02610
Optimization and Data Fitting

Week 6: Data Fitting and Linear Least-Squares Problems

Yiqiu Dong

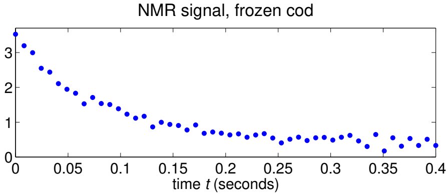
DTU Compute
Technical University of Denmark

Lecture Materials

- J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd Edition, Springer.
  - Chapter 10: Least-squares problems.
  - We cover: 10.1 and 10.2.
- P. C. Hansen, V. Pereyra and G. Scherer, *Least Squares Data Fitting with Applications*, Johns Hopkins University Press.
  - Chapter 1: The linear data fitting problem.
  - We cover: 1.1, 1.2 and 1.4.
  - Chapter 2: The linear least squares problem.
  - We cover: 2.1 and 2.2.

Example: Parameter estimation

NMR techniques can be used to examine the molecular structure and distinguish different isotopes. Here is an NMR signal from frozen cod:



Data: Noisy measurements of the NMR signal from frozen cod at different time.

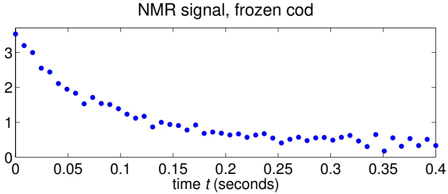
Model: The underlying mathematical model for the NMR signal is:

$$F(t) = c_0 + c_1 e^{-\lambda_1 t} + c_2 e^{-\lambda_2 t},$$

- t denotes the time and λ_1 and λ_2 are known.
- c_1 and c_2 are proportional to the amount of water containing the two kinds of protons.
- c_0 accounts for an undesired background (bias) in the measurements.

Example: Parameter estimation

NMR techniques can be used to examine the molecular structure and distinguish different isotopes. Here is an NMR signal from frozen cod:



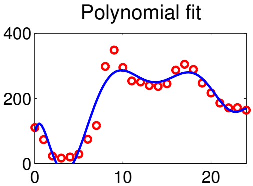
Data: Noisy measurements of the NMR signal from frozen cod at different time.

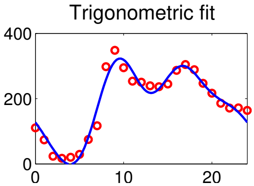
Model: The underlying mathematical model for the NMR signal is:

$$F(t) = c_0 + c_1 e^{-\lambda_1 t} + c_2 e^{-\lambda_2 t},$$

Goal: Estimate the unknown parameters c_0, c_1, c_2 in the model, and then compute the different kinds of water contents in the cod sample.

Example: Data approximation





Data: Measurements of air pollution, in the form of the NO concentration, over a period of 24 hours, at H. C. Andersens Boulevard on a Thursday.

Model: Polynomial and periodic models:

$$F(t) = c_0 + c_1 t + c_2 t^2 + \cdots + c_n t^n$$

$$F(t) = c_0 + c_1 \sin(\omega t) + c_2 \cos(\omega t) + c_3 \sin(2\omega t) + c_4 \cos(2\omega t) + \cdots$$

Goal: Fit a smooth curve to the measurements, so that we can compute the concentration at an arbitrary time between 0 and 24 hours.

Data Fitting

Given: data (t_i, y_i) with measurement errors.

We want: to fit a model – a function $\phi(t)$ – to these data.

Requirement: $\phi(t)$ captures the "overall behavior" of the data without being too sensitive to the errors.

Remark

- Data fitting is distinctly different from *interpolation*, where we seek a model that interpolates the given data, i.e., it satisfies $\phi(t_i) = y_i$ for all the data points.
- In data fitting approach there are *more data than unknown parameters*, which helps to decrease the uncertainty in the parameters of the model.
- we usually assume that the abscissas t_i appear in non-decreasing order, i.e.,

$$t_1 \leq t_2 \leq \cdots \leq t_m.$$

The Linear Data Fitting Problem

We wish to compute an approximation to the data, given by the *fitting model* $\phi(\boldsymbol{x}; t)$.

The vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$ contains n parameters that characterize the model, which are to be determined from the given noisy data.

The linear data fitting problem:

$$\boxed{\text{Linear fitting model:} \quad \phi(\boldsymbol{x}; t) = \sum_{j=1}^{n} x_j\, f_j(t).}$$

- The functions $f_j(t)$ are called *basis functions*, and are chosen either because the reflect the underlying physical/chemical/... model or because they are "easy" to work with.
- The *order* n should be somewhat smaller than the number m of data points.

The Residual

We introduce the residual r_i which shows the difference between y_i and the fit function value at t_i:

$$r_i = y_i - \phi(\boldsymbol{x}; t_i), \qquad i = 1, 2, \ldots, m \ .$$

r_i is the residual for the data pair (t_i, y_i).

