The method of least squares

$\vec{x}_i$ = independent variables (n vectors of any dimension, $i = 1..n$)
y_i = dependent variables (real numbers)
$1/\sigma_i^2$ = weights
$\vec{a}$ = parameters (p real parameters written as a vector), $p \le n$, preferably $p \ll n$

We are looking for function $f_{\vec{a}}(\vec{x})$ (called "model") dependent on p parameters $\vec{a}$ which describes data $(\vec{x}_i, y_i)$. The parameters $\vec{a}$ are to be determined so that the sum of squared deviations is minimized:

$$\min_{\vec{a}} S^2, \quad S^2 = \sum_{i=1}^{n} \left[\frac{f_{\vec{a}}(\vec{x}_i) - y_i}{\sigma_i}\right]^2$$

Theorem (Gauss–Markov): for function $f_{\vec{a}}$ linearly dependent on $\vec{a}$, the above solution is the:

Best (gives the smallest variance of the estimated $\vec{a}$)
Linear (the assumption)
Unbiased ($\langle\vec{a}\rangle$ is correct)
Estimate (BLUE).

$\langle S^2\rangle = n - p$

- In the limit $n \to \infty$ it holds $s = \sqrt{S^2/(n-p)} \to 1$ (assessment of the fit)

The method of least squares

Example. For $f_a(\vec{x}) = a$ (a constant) and $\sigma_i = 1$ find the estimate of a

$\bar{y} = a$

The results of **fitting** (correlation, regression) include:

- the estimate of $\vec{a}$
- the estimates of standard errors of $\vec{a}$
- the correlation between parameters (covariances)
- often, the estimate of a function $g(\vec{a})$ (incl. its error estimate)

Linearization

Let $\vec{a}_0$ be the exact (looked for) value of parameters. For each $\vec{x}$:

$$f_{\vec{a}}(\vec{x}) \approx f_{\vec{a}_0} + \sum_{j=1}^{p} \Delta a_j f_j(\vec{x}), \quad f_j(\vec{x}) = \frac{\partial f_{\vec{a}_0}(\vec{x})}{\partial a_j}$$

where $\vec{a} = \vec{a}_0 + \Delta\vec{a}$.

If the changes in parameters $\vec{a}$ are small, it is enough (without loss of generality) to study the linear model, and for notation simplicity set $f_{\vec{a}_0} = 0$ and $\vec{a}_0 = 0$

$$f_{\vec{a}}(\vec{x}) = \sum_{j=1}^{p} a_j f_j(\vec{x}),$$

where $\{f_j(\vec{x})\}_{j=1}^{p}$ is a basis (not necessarily orthogonal)

Linear model

$$f_{\vec{a}}(\vec{x}) = \sum_{j=1}^{p} a_j f_j(\vec{x})$$

Let us assume that data y_i are independent random variables, but generally with different standard deviations σ_i; we will write this as the correct value + random variable δy_i:

Kronecker delta: $\delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \ne j \end{cases}$

$$y_i = \sum_{j=1}^{p} a_j f_j(\vec{x}) + \delta y_i, \quad \langle\delta y_i\rangle = 0, \ \langle\delta y_i \delta y_j\rangle = \sigma_i^2 \delta_{ij}$$

We shall minimize the following object function:

$$S^2 = \sum_{i=1}^{n} \left[\frac{\sum_{j=1}^{p} a_j f_j(\vec{x}_i) - y_i}{\sigma_i}\right]^2$$

Necessary condition for the minimum:

$$\frac{1}{2}\frac{\partial S^2}{\partial a_k} = \sum_{i=1}^{n} \frac{f_k(\vec{x}_i)}{\sigma_i} \left[\frac{\sum_{j=1}^{p} a_j f_j(\vec{x}_i) - y_i}{\sigma_i}\right] = (A \cdot \vec{a} - \vec{b})_k \stackrel{!}{=} 0$$

Let $F_{ki} = f_k(x_i)/\sigma_i$ (matrix $p \times n$), $Y_i = y_i/\sigma_i$, then $A = F \cdot F^{\mathrm{T}}$, $\vec{b} = F \cdot \vec{Y}$ $\quad\langle\delta Y_i \delta Y_j\rangle = \delta_{ij}$

