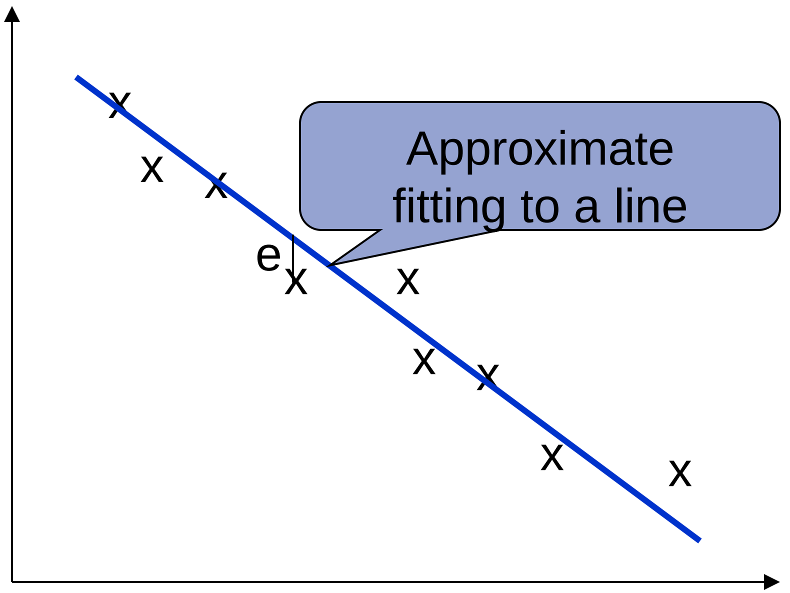
Numerical Analysis – Data Fitting

Hanyang University

Jong-Il Park

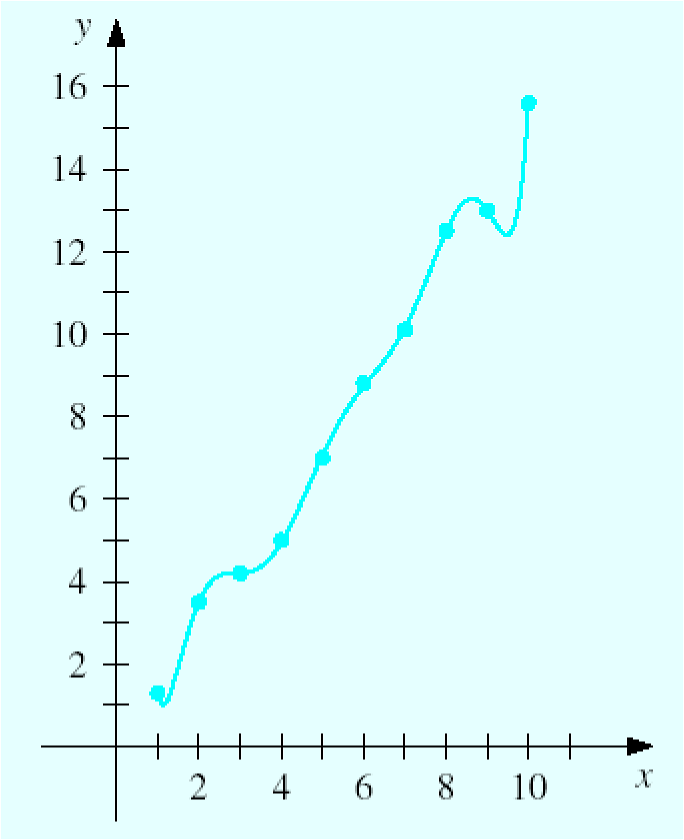
Fitting

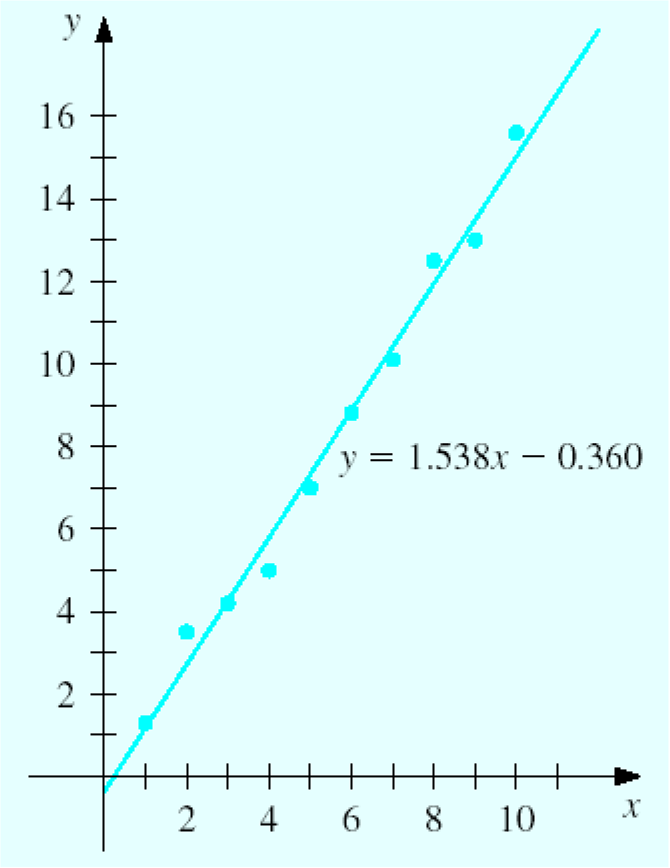
- Exact fit
  - Interpolation
  - Extrapolation