Remark:

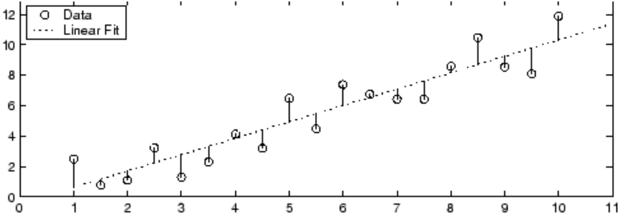
- Each residual is a function of the parameter vector $\boldsymbol{x}$, i.e., $r_i = r_i(\boldsymbol{x})$.
- The absolute value of the residual, $|r_i|$, is the vertical distance between the known data and the fit function.

The Residual

We introduce the residual r_i which shows the difference between y_i and the fit function value at t_i:

$$r_i = y_i - \phi(\boldsymbol{x}; t_i), \qquad i = 1, 2, \ldots, m \; .$$

r_i is the residual for the data pair (t_i, y_i).



A Good Fit Based on Residual

Two commonly used criteria for choosing the "good" fit

$$\min_{\boldsymbol{x}} \sum_{i=1}^{m} |r_i(\boldsymbol{x})| = \min_{\boldsymbol{x}} \sum_{i=1}^{m} \big| y_i - \phi(\boldsymbol{x}, t_i) \big|$$

$$\text{or} \qquad \min_{\boldsymbol{x}} \sum_{i=1}^{m} r_i(\boldsymbol{x})^2 = \min_{\boldsymbol{x}} \sum_{i=1}^{m} (y_i - \phi(\boldsymbol{x}, t_i))^2.$$

- Sum-of-absolute-values of the residuals makes the problem harder to solve.
- Sum-of-squares of the residuals is much easy to solve.
- If the data errors follow a normal probability distribution, then the second minimization problem produces a best estimate of ϕ.

The Least Squares (LSQ) Fit

A least squares fit is a choice of the parameter vector $\boldsymbol{x}$ that minimizes the sum-of-squares of the residuals:

$$\boxed{\text{LSQ fit:} \quad \min_{\boldsymbol{x}} \sum_{i=1}^{m} r_i(\boldsymbol{x})^2 = \min_{\boldsymbol{x}} \sum_{i=1}^{m} \big(y_i - \phi(\boldsymbol{x}; t_i)\big)^2.}$$

Given: The data (t_i, y_i) for $i = 1, \ldots, m$ with measurement errors.

Goal: Find a fit function (model) $\phi(t)$, i.e. find the coefficients $\boldsymbol{x}$ in ϕ, by solving LSQ fit problem.

The Underlying Idea

Assume our underlying data model:

$$y_i = \Gamma(t_i) + e_i, \qquad i = 1, 2, \dots, m.$$

- $\Gamma(t)$ is called as *pure-data function*.
- $e_1, e_2, \cdots, e_m$ are the data errors (we may have some statistical information about them).

The Underlying Idea

Assume our underlying data model:

$$y_i = \Gamma(t_i) + e_i, \qquad i = 1, 2, \ldots, m.$$

Consider the residuals for $i = 1, \ldots, m$:

$$\begin{aligned} r_i &= y_i - \phi(\boldsymbol{x}; t_i) \\ &= \big(y_i - \Gamma(t_i)\big) + \big(\Gamma(t_i) - \phi(\boldsymbol{x}; t_i)\big) \\ &= e_i + \big(\Gamma(t_i) - \phi(\boldsymbol{x}; t_i)\big) \ . \end{aligned}$$

- The *data error* e_i is from the measurements.
- The *approximation error* $\Gamma(t_i) - \phi(\boldsymbol{x}; t_i)$ is due to the discrepancy between the pure-data function and the fitting model.

A good fitting model $\phi(\boldsymbol{x}, t)$ is one for which the residuals are of the same size as the data errors.

Continue...

With a good fit, we expect that

$$r_i = y_i - \phi(\boldsymbol{x}; t_i) = e_i \quad \text{for } i = 1, \cdots, m.$$

It often is reasonable to assume that e_i's are independent and identically *normally* distributed with mean 0 and variance ς^2, i.e., the probability density function of e is

$$g_\varsigma(e) = \frac{1}{\sqrt{2\pi\varsigma^2}} \exp\left(-\frac{e^2}{2\varsigma^2}\right).$$

Continue...

The likelihood of a particular set of data y_i, $i = 1, \cdots, m$, with given parameters $\boldsymbol{x}$ is shown as

$$p(\boldsymbol{y}|\boldsymbol{x}, \varsigma) = \prod_{i=1}^{m} g_\varsigma(e_i) = \prod_{i=1}^{m} g_\varsigma(y_i - \phi(\boldsymbol{x}; t_i)).$$

Continue...