The covariance matrix

$$A \cdot \vec{a} = \vec{b}, \quad \vec{a} = A^{-1} \cdot \vec{b} = A^{-1} \cdot F \cdot \vec{Y}$$

Errors of estimates and the correlations of parameters:

$\sum$ is over pairs of the same indices

$$\begin{aligned}
\mathrm{Cov}(a_i, a_j) = \langle\Delta a_i \Delta a_j\rangle &= \sum A^{-1}_{i\alpha} F_{\alpha k} \delta Y_k A^{-1}_{j\beta} F_{\beta l} \delta Y_l \\
&= \sum A^{-1}_{i\alpha} F_{\alpha k} A^{-1}_{j\beta} F_{\beta l} \delta_{kl} \\
&= \sum A^{-1}_{i\alpha} F_{\alpha k} A^{-1}_{j\beta} F_{\beta k} \\
&= \sum A^{-1}_{i\alpha} A_{\alpha\beta} A^{-1}_{j\beta} \\
&= \sum A^{-1}_{i\alpha} A_{\alpha\beta} A^{-1}_{\beta j} \\
&= A^{-1}_{ij}
\end{aligned}$$

The above matrix is called "covariance" or "variance-covariance" matrix (there are variances in the diagonal)

The result of fitting includes not only the error estimates (on the diagonal), but also their correlations (covariances)!

Linear fitting: Notes

- Remember: if all σ_i are accurate estimates of standard deviations and there are enough data points, n, then

  $n - p$ is called the "number of degrees of freedom" and often denoted as ν

  $$s = \sqrt{\frac{S^2}{n-p}}$$

  should be close to unity.
- Often σ_i's are not known but it may be assumed that all are the same. Then, equation $s = 1$ may be used to back calculate σ:

  $$\sigma = \sqrt{\frac{S^2}{n-p}}$$

  Put another way (most software incl. Maple works like this): If we define $F_{ki} = f_k(x_i)$, $A = F \cdot F^{\mathrm{T}}$, $\vec{b} = F \cdot \vec{y}$, then (with the above σ and enough n) it holds:

  $$\mathrm{Cov}(a_i, a_j) = A^{-1}_{ij} \sigma^2$$
- If functions f_j are perpendicular, then A is diagonal and the parameters are not correlated. This is difficult to fulfill in practice for a nonlinear (but locally linearized) estimate.

Error of a function of parameters

We have to calculate $g(\vec{a})$ (incl. the error)

$$g_{\vec{a}} \approx g_{\vec{a}_0} + \sum_{j=1}^{p} a_j g_j(\vec{x}), \quad g_j = \frac{\partial g_{\vec{a}_0}}{\partial a_j} \tag{1}$$

$$\langle (g_{\vec{a}} - g_{\vec{a}_0})^2 \rangle = \langle \sum_{ij} a_i g_i a_j g_j \rangle = \sum_{ij} g_i \mathrm{Cov}(a_i, a_j) g_j$$

Examples of $g(\vec{a})$: a_i (one of the parameters), $\int_{x_0}^{x_1} f(x)\mathrm{d}x$

Errors by MC sampling

- Minimize $S^2 \Rightarrow$ we get $\vec{a}_0$ and $g(\vec{a}_0)$
- For $k = 1..m$:
  - Fabricate data:

    $$y_i^{(k)} = f_{\vec{a}_0}(\vec{x}_i) + \sigma_i u$$

    where u is a random number with normalized Gauss' distribution
    (we know errors σ_i of data y_i; if not, $\sigma_i = [S^2/(n-p)]^{1/2}$ can be used)
  - Calculate parameters $\vec{a}^{(k)}$ by the least squares
  - Calculate $g(a^{(k)})$
- Treat the results $g(a^{(k)})$ for $k = 1..m$ as independent data $\to$ estimate of the standard error $\sigma(g)$

Numerical notes

Linear model: Solvable by the linear algebra methods, usually easy. In case of problems, orthonormalization of a basis helps.