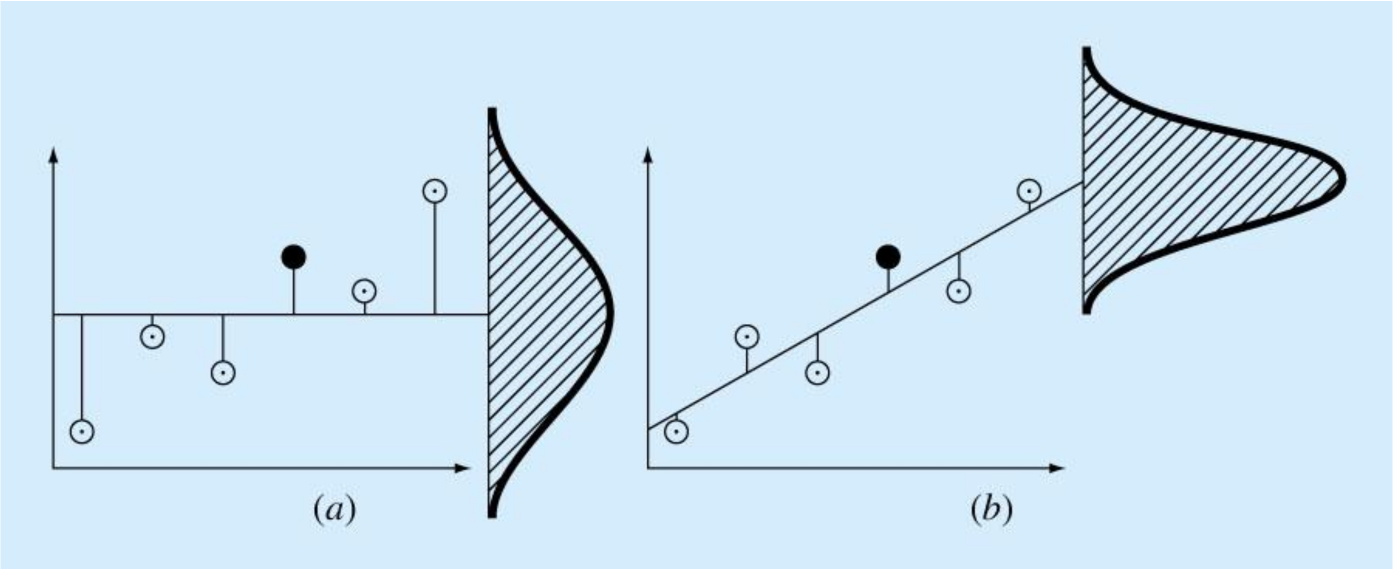
- Approximate fit
  - Allows some errors
  - Optimality depends on noise model

Eg. Interpolation vs. Data Fitting



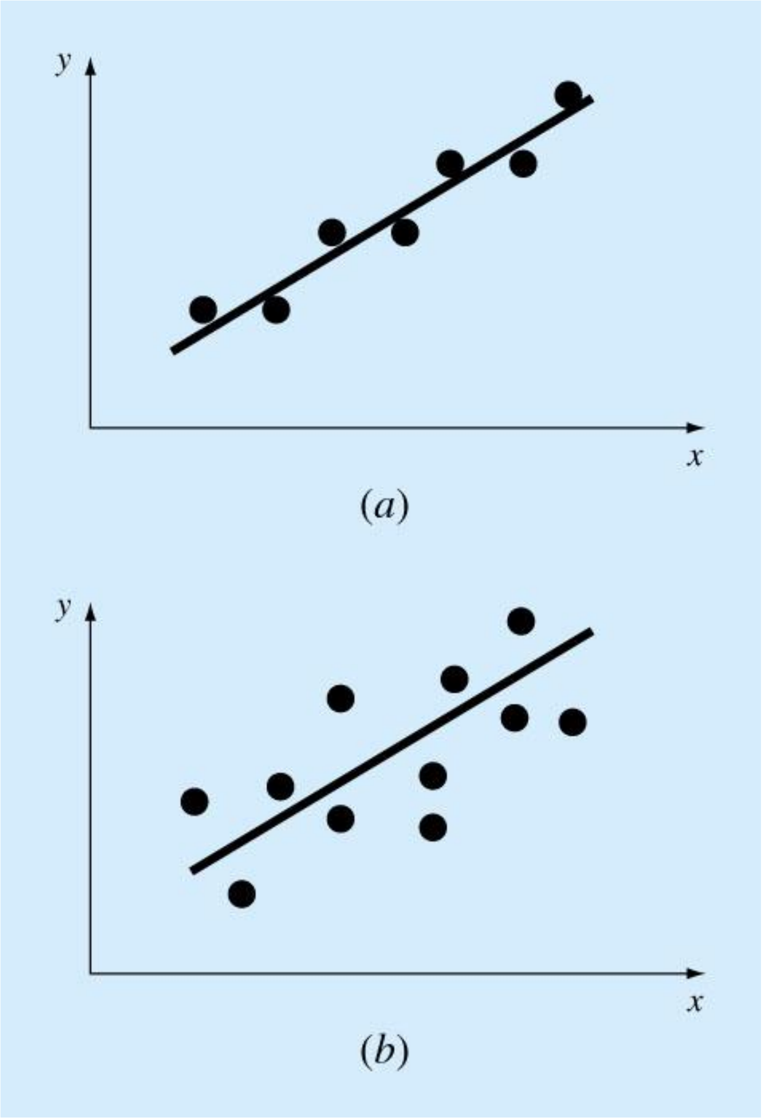


Regression errors(I)



Zero-order model 1st-order model

Regression errors(I)