The likelihood of a particular set of data y_i, $i = 1, \cdots, m$, with given parameters $\boldsymbol{x}$ is shown as

$$p(\boldsymbol{y}|\boldsymbol{x}, \varsigma) = (2\pi\varsigma^2)^{-m/2} \exp\left(-\frac{1}{2\varsigma^2}\sum_{i=1}^{m}\left(y_i - \phi(\boldsymbol{x}; t_i)\right)^2\right)$$

Maximum likelihood estimate is the "most likely" value of $\boldsymbol{x}$ which is obtained by maximizing $p(\boldsymbol{y}|\boldsymbol{x}, \varsigma)$ with given $y_1, \cdots, y_m$, i.e.,

$$\begin{aligned} & \max_{\boldsymbol{x}} \; p(\boldsymbol{y}|\boldsymbol{x}, \varsigma) \\ \Longleftrightarrow & \min_{\boldsymbol{x}} \; -\ln p(\boldsymbol{y}|\boldsymbol{x}, \varsigma) \\ \Longleftrightarrow & \min_{\boldsymbol{x}} \; \frac{1}{2\varsigma^2}\sum_{i=1}^{m}[y_i - \phi(\boldsymbol{x}; t_i)]^2. \end{aligned}$$

Linear LSQ Data Fitting Problems

Consider a linear fit model

$$\phi(\boldsymbol{x};t) = \sum_{j=1}^{n} x_j\, f_j(t), \qquad f_j(t) = \text{given basis functions.}$$

The linear least squares data fitting problem is

$$\begin{aligned}\min_{\boldsymbol{x}} \sum_{i=1}^{m} r_i(\boldsymbol{x})^2 &= \min_{\boldsymbol{x}} \sum_{i=1}^{m} \big(y_i - \phi(\boldsymbol{x};t_i)\big)^2 \\ &= \min_{\boldsymbol{x}} \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} x_j\, f_j(t_i) \right)^2\end{aligned}$$

Matrix-Vector Notation

Define the matrix $A \in \mathbb{R}^{m\times n}$, the vectors $\mathrm{y}, \mathrm{r} \in \mathbb{R}^m$ and $\boldsymbol{x} \in \mathbb{R}^n$ as follows,

$$A = \begin{bmatrix} f_1(t_1) & f_2(t_1) & \cdots & f_n(t_1) \\ f_1(t_2) & f_2(t_2) & \cdots & f_n(t_2) \\ \vdots & \vdots & & \vdots \\ f_1(t_m) & f_2(t_m) & \cdots & f_n(t_m) \end{bmatrix},$$

$$\mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad \mathrm{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- $\boldsymbol{y}$ is the vector of measurements.
- $\boldsymbol{r}$ is the vector of residuals.
- $\boldsymbol{x}$ is the vector of coefficients.
- A is constructed such that the jth column is the jth basis function sampled at the abscissas $t_1, t_2, \cdots, t_m$.

Matrix-Vector Notation

Define the matrix $A \in \mathbb{R}^{m \times n}$, the vectors $\mathrm{y}, \mathrm{r} \in \mathbb{R}^m$ and $\boldsymbol{x} \in \mathbb{R}^n$ as follows,

$$A = \begin{bmatrix} f_1(t_1) & f_2(t_1) & \cdots & f_n(t_1) \\ f_1(t_2) & f_2(t_2) & \cdots & f_n(t_2) \\ \vdots & \vdots & & \vdots \\ f_1(t_m) & f_2(t_m) & \cdots & f_n(t_m) \end{bmatrix},$$

$$\mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad \mathrm{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Then,

$$r_i = y_i - \sum_{j=1}^{n} x_j\, f_j(t_i), \;\; i = 1, \cdots, m. \quad \Longleftrightarrow \mathrm{r} = \mathrm{y} - A\boldsymbol{x}.$$

Matrix-Vector Notation

Then,

$$r_i = y_i - \sum_{j=1}^{n} x_j\, f_j(t_i),\ \ i = 1, \cdots, m. \quad \Longleftrightarrow \mathrm{r} = \mathrm{y} - A\boldsymbol{x}.$$

Define

$$\rho(\boldsymbol{x}) := \sum_{i=1}^{m} r_i(\boldsymbol{x})^2 = \|\boldsymbol{r}\|_2^2 = \|\boldsymbol{y} - A\,\boldsymbol{x}\|_2^2.$$

The **linear LSQ problem** in linear algebra notation is:

$$\boxed{\min_{\boldsymbol{x}}\ \rho(\boldsymbol{x}) = \|\boldsymbol{y} - A\,\boldsymbol{x}\|_2^2.}$$

Characterization of the LSQ Solution

The linear least squares problem:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\,\boldsymbol{x}\|_2^2,$$

where $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{x} \in \mathbb{R}^n$. Assume $m \geq n$.