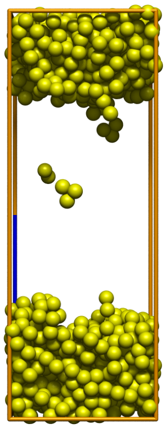
Nonlinear model:
Problem 1: several local minima, some of them $\to \infty$
Problem 2: long curved valleys – slow minimization

Minimization of nonlinear functions of many variables:

- grid search (at start)
- Monte Carlo search (at start)
- steepest descent (greedy)
- conjugated gradients
- amoeba (Nelder–Mead)
- (Gauss–)Newton method (close to the solution)
- (Levenberg–)Marquardt method (Newton + gradient, damping)
- simulated annealing

Real example

A simulation of a model of Pt in the slab geometry gave the following data for pressure in the direction perpendicular to the slab:



T/K	p/bar	stderr/bar
3700	14.7	2.2
3750	11.9	1.4
3800	14.9	2.6
3850	18.9	2.8
3900	16.3	1.8
3950	16.5	3.2
4000	26.5	3.3
4050	24.3	2.6
4100	30.6	2.6
4150	28.5	3.5
4200	34.5	3.5
4250	43.4	2.6
4300	48.0	3.1

Calculate the boiling point of Pt at 1 bar and estimate the error.

Solution

We will assume the Clausius–Clapeyron equation and constant vaporization enthalpy:

$$\ln p = a + b/T$$

where a and b are constants to fit. Then, function g is the solution of equation $\ln p = a + b/T$ for $p = 1$ bar.

- Direct fitting to $p = \exp(a + b/T)$:
  $s = 1.067$, $T_{\rm vap} = 3021(55)$ K, rescaled by s (59)
- Fitting $\ln(p)$ vs. $1/T$ (linear regression):
  $s = 1.081$, $T_{\rm vap} = 2992(53)$ K, rescaled by s (57)
- Without knowledge of standard errors of the data:
  $T_{\rm vap} = 3015(74)$ K
- Since the data are based on trajectories of the same length, the errors may be smoothed. Then:
  $s = 1.138$, $T_{\rm vap} = 2965(63)$ K, rescaled by s (72)

Summary: $T_{\rm vap} = 2965(72)$

Another example

Fit the data to a suitable function $f(x)$ and provide the solution x_0 of equation $f(x_0) = 1$, including the standard error estimate.

12.49(5)

x	y	σ
2	4.001	0.014
3	3.424	0.013
4	3.039	0.011
5	2.710	0.010
6	2.482	0.009
7	2.208	0.008
8	1.985	0.008
9	1.749	0.007
10	1.528	0.007

Notes

- Maple calculates the standard errors of parameters (option 'standarderrors' in Maple) and the covariance matrix ('variancecovariancematrix') from the (weighted) sum of squares even if weights $w_i = 1/\sigma_i^2$ are given. If your σ_is are reliable, you should divide the 'standarderrors' by the 'residualstandarddeviation', and the 'variancecovariancematrix' by its square.
- 'residualstandarddeviation' should be close to 1 (with precision perimitted by the number of data points).
- The sensitivities g_i, eq. (1), of the root of equation $f(x) = y$ on parameters can be obtained from the formula for the derivative of implicit function:

$$f(a_i + \mathrm{d}a_i, x + \mathrm{d}x) = y$$

$$\frac{\partial f}{\partial a_i}\mathrm{d}a_i + \frac{\partial f}{\partial x}\mathrm{d}x = 0$$

$$g_i \equiv \frac{\partial x}{\partial a_i} = -\frac{\partial f}{\partial a_i} \Big/ \frac{\partial f}{\partial x}$$

Fitting in Excel and LibreOffice: Intro

Excel and LibreOffice provide a general routine for linear regression LINEST.