Small errors

Large errors

Least-Square Data Fitting

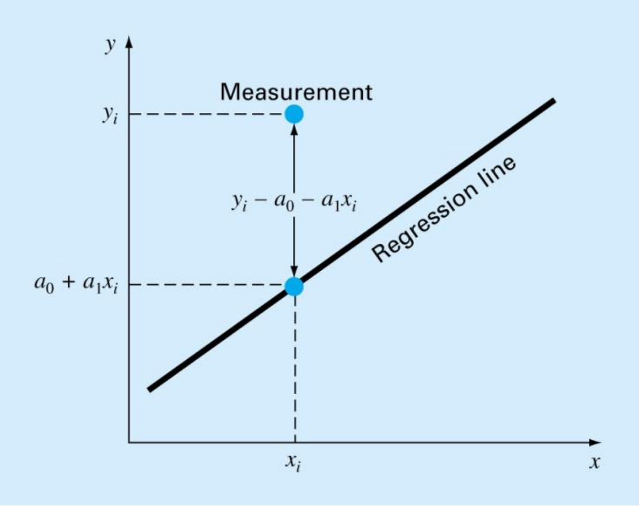
- Problem statement

Given $\begin{cases} \text{N data points } \underline{(x_i, y_i)}, \text{ i=1, ... ,N} \\ \text{a model that has } \underline{\text{M}} \text{ adjustable parameters,} \end{cases}$

$$\underline{a_j}, \quad j = 1, \cdots, M$$

find $\underline{\mathbf{a}} = [a_1, a_2, \cdots, a_M]$ that minimizes

$$S = \sum_{i=1}^{N} [\underbrace{y_i - y(x_i; \mathbf{a})}_{= e_i}]^2$$

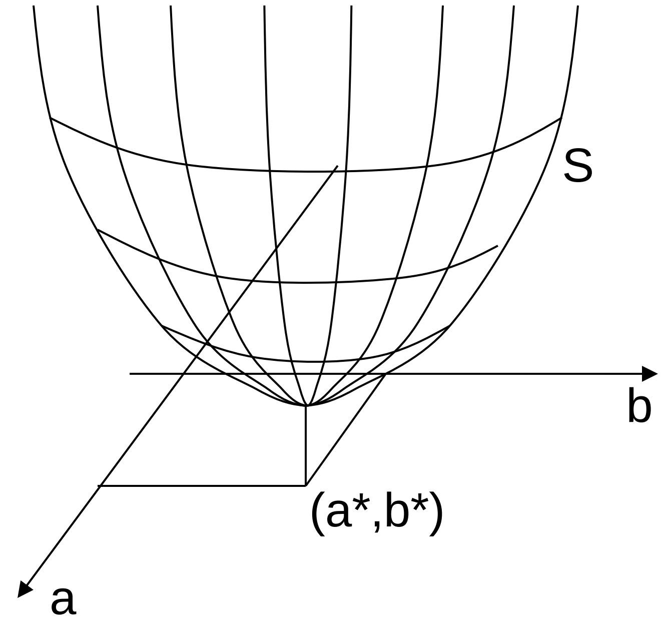


※ Maximum Likelihood Estimation

ML ≡ Least−square

if e_i is independently distributed Gaussian

Fitting data to a straight line



Model $y = a + bx$

Error $e_i = y_i - y(x_i)$

$= y_i - (a + bx_i)$

Sum of Errors $S(a,b) = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} [y_i - (a + bx_i)]^2$

At the minimum error

$$\frac{\partial S}{\partial a} = 2\sum_{i=1}^{N} [y_i - a - bx_i](-1) = 0$$

$$\frac{\partial S}{\partial b} = 2\sum_{i=1}^{N} [y_i - a - bx_i](-x_i) = 0$$

$$a\sum 1 + b\sum x_i = \sum y_i$$
$$a\sum x_i + b\sum x_i^2 = \sum x_i y_i$$

$$\Rightarrow \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum 1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

Data fitting to a polynomial(I)

Model $y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = \sum_{j=0}^{n} a_j x^j$

At the minimum error $S = \sum_{i=1}^{N} [y_i - \sum_{j=0}^{n} a_j x_i^j]^2$

$$\frac{\partial S}{\partial a_k} = \sum_{i=1}^{N} 2[y_i - \sum_{j=0}^{n} a_j x_i^j](-x_i^k) = 0$$

$$k = 0, 1, \cdots, n$$

$$\left\{ \begin{array}{l} \text{(n+1) simultaneous eg. (linear)} \\ \text{(n+1) unknowns} \end{array} \right.$$

Data fitting to a polynomial(II)

$$
\begin{aligned}
a_0 \sum 1 + a_1 \sum x_i + \cdots + a_n \sum x_i^n &= \sum y_i \\
a_0 \sum x_i + a_1 \sum x_i^2 + \cdots + a_n \sum x_i^{n+1} &= \sum x_i y_i \\
\vdots \qquad\qquad & \qquad \vdots \\
a_0 \sum x_i^n + a_1 \sum x_i^{n+1} + \cdots + a_n \sum x_i^{2n} &= \sum x_i^n y_i
\end{aligned}
$$

Rewriting the eq.'s into a matrix form

$$\boxed{\mathbf{F}^{\mathbf{T}}\mathbf{F}\mathbf{a} = \mathbf{F}^{\mathbf{T}}\mathbf{y}}$$

Linear equation

where

$$
\begin{cases}
\mathbf{a} = [\mathbf{a}_1\, \mathbf{a}_2 \cdots \mathbf{a}_n]^{\mathbf{T}} \\
\mathbf{y} = [\mathbf{y}_1\, \mathbf{y}_2 \cdots \mathbf{y}_n]^{\mathbf{T}} \\
\mathbf{F}^{\mathbf{T}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_N \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \cdots & x_N^n \end{bmatrix}
\end{cases}
$$

General linear least-square(I)