Theorem. The sufficient and necessary condition that $\boldsymbol{x}^*$ is a minimizer of $\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2$ is that $A^T(\boldsymbol{y} - A\boldsymbol{x}) = 0$.

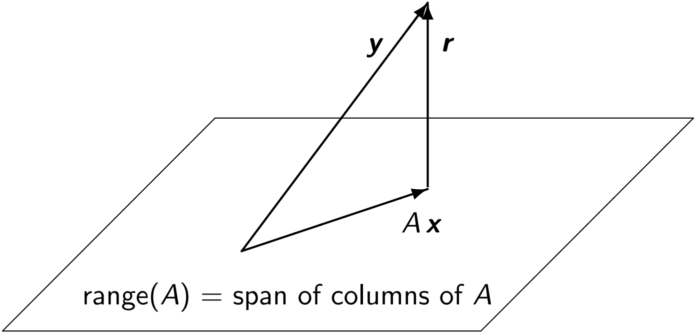
$$\boxed{\text{Normal equations:} \quad A^\top A\,\boldsymbol{x} = A^\top \boldsymbol{y}.}$$

Geometric Interpretation

The smallest residual should satisfy $\boldsymbol{r} \perp \mathrm{range}(A)$, i.e., a minimizer of the linear LSQ problem, $\boldsymbol{x}^*$, should satisfies that $\boldsymbol{r}^*$ is orthogonal to any $A\boldsymbol{x}$:

$$
\begin{aligned}
(A\,\boldsymbol{x})^\top \boldsymbol{r}^* = 0 &\Leftrightarrow \boldsymbol{x}^\top A^\top(\boldsymbol{y} - A\,\boldsymbol{x}^*) = 0 \\
&\Leftrightarrow \boldsymbol{x}^\top (A^\top \boldsymbol{y} - A^\top A\,\boldsymbol{x}^*) = 0
\end{aligned}
$$

Normal equations: $A^\top A\,\boldsymbol{x} = A^\top \boldsymbol{y}.$



NMR Example

We return to the NMR data fitting problem. For this problem there are $m = 50$ measured data points and the fit function is

$$M(\boldsymbol{x}, t) = x_1 e^{-\lambda_1 t} + x_2 e^{-\lambda_2 t} + x_3, \qquad \lambda_1, \lambda_2 \text{ are known.}$$

The model basis functions are:

$$f_1(t) = e^{-\lambda_1 t}, \qquad f_2(t) = e^{-\lambda_2 t}, \qquad f_3(t) = 1,$$

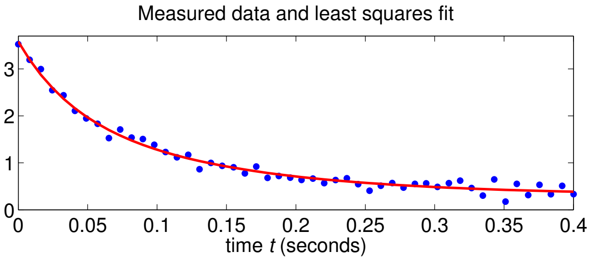
with $n = 3$.

Continue...

The LSQ solution to the least squares problem is $\boldsymbol{x}^* = (x_1^*, x_2^*, x_3^*)^\top$ with elements

$$x_1^* = 1.303, \qquad x_2^* = 1.973, \qquad x_3^* = 0.305.$$

The exact parameters used to generate the data are 1.27, 2.04, 0.3 .



Normal Equations

$$\boxed{\text{Normal equations:} \quad A^\top A\,\boldsymbol{x} = A^\top \boldsymbol{y}.}$$

$A^\top A$ is symmetric positive semidefinite. If $A^\top A$ is invertible (i.e., A has full column rank, that is, basis functions are linear independent), then we have a unique solution

$$\boldsymbol{x}^* = (A^\top A)^{-1} A^\top \boldsymbol{y}.$$

Computational aspect – sensitivity to rounding errors controlled by

$$\mathrm{cond}(A^T A) = \mathrm{cond}(A)^2.$$

Remark: the matrix $A^\dagger = (A^\top A)^{-1} A^\top$ in the above expression for $\boldsymbol{x}^*$ is called the *pseudoinverse* of A.

Avoiding the Normal Equations

QR factorization of $A \in \mathbb{R}^{m\times n}$ with $m > n$:

$$A = Q\,R \qquad \text{with} \qquad Q \in \mathbb{R}^{m\times m}, \quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \in \mathbb{R}^{m\times n},$$

where $R_1 \in \mathbb{R}^{n\times n}$ is upper triangular and Q is orthogonal, i.e.,

$$Q^T Q = I_m, \qquad Q\,Q^T = I_m, \qquad \|Q\,\boldsymbol{v}\|_2 = \|\boldsymbol{v}\|_2 \text{ for any } \boldsymbol{v}.$$

Splitting: $Q = [\;Q_1,\;\; Q_2\;] \quad \Rightarrow \quad A = Q_1 R_1$, where $Q_1 \in \mathbb{R}^{m\times n}$.