Function LINEST fits data $\vec{y}$ (n-vector) to a linear function of p vectors $\vec{x}_j$, $j = 1..p$:

$$\vec{y} = a_0 + \sum_{j=1}^{p} b_j \vec{x}_j \tag{2}$$

The absolute term a_0 is optional, cf. the 3rd argument to LINEST.
Function LINEST returns the values of parameters a_j including the standard errors and $S/\sqrt{n-p}$ for simple linear regression without weights.*

Example. For fitting to $a_1 + a_2 x + a_3 \ln x$, the basis vectors $\vec{x}_j$ are:

$$\begin{aligned} \vec{x}_1 &= (\vec{x})^0 = (1, 1, .., 1)^{\rm T} \quad \text{or use version with } a_0+ \\ \vec{x}_2 &= \vec{x} = (x_1, x_2, .., x_n)^{\rm T} \\ \vec{x}_3 &= \ln(\vec{x}) = (\ln x_1, \ln x_2, .., \ln x_n)^{\rm T} \end{aligned}$$

where $\vec{x}$ is the original vector of independent x's and the functions are interpreted by elements.

*AFAIK, the covariance matrix is not provided; after some effort it can be evaluated using formulas on pages 3–4 and Excel/LibreOffice matrix functions as MMULT, MINVERSE.

Fitting in Excel and LibreOffice: Syntax

Only a minimum subset of syntax is explained

- Prepare column vectors $\vec{y}$ and $\vec{x}$
- Prepare the vectors with bases $\vec{x}_j$, $j = 1..p$
- Mark rectangle (array) of size $(p+1) \times 4$ cells to accommodate the results
- To the first cell of this rectangle, type
  =LINEST(Y1:Yn,X1:X^{p}n,0,1)
  where the arguments are:
  1. $\vec{y} = Y1 : Yn$ (column)
  2. $\vec{x}_1..\vec{x}_p = X1 : X^p n$ is a $p \times n$ matrix (p columns)
  3. 0 means that a_0+ is not considered
  4. 1 means rich output
- Type the "three-finger salute" **Ctrl-Shift-Enter**

Use ; instead of , in Czech localization

The resulting estimates are in the form of $(p+1) \times 4$ array:

$\langle b_p \rangle$	$\langle b_{p-1} \rangle$	...	$\langle b_2 \rangle$	$\langle b_1 \rangle$	?
$\sigma(b_p)$	$\sigma(b_{p-1})$	...	$\sigma(b_2)$	$\sigma(b_1)$	n.a.
r^2	$S/\sqrt{n-p}$	n.a.	n.a.	n.a.	n.a.
F-value	$n-p$	n.a.	n.a.	n.a.	n.a.
?	$S^2 = \sum_i [f(x_i) - y_i]^2$	n.a.	n.a.	n.a.	n.a.

Note the reversed order of the calculated parameters
r = correlation coefficient

Fitting in Excel and LibreOffice: Data with errors

If data $\vec{y}$ are provided with reliable standard errors $\vec{\sigma}$, we first prepare columns containing (cf. page 1):

$$\vec{y}' = \frac{\vec{y}}{\vec{\sigma}}, \ \vec{x}_j' = \frac{\vec{x}_j}{\vec{\sigma}}, \ j = 1..p$$

where division of vectors is defined element-by-element.

The analysis is the same as on the previous page.

Note that the value of $S/\sqrt{n-p}$ should be around 1. If the individual error estimates of data, $\vec{\sigma}$, are more reliable (based on more points) than this analysis, the obtained $\sigma(a_j)$ should be divided by $S/\sqrt{n-p}$.

Example. Fit the following data to function $a + bx + cx^2$:

x	−2.0	−1.8	−1.6	−1.4	−1.2	−1.0	−0.8	−0.6	−0.4	−0.2	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
y	11.876	10.918	9.746	8.761	7.791	7.003	6.408	5.452	5.010	4.325	3.622	3.466	4.087	3.257	3.517	3.546	2.575	2.525	3.807	3.162	4.141
σ	0.70	0.66	0.62	0.58	0.54	0.50	0.46	0.42	0.38	0.34	0.30	0.34	0.38	0.42	0.46	0.50	0.54	0.58	0.62	0.66	0.70

$a = 4.02(13)$, $b = -1.98(11)$, $c = 0.99(9)$