Model $$y(x) = \sum_{j=1}^{M} c_j f_j(x)$$

Error $$\begin{aligned} e_i &= y_i - y(x_i) \\ &= y_i - \sum_{j=1}^{M} c_j f_j(x_i) \end{aligned}$$

Sum of Errors $$S = \sum_{i=1}^{N} [y_i - \sum_{j=1}^{M} c_j f_j(x_i)]^2$$

At the minimum error

$$\frac{\partial S}{\partial c_k} = 2\sum_{i=1}^{N} \underbrace{[y_i - \sum_{j=i}^{M} c_j f_j(x_i)]}_{= e_i} \underbrace{(-f_k(x_i))}_{= \frac{\partial e_i}{\partial c_k}} = 0 \qquad k = 1, 2, \cdots, M$$

General linear least-square(II)

Matrix form

$$\mathbf{J}^T \mathbf{e} = \mathbf{0}$$

where $\mathbf{e} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \cdots \ \mathbf{e}_N]^T$

$$\mathbf{J}^T = \begin{bmatrix} \frac{\partial e_1}{\partial c_1} & \frac{\partial e_2}{\partial c_1} & \cdots & \frac{\partial e_N}{\partial c_1} \\ \frac{\partial e_1}{\partial c_2} & \ddots & & \frac{\partial e_N}{\partial c_2} \\ \vdots & & \ddots & \vdots \\ \frac{\partial e_1}{\partial c_M} & \frac{\partial e_2}{\partial c_M} & \cdots & \frac{\partial e_N}{\partial c_M} \end{bmatrix} = -\begin{bmatrix} f_1(x_1) & f_1(x_2) & \cdots & f_1(x_N) \\ f_2(x_1) & \ddots & & f_2(x_N) \\ \vdots & & \ddots & \vdots \\ f_M(x_1) & f_M(x_2) & \cdots & f_M(x_N) \end{bmatrix}$$

Since

$$\mathbf{e} = \mathbf{y} - \mathbf{J}\mathbf{c}, \quad \mathbf{c} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \cdots \ \mathbf{c}_M]^T$$

We obtain

$$\mathbf{J}^T y - \mathbf{J}^T \mathbf{J}\mathbf{c} = \mathbf{0}$$

$$\boxed{\therefore \quad \mathbf{J}^T \mathbf{J}\mathbf{c} = \mathbf{J}^T \mathbf{y}}$$

Homework #6: Programming

[Due: Nov. 25]

Part 1: Given data

<Linear Data Fitting>

Given N observations $(x_i,\ y_i,\ x_i',\ y_i'),\ i = 1, 2, \cdots, N$ and a linear mapping model:

$$x' = a_1 x + a_2 y + a_3$$
$$y' = a_4 x + a_5 y + a_6$$

find the "best" (in the least-square sense) set of parameters $\boldsymbol{a} = (a_1, \cdots, a_6)$ that fits the given data.

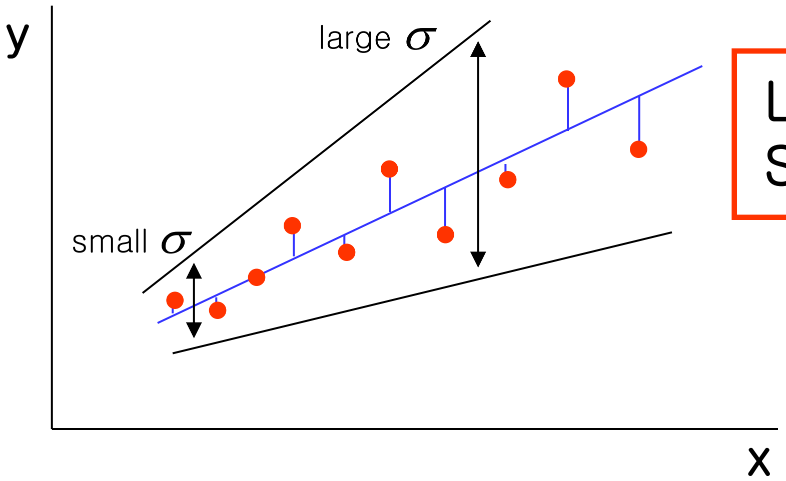
Data files are given in the course homepage (fitdata1.dat, fitdata2.dat, fitdata3.dat).

Chi-Square Fitting

If each data point (x_i, y_i) has its own, known standard deviation σ_i, we can formulate a weighted least-square problem :

$$\chi^2 \equiv \sum_{i=1}^{N} \left(\frac{y_i - y(x_i, a_1, \cdots, a_M)}{\sigma_i} \right)^2$$

Called the "chi-square"



Large σ : less weighted
Small σ: more weighted

Eg. Chi-square fitting

- Fitting to a line

$$x^2(a,b)=\sum_{i=1}^{N}\left(\frac{y_i-a-bx_i}{\sigma_i}\right)^2$$

At the minimum error

$$\frac{\partial x^2}{\partial a}=-2\sum\frac{y_i-a-bx_i}{\sigma_i^2}=0$$

$$\frac{\partial x^2}{\partial b}=-2\sum\frac{x_i(y_i-a-bx_i)}{\sigma_i^2}=0$$

Define

$$S\equiv\sum\frac{1}{\sigma_i^2}\qquad S_x\equiv\sum\frac{x_i}{\sigma_i^2}\qquad S_y\equiv\sum\frac{y_i}{\sigma_i^2}$$

$$S_{xx}\equiv\sum\frac{x_i^2}{\sigma_i^2}\qquad S_{xy}\equiv\sum\frac{x_iy_i}{\sigma_i^2}$$