- full QR factorization, $\mathtt{Q} = Q$ and $\mathtt{R} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$:
  - (Python) `Q, R = numpy.linalg.qr(A)`
  - (Matlab) `[Q,R] = qr(A);`
- economy-size version with $\mathtt{Q} = Q_1$ & $\mathtt{R} = R_1$:
  - (Python) `Q, R = numpy.linalg.qr(A, mode='reduced')`
  - (Matlab) `[Q,R] = qr(A,0);`

Least Squares Solution

$$\begin{aligned}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 &= \|Q^\top(\boldsymbol{y} - QR\boldsymbol{x})\|_2^2 \\ &= \left\| \begin{bmatrix} Q_1^\top \\ Q_2^\top \end{bmatrix} \boldsymbol{y} - \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \boldsymbol{x} \right\|_2^2 \\ &= \left\| Q_1^\top \boldsymbol{y} - R_1 \boldsymbol{x} \right\|_2^2 + \left\| Q_2^\top \boldsymbol{y} \right\|_2^2 .\end{aligned}$$

Last term is independent on $\boldsymbol{x}$, so

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 \qquad \Longleftrightarrow \qquad \min_{\boldsymbol{x}} \left\| Q_1^\top \boldsymbol{y} - R_1 \boldsymbol{x} \right\|_2^2 .$$

If A have full rank then so has R_1 and therefore

$$\boldsymbol{x}^* = R_1^{-1} Q_1^\top \boldsymbol{y} \qquad \text{and} \qquad \boldsymbol{r}^* = Q_2 Q_2^\top \boldsymbol{y}.$$

I = Q(Q^T) = Q1(Q2)^T + Q2(Q2)^T

Note that the upper triangular matrix R_1 is also the Cholesky factor of the normal equation matrix, i.e., $A^T A = R_1^T R_1$.

NMR Example

A and Q_1:

$$\begin{bmatrix} 1.00 & 1.00 & 1 \\ 0.80 & 0.94 & 1 \\ 0.64 & 0.88 & 1 \\ \vdots & \vdots & \vdots \\ 3.2 \cdot 10^{-5} & 4.6 \cdot 10^{-2} & 1 \\ 2.5 \cdot 10^{-5} & 4.4 \cdot 10^{-2} & 1 \\ 2.0 \cdot 10^{-5} & 4.1 \cdot 10^{-2} & 1 \end{bmatrix}, \quad \begin{bmatrix} 0.597 & -0.281 & 0.172 \\ 0.479 & -0.139 & 0.071 \\ 0.384 & -0.029 & -0.002 \\ \vdots & \vdots & \vdots \\ 1.89 \cdot 10^{-5} & 0.030 & 0.224 \\ 1.52 \cdot 10^{-5} & 0.028 & 0.226 \\ 1.22 \cdot 10^{-5} & 0.026 & 0.229 \end{bmatrix},$$

$$R_1 = \begin{bmatrix} 1.67 & 2.40 & 3.02 \\ 0 & 1.54 & 5.16 \\ 0 & 0 & 3.78 \end{bmatrix}, \qquad Q_1^T \boldsymbol{y} = \begin{bmatrix} 7.81 \\ 4.32 \\ 1.19 \end{bmatrix}.$$

$$\boldsymbol{x}^* = R_1^{-1} Q_1^\top \boldsymbol{y} = \begin{bmatrix} 1.303 \\ 1.973 \\ 0.305 \end{bmatrix}.$$

Normal Equations vs QR Factorization

Normal equations

- Squared condition number: $\mathrm{cond}(A^T A) = \mathrm{cond}(A)^2$.
- OK for well conditioned A.
- Work $= mn^2 + (1/6)n^3$ flops.

QR factorization

- No squaring of condition number, $\mathrm{cond}(R) = \mathrm{cond}(A)$
- Can better handle ill conditioned A.
- Work $= 2mn^2 - (2/3)n^3$ flops.

Solving normal equations directly always faster, but risky for ill-conditioned system matrices.

Solving with QR factorization is always stable, so it is a better black-box method.

Singular Value Decomposition (SVD)

$$A = U\Sigma V^\top, \qquad \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{bmatrix}.$$

- $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices. The columns of $U = [\mathrm{u}_1, \mathrm{u}_2, \cdots, \mathrm{u}_m]$ and $V = [\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_n]$ are the left and right singular vectors, and form an orthonormal basis for $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.
- $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix. The diagonal elements σ_i of Σ are the singular values of A, and they appear in descending order.
- $rank(A) = r$ iff $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and $\sigma_{r+1} = \cdots = \sigma_n = 0$.

