theta}_p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>62.696525</td>
</tr>
<tr>
<td>1</td>
<td>40.303186</td>
</tr>
<tr>
<td>2</td>
<td>4.5989513</td>
</tr>
<tr>
<td>3</td>
<td>4.2074337</td>
</tr>
<tr>
<td>4</td>
<td>7.8923051</td>
</tr>
<tr>
<td>5</td>
<td>9.2838043</td>
</tr>
<tr>
<td>6</td>
<td>8.690164</td>
</tr>
<tr>
<td>7</td>
<td>73.116871</td>
</tr>
<tr>
<td>8</td>
<td>410.55303</td>
</tr>
</tbody>
</table>
Take home message #4:

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