Then we obtain

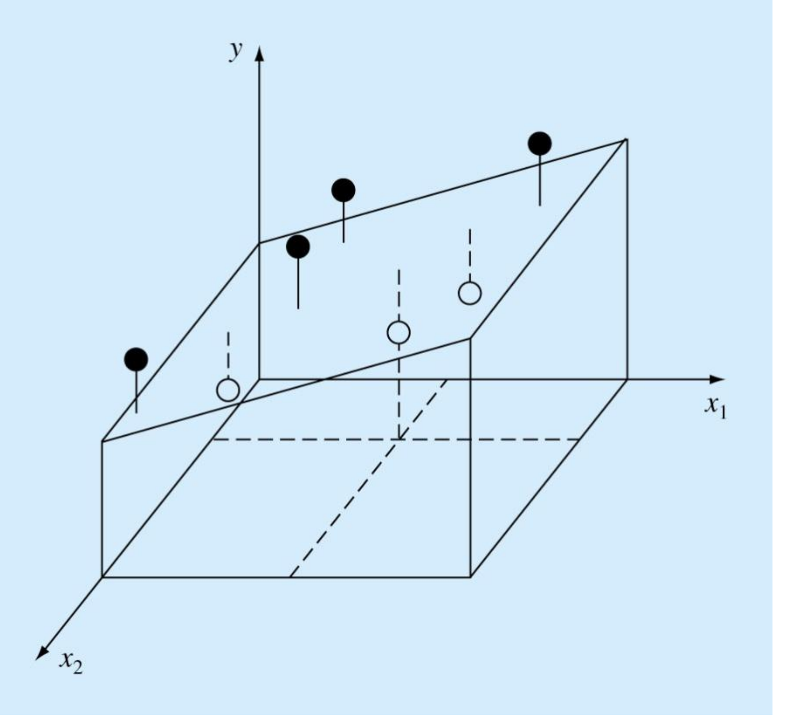
$$aS+bS_x=S_y$$
$$aS_x+bS_{xx}=S_{xy}$$

$$\Rightarrow\quad\begin{bmatrix}S & S_x\\ S_x & S_{xx}\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix}=\begin{bmatrix}S_x\\ S_{xy}\end{bmatrix}$$

Multi-Dimensional Fit

Model $$y(\mathbf{x}) = \sum_{j=1}^{M} c_j f_j(\mathbf{x})$$

Error $$e_i = y_i - y(\mathbf{x}_i)$$



Similarly to the derivation for the 1D fits, we get

$$\mathbf{J}^T \mathbf{J} \mathbf{c} = \mathbf{J}^T \mathbf{y}$$

The only difference is that the dimension of **x** is generalized to an arbitrary dimension

Nonlinear Models

Model $y(\mathbf{x}) = \underbrace{f}_{\text{nonlinear fcn. of } \mathbf{a}}(\mathbf{x}, \underbrace{\mathbf{a}}_{\text{parameters}})$

Eg.)

$$y = f(x_1, x_2, a_1, a_2, a_3, a_4)$$
$$= x_1 + a_3 x_2 - \frac{a_2}{a_4} x_1^2 + \frac{a_1}{a_4} x_1 x_2 - a_2 x_4$$

<u>Problem</u>

Given $(x_{1i}, x_{2i}, y_i), \quad i = 1, 2, \cdots, N$

find the least-square solution $\mathbf{a}$

Nonlinear fitting: easy case

- Some Nonlinear functions can be transformed into a linear form.

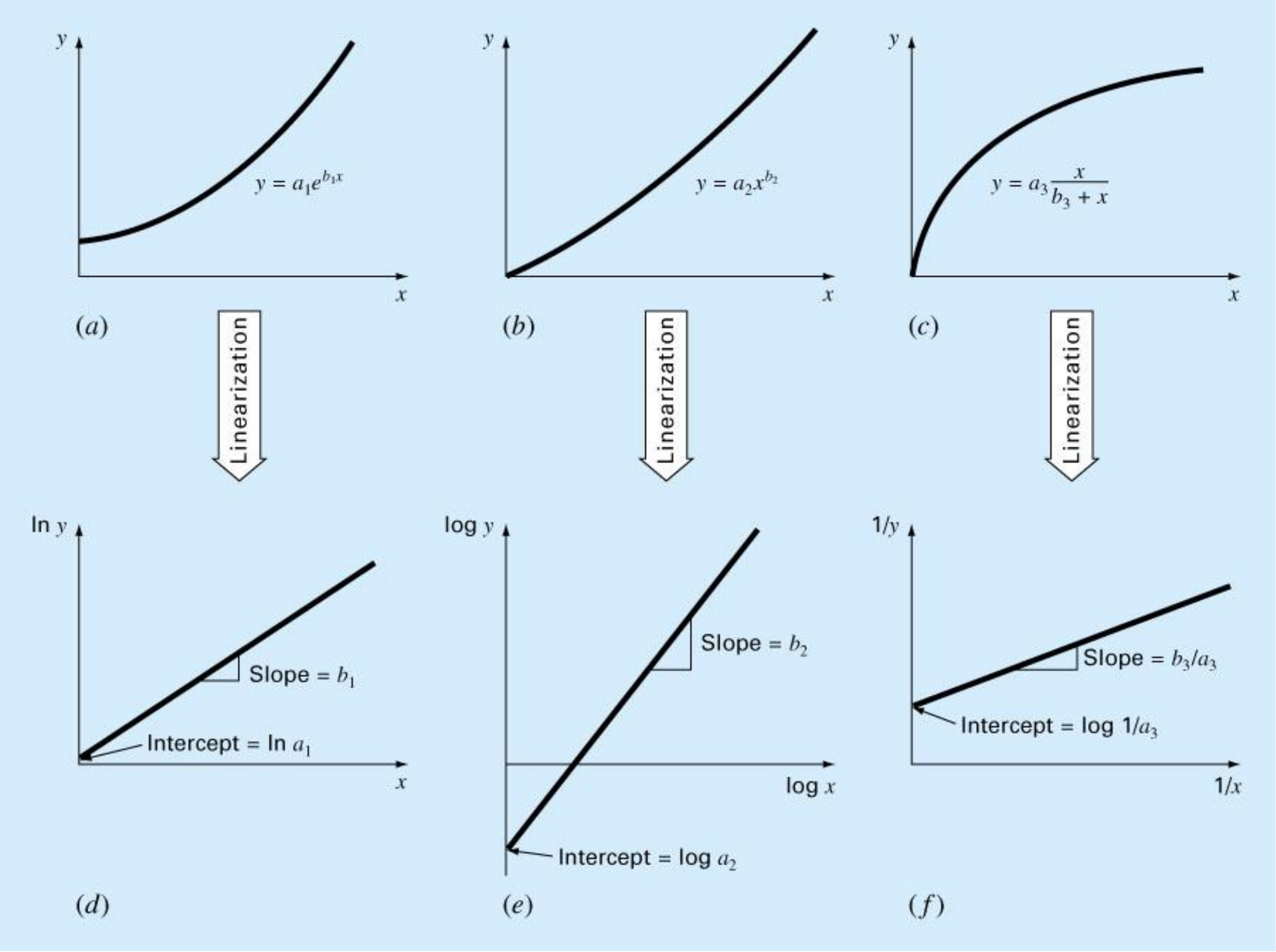
$$y = \alpha e^{\beta x} \xrightarrow{\ln} \ln y = \ln \alpha + \beta x$$