Singular Value Decomposition (SVD)

$$A = [\mathrm{u}_1, \cdots, \mathrm{u}_r] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} \mathrm{v}_1^\top \\ \vdots \\ \mathrm{v}_r^\top \end{bmatrix} = \sum_{i=1}^{r} \sigma_i \mathrm{u}_i \mathrm{v}_i^\top$$

- $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices. The columns of $U = [\mathrm{u}_1, \mathrm{u}_2, \cdots, \mathrm{u}_m]$ and $V = [\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_n]$ are the left and right singular vectors, and form an orthonormal basis for $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.
- $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix. The diagonal elements σ_i of Σ are the singular values of A, and they appear in descending order.
- $rank(A) = r$ iff $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and $\sigma_{r+1} = \cdots = \sigma_n = 0$.

Some Properties

- $A\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i$ and $A^\top \boldsymbol{u}_i = \sigma_i \boldsymbol{v}_i$ for $i = 1, \cdots, n$, while $A^\top \boldsymbol{u}_i = 0$ for $i = n+1, \cdots, m$.
- if $r < n$, then $A\boldsymbol{v}_i = 0$ and $A^\top \boldsymbol{u}_i = 0$ for $i = r+1, \cdots, n$.
- $(A^\top A)\boldsymbol{v}_i = \sigma_i^2 \boldsymbol{v}_i$ and $(AA^\top)\boldsymbol{u}_i = \sigma_i^2 \boldsymbol{u}_i$ for $i = 1, \cdots, r$.
- $\|A\|_2 = \sigma_1$.
- $\|A\|_F = (\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2)^{1/2}$.

Least Squares Solution

- **Full rank case (i.e., $r = n$):** The solution of $\min_{\boldsymbol{x}} \|\boldsymbol{y} - A\boldsymbol{x}\|_2^2$ is

$$\boldsymbol{x}^* = (A^\top A)^{-1} A^\top \boldsymbol{y} = V\Sigma^\dagger U^\top \boldsymbol{y} = \sum_{i=1}^{n} \frac{\boldsymbol{u}_i^\top \boldsymbol{y}}{\sigma_i} \boldsymbol{v}_i,$$

where $\Sigma^\dagger = \begin{bmatrix} \sigma_1^{-1} & & & \\ & \ddots & & 0 \\ & & \sigma_n^{-1} & \end{bmatrix}$ is the pseudoinverse of Σ.

Further, $A^\dagger = V\Sigma^\dagger U^\top$.

Least Squares Solution

- **Rank deficient case (i.e., $r < n$):**

$$\begin{aligned}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 &= \|U^\top \boldsymbol{y} - \Sigma V^\top \boldsymbol{x}\|_2^2 \\ &= \left\| \begin{bmatrix} U_r^\top \\ U_{m-r+1}^\top \end{bmatrix} \boldsymbol{y} - \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^\top \\ V_{n-r+1}^\top \end{bmatrix} \boldsymbol{x} \right\|_2^2 \\ &= \left\| U_r^\top \boldsymbol{y} - \Sigma_r V_r^\top \boldsymbol{x} \right\|_2^2 + \left\| U_{m-r+1}^\top \boldsymbol{y} \right\|_2^2 .\end{aligned}$$

Hence, to minimize $\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2$ we have

$$\boldsymbol{x}^* = V_r \Sigma_r^{-1} U_r^\top \boldsymbol{y} = \sum_{i=1}^{r} \frac{\boldsymbol{u}_i^\top \boldsymbol{y}}{\sigma_i} \boldsymbol{v}_i \quad \text{and} \quad \|\boldsymbol{r}^*\|_2^2 = \sum_{i=r+1}^{m} (\boldsymbol{u}_i^\top \boldsymbol{y})^2,$$

where $\boldsymbol{x}^*$ is the solution of minimal l_2-norm, i.e.,

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2, \quad \text{subject to } \|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 = \min$$

Least Square Solution

- **Ill-conditioned case (i.e., $r = n$ but $\mathrm{cond}(A)$ very large):**. The solution of LSQ problem is

$$\boldsymbol{x}^* = \sum_{i=1}^{n} \frac{\boldsymbol{u}_i^\top \boldsymbol{y}}{\sigma_i} \boldsymbol{v}_i,$$

which can be dominated by the terms with $\sigma_n, \sigma_{n-1}, \cdots$, because of their small values. It turns out that the solution is very sensitive to perturbations. In which case, the solution of LSQ problem becomes

$$\boldsymbol{x} = \sum_{i=1}^{n} \frac{\boldsymbol{u}_i^\top (\boldsymbol{y} + \epsilon)}{\sigma_i} \boldsymbol{v}_i = \boldsymbol{x}^{\mathrm{exact}} + \sum_{i=1}^{n} \frac{\boldsymbol{u}_i^\top \epsilon}{\sigma_i} \boldsymbol{v}_i,$$

which can be dominated by the terms with very small σ_i.