$$\boxed{y' = \alpha' + \beta' x}$$

; linear

Simple BUT
Not an optimum fitting!

Nonlinear fitting by linearization



Nonlinear Least-Square Fitting

Model $y = f(\mathbf{x}, \mathbf{a})$

where $\mathbf{a} = [a_1, a_2, \cdots, a_M]$

Cost function $\chi^2(\mathbf{a}) = \sum_{i=1}^{N} \left[\frac{y_i - f(\mathbf{x_i}, \mathbf{a})}{\sigma_i} \right]^2$

Problem

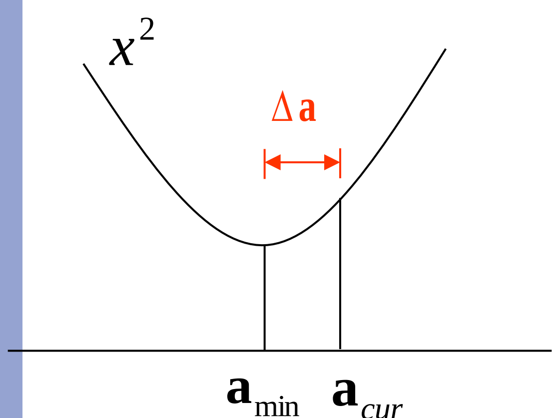
Minimize χ^2, $w.r.t.\ \mathbf{a}$

$$\rightarrow \boxed{\mathbf{a}^* = \arg\min_{\mathbf{a}} \chi^2(\mathbf{a})}$$

Levenberg-Marquardt Method(I)

- Some insight

$\mathbf{a}_{cur}$: Near the minimum



$$\mathbf{a}_{\min} = \mathbf{a}_{cur} + \mathbf{H}^{-1}[-\nabla\chi^2(\mathbf{a}_{cur})]$$

$$\Delta\mathbf{a} = \mathbf{a}_{\min} - \mathbf{a}_{cur} = \mathbf{H}^{-1}[-\nabla\chi^2(\mathbf{a}_{cur})]$$

$$\therefore \underbrace{\mathbf{H}}_{\text{Hessian matrix}} \underbrace{\Delta\mathbf{a}}_{\text{update term}} = -\underbrace{\nabla\chi}_{\text{gradient}}{}^2$$

Inverse–Hessian method

$\mathbf{a}_{cur}$: Far from the minimum

$$\Delta\mathbf{a} = -const \times \nabla\chi^2(\mathbf{a}_{cur})$$

Steepest Descent method

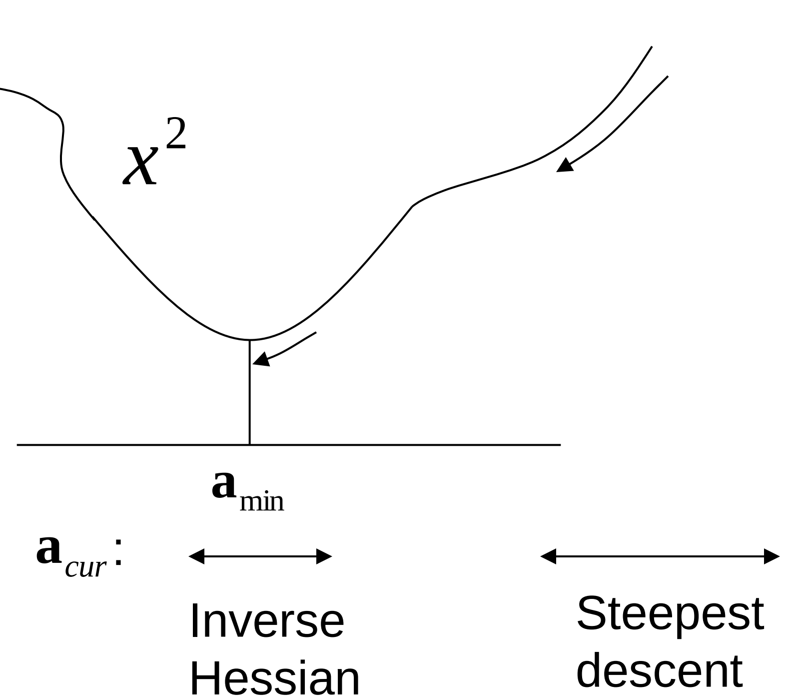
Levenberg-Marquardt Method(II)

Idea

- Start : steepest descent method

gradually change

- End : Inverse-Hessian method



Levenberg-Marquardt Method(III)

- Algorithm

i) Guess $\mathbf{a}_{cur}$
ii) compute $\chi^2(\mathbf{a}_{cur})$
iii) pick a modest λ, say $\lambda = 0.001$
iv) solve

$$H' \cdot \Delta \mathbf{a} = -\nabla \chi^2(\mathbf{a}_{cur}) \qquad \text{①}$$

where $H' = H + \lambda I$

v) if $\mathbf{x}^2(\mathbf{a}_{cur} + \Delta\mathbf{a}) \geq \mathbf{x}^2(\mathbf{a}_{cur})$

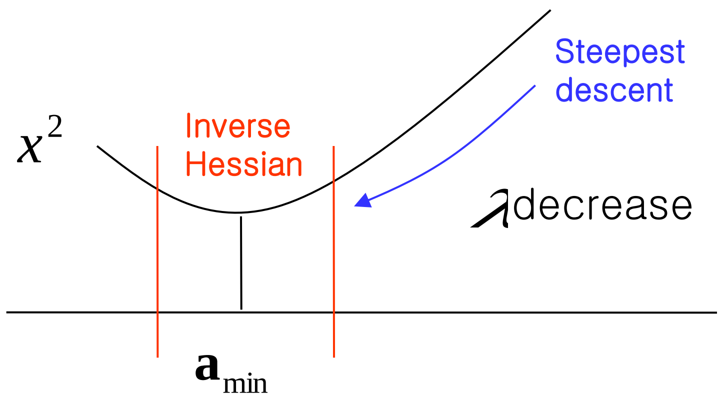
increase λ by a factor of 10 and go to iv)

else

decrease λ by a factor of 10 and

update $\mathbf{a}_{cur} = \mathbf{a}_{cur} + \Delta\mathbf{a}$ and go to iv)

Levenberg-Marquardt Method(IV)



※ For large λ, in eq.①,
the $\mathbf{H}'$ is diagonal dominant

⇨ eq.① is close to steepest descent!

Calculation of the gradient

- Calculation of the gradient and the Hessian of

$$\chi^2(\mathbf{a}) = \sum_{i=1}^{N} \left[\frac{y_i - y(\mathbf{x_i}, \mathbf{a})}{\sigma_i} \right]^2$$

- Gradient

$$\frac{\partial \chi^2}{\partial a_k} = -2 \sum_{i=1}^{N} \frac{[y_i - y(\mathbf{x_i}, \mathbf{a})]}{\sigma_i^2} \frac{\partial y(\mathbf{x_i}, \mathbf{a})}{\partial a_k}, \quad k = 1, 2, \cdots, M$$

$$\nabla \chi^2(\mathbf{a}) = \left[\frac{\partial \chi^2}{\partial a_1} \ \frac{\partial \chi^2}{\partial a_2} \ \cdots \ \frac{\partial \chi^2}{\partial a_M} \right]^{\mathbf{T}}$$

Calculation of the Hessian

ignore

$$\frac{\partial^2 \chi^2}{\partial a_k \partial a_l} = 2\sum_{i=1}^{N}\frac{1}{\sigma_i^2}\left[\frac{\partial y(\mathbf{x}_i,\mathbf{a})}{\partial a_k}\frac{\partial y(\mathbf{x}_i,\mathbf{a})}{\partial a_l} - \cancel{(y_i - y(\mathbf{x}_i,\mathbf{a}))\frac{\partial^2 y(\mathbf{x}_i,\mathbf{a})}{\partial a_k \partial a_l}}\right]$$