Least Square Solution

- **Ill-conditioned case (i.e., $r = n$ but $\mathrm{cond}(A)$ very large):**. An intuitive approximate solution with less sensitivity is to omit the terms corresponding to the small σ_i, i.e.,

$$\boldsymbol{x}^* = \sum_{i=1}^{k} \frac{\boldsymbol{u}_i^\top \boldsymbol{y}}{\sigma_i} \boldsymbol{v}_i,$$

where $\sigma_1 \geq \cdots \geq \sigma_k \geq \tau > \sigma_{k+1} \geq \cdots \geq \sigma_n$.

In fact, x^* is the minimal l_2-norm solution of the LSQ problem:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - A_k \boldsymbol{x}\|_2^2,$$

where A_k is the rank-k approximation of A.

NMR Example

$$\Sigma_3 = \begin{bmatrix} 7.46 & 0 & 0 \\ 0 & 2.23 & 0 \\ 0 & 0 & 0.59 \end{bmatrix},$$

$$V_3 = \begin{bmatrix} 0.114 & 0.613 & 0.782 \\ 0.312 & 0.725 & -0.614 \\ 0.943 & -0.314 & 0.109 \end{bmatrix}, \qquad U_3^\top \boldsymbol{y} = \begin{bmatrix} 7.846 \\ 4.758 \\ -0.095 \end{bmatrix}$$

Then,

$$\boldsymbol{x}^* = \sum_{i=1}^{3} \frac{\boldsymbol{u}_i^\top \boldsymbol{y}}{\sigma_i} \boldsymbol{v}_i = \begin{bmatrix} 1.303 \\ 1.973 \\ 0.305 \end{bmatrix}.$$

Introduction of Weights

The statistical assumptions about the errors:

$$\mathcal{E}(e_i) = 0, \qquad \mathcal{E}(e_i^2) = \varsigma^2, \qquad i = 1, 2, \ldots, m,$$

The maximum likelihood principle in statistics tells us that we should solve the following optimization problem:

$$\min_{\boldsymbol{x}} \frac{1}{2\varsigma^2} \sum_{i=1}^{m} (y_i - \phi(\boldsymbol{x}; t_i))^2 = \min_{\boldsymbol{x}} \sum_{i=1}^{m} \left(\frac{r_i(\boldsymbol{x})}{\varsigma} \right)^2$$

Introduction of Weights

The statistical assumptions about the errors:

$$\mathcal{E}(e_i) = 0, \qquad \mathcal{E}(e_i^2) = {\varsigma_i}^2, \qquad i = 1, 2, \ldots, m,$$

The maximum likelihood principle in statistics tells us that we should solve the following optimization problem:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{m} \left(\frac{y_i - \phi(\boldsymbol{x}; t_i)}{\varsigma_i} \right)^2 = \min_{\boldsymbol{x}} \sum_{i=1}^{m} \left(\frac{r_i(\boldsymbol{x})}{\varsigma_i} \right)^2$$

Intuitive result:
we can allow the expected value of the approximation errors to be larger for those data (t_i, y_i) that have larger errors.

Weights in Matrix-Vector Formulation

We introduce the diagonal matrix

$$W = \mathrm{diag}(w_1, \dots, w_m), \qquad w_i = \varsigma_i^{-1}, \quad i = 1, 2, \dots, m.$$

Then, we define the **linear weighted LSQ problem** as

$$\boxed{\min_{\boldsymbol{x}} \rho_W(\boldsymbol{x})}$$

where

$$\rho_W(\boldsymbol{x}) = \sum_{i=1}^{m} \left(\frac{r_i(\boldsymbol{x})}{\varsigma_i} \right)^2 = \| W\,(\boldsymbol{y} - A\,\boldsymbol{x}) \|_2^2.$$

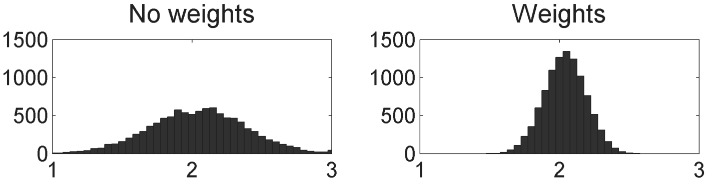
Same computational problem as before, with $\boldsymbol{y} \to W\,\boldsymbol{y}$, $A \to WA$.

Example with Weights

The NMR data again, but we add larger Gaussian noise to the first 10 data points, with standard deviation 0.5:

- $\varsigma_i = 0.5,\ i = 1, 2, \ldots, 10$ (first 10 data with larger errors)
- $\varsigma_i = 0.1,\ i = 11, 12, \ldots, 50$ (remaining data with smaller errors).

The weights: $w_i = \varsigma_i^{-1}$ are $2, 2, \ldots, 2, 10, 10, \ldots, 10$.
We solve the problem with and without weights for $10,000$ instances of the noise, and consider x_2 (the exact value is 2.04).