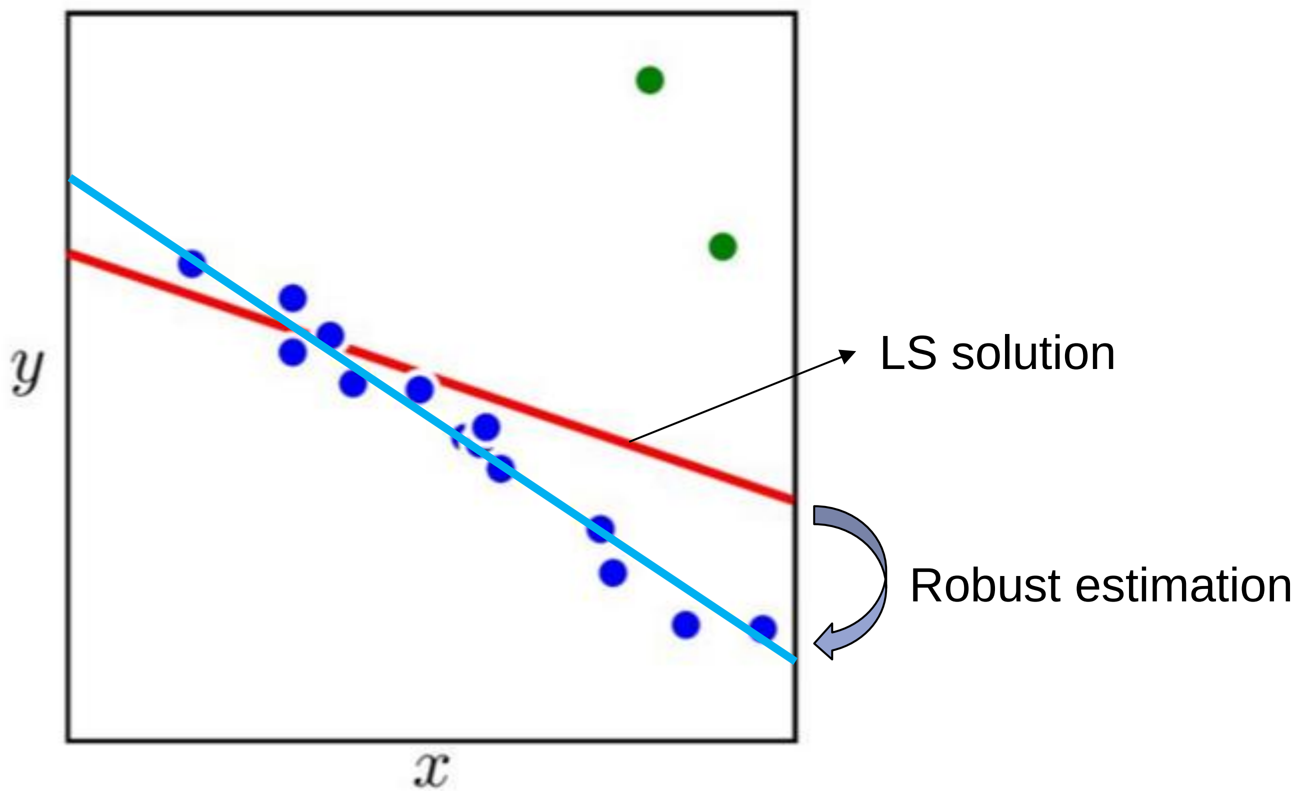
Sensitive to noise

$$\mathbf{H} = \begin{bmatrix} \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_1}\right)^2 & \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_1}\frac{\partial y}{\partial a_2}\right) & \cdots & \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_1}\frac{\partial y}{\partial a_M}\right) \\ \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_2}\frac{\partial y}{\partial a_1}\right) & \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_2}\right)^2 & \cdots & \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_2}\frac{\partial y}{\partial a_M}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_M}\frac{\partial y}{\partial a_1}\right) & \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_M}\frac{\partial y}{\partial a_2}\right) & \cdots & \sum_{i=1}^{N}\frac{2}{\sigma_i^2}\left(\frac{\partial y}{\partial a_M}\right)^2 \end{bmatrix}$$

; symmetric

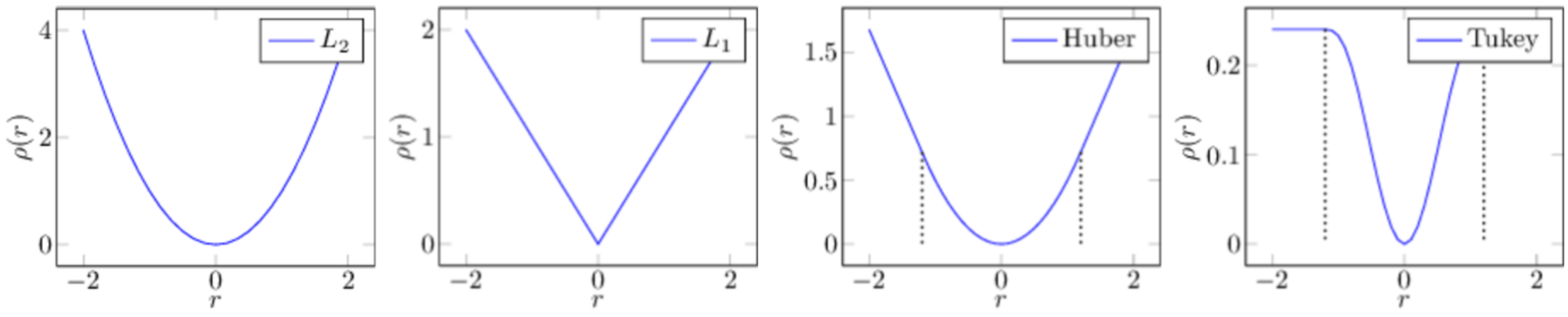
Robust data fitting

- Outliers → Bad LS solution

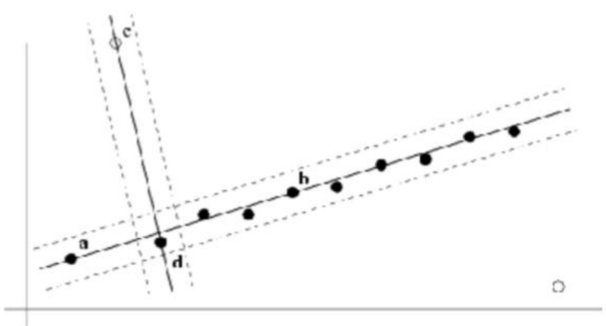


Robust estimation

- Approach 1: Using robust measures



- Approach 2: Random sampling



RANdom SAmple Consensus:
RANSAC determines the consistency of a hypothesis by counting the number of points within a threshold RANSAC determines the consistency of a hypothesis by counting the number of points within a threshold distance (given by the dashed line).

Homework : Programming

Part 2: Nonlinear data fitting

Model: 2D transformation between given images

$$x' = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + 1}$$

$$y' = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + 1}$$

1. Establish the feature correspondence between the two images (Output: x_i, y_i, x_i' , y_i', i=1~N)
2. Find the parameters using the correspondence data.

Add zero mean Gaussian noise(SD=1,10,30) on the image coordinates and discuss the accuracy of estimation.