Residual Analysis

$$r_i = \underline{e_i} + \big(\underline{\Gamma(t_i) - \phi(\boldsymbol{x}; t_i)}\big) \qquad i = 1, \cdots, m.$$

1. The model captures the pure-data function "well enough" when the approximation errors are smaller than the data errors. Then the residuals are dominated by the data errors and some of the statistical properties of the errors carry over to the residuals.

2. If the fit model does not capture the behavior of the pure-data function, then the residuals are dominated by the approximation errors. Then the residuals will tend to behave as a sampled signal and show strong local correlations.

Assumptions and Techniques

We make the following assumptions about the data errors e_i:

- They are random variables with mean zero and identical variance ς^2, i.e., $\mathcal{E}(e_i) = 0$ and $\mathcal{E}(e_i^2) = \varsigma^2$ for $i = 1, 2, \ldots, m$.
- They follow normal distribution, $e_i \sim \mathcal{N}(0, \varsigma^2)$.

We describe two tests with two different properties.

- Randomness test: check for randomness of the signs of r_i.
- Autocorrelation test: check if the residuals are uncorrelated.

Test for Random Signs

Can we consider the signs of the residuals to be random? We can utilize **Run test** from time series analysis.

Given a sequence of two symbols — in our case, "$+$" and "$-$" for positive and negative residuals r_i — a *run* is defined as a succession of identical symbols surrounded by different symbols.

The sequence "$+++----++-----+++$" has:

- $m = 17$ elements,
- $n_+ = 8$ pluses,
- $n_- = 9$ minuses, and
- $u = 5$ runs: $+++$, $----$, $++$, $-----$, and $+++$.

The Run Test

The number of runs u (not the residuals!) can be approximated by a normal distribution with mean μ_u and standard deviation ς_u given by

$$\mu_u = \frac{2\, n_+\, n_-}{m} + 1, \qquad \varsigma_u^2 = \frac{(\mu_u - 1)\,(\mu_u - 2)}{m-1}.$$

With a 5 % significance level we will accept the sign sequence as random if

$$z_\pm = \frac{|u - \mu_u|}{\varsigma_u} < 1.96\ .$$

In the above example with 5 runs we have $z_\pm = 2.25$ and the sequence of signs cannot be considered random.

Estimation of Noise Standard Deviation

Recall that $\boldsymbol{y} = \Gamma + \mathrm{e}$. Covariance matrix for $\boldsymbol{y}$:

$$\mathrm{Cov}(\boldsymbol{y}) = \mathcal{E}\big((\boldsymbol{y} - \mathcal{E}(\boldsymbol{y}))\,(\boldsymbol{y} - \mathcal{E}(\boldsymbol{y}))^\top\big) = \mathcal{E}(\mathrm{e}\,\mathrm{e}^\top).$$

White noise in the data: $\mathrm{Cov}(\boldsymbol{y}) = \varsigma^2\, I_m$.
According to $\boldsymbol{r}^* = Q_2\, Q_2^\top \boldsymbol{y}$, we can derive

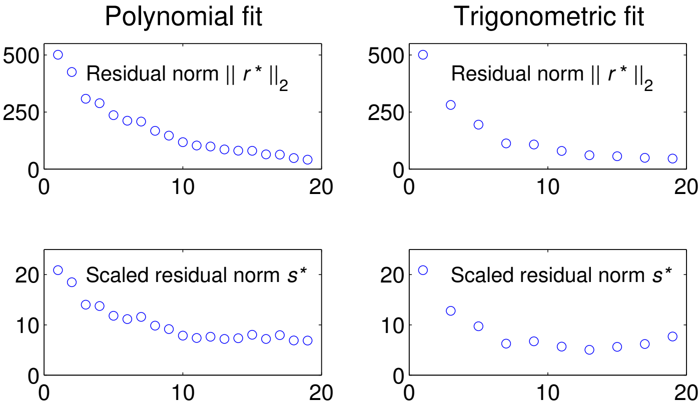
$$\begin{aligned} \mathrm{Cov}(\boldsymbol{r}^*) &= \varsigma^2 Q_2\, Q_2^\top \\ \mathcal{E}(\|\boldsymbol{r}^*\|_2^2) &= \|Q_2^\top \Gamma\|_2^2 + \mathcal{E}(\|Q_2^\top \mathrm{e}\|_2^2) \\ &= \|Q_2^\top \Gamma\|_2^2 + (m-n)\,\varsigma^2. \end{aligned}$$

Hence, if the approximation errors are smaller than the data errors then $\|\boldsymbol{r}^*\|_2^2 \approx (m-n)\,\varsigma^2$ and the scaled residual norm

$$s^* = \frac{\|\boldsymbol{r}^*\|_2}{\sqrt{m-n}}$$

is an estimate of the standard deviation ς of the errors in the data. We can monitor s^* as a function of n to estimate ς.

$\|r^*\|_2$ and s^*



Example on air pollution: The residual norm is monotonically decreasing while s^* has a plateau/minimum.