Method of Least Squares

IPCST
Seoul National University

Target Systems of Equations

- Overdetermined systems
  - more equations than unknowns
- Inexactly specified systems

$$\text{set of data } \{\mathbf{x}_i, \mathbf{y}_i\}$$

$$\text{model function } \mathbf{y} = f(\mathbf{x}, \mathbf{p})$$

$\mathbf{p}$: parameters (what to be determined)

Linear Regression

- $f(x, a, b) = ax + b$

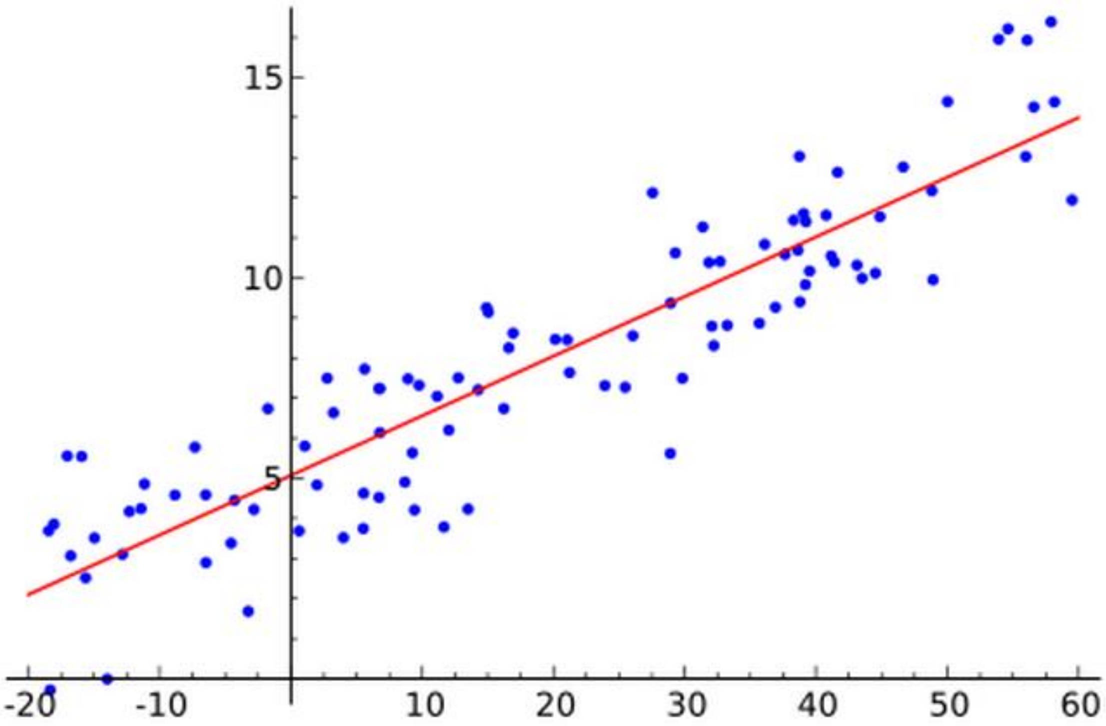


Figure from Wikipedia

Linear Regression

- $f(x, a, b) = ax + b$
- Error function

$$J(a,b) = \sum_{i=0}^{M} (ax_i + b \ - y_i)^2$$

  - Minimizing this sum of the error squares

    "**Least squares**"

  - $\frac{\partial J}{\partial a} = \frac{\partial J}{\partial b} = 0$

Linear Regression

- $J(a,b) = \sum_{i=0}^{m}(ax_i + b - y_i)^2$

  $\frac{\partial J}{\partial a} = 0 \rightarrow a\left(\sum_{i=0}^{m} x_i^2\right) + b\left(\sum_{i=0}^{m} x_i\right) = \sum_{i=0}^{m} x_i y_i$

  $\frac{\partial J}{\partial b} = 0 \rightarrow a\left(\sum_{i=0}^{m} x_i\right) + b\underbrace{(m+1)}_{\text{Number of datapoints}} = \sum_{i=0}^{m} y_i$

  ➡ System of linear equations

  – You can get a & b.

Do It Yourself

- Wen Shen Example 8.1
  - Fit this data

T_k	0	1	2	3	4	5	6	7
S_k	1.15	2.32	3.32	4.53	5.65	6.97	8.02	9.23

to

$$S = aT + b$$

  - Plot the data and the fitted function

Linear Models for Curve Fitting

- Straight line

  $f(t) \approx \beta_1 t + \beta_2$

- Polynomials

  $f(t) \approx \beta_1 t^{n\text{-}1} + \cdot \cdot \cdot + \beta_{n\text{-}1} t + \beta_n$

- Linear combination of no parameter functions

Design Matrix

- If a model $f(t)$ can be expressed by a linear combination of n basis functions $g_j(t)$'s,

$$f(t) \approx \beta_1 g_1(t) + \cdot \cdot \cdot + \beta_{n-1} g_{n-1}(t) + \beta_n g_n(t)$$

- Design matrix **X**

$$X_{i,j} = g_j(t_i)$$

$$\mathbf{y} \approx \mathbf{X\beta} \qquad \text{or} \qquad \mathbf{X\beta} \approx \mathbf{y}$$

data points: (t_i, y_i)
$y_i \approx f(t_i)$

Residuals and Squares

- Residuals $r_i = y_i - \sum_{j=1}^{n} \beta_j g_j(t_i)$, $i = 0,1,\ldots,m$

$$\mathbf{r} = \mathbf{y} - \mathbf{X\beta}$$

- **Least squares:** Minimize the sum of the squares of the residuals (the error function)

$$J(\boldsymbol{\beta}) = \sum_{i=0}^{m} r_i^2 = \sum_{i=0}^{m} \left[y_i - \sum_{j=1}^{n} \beta_j g_j(t_i) \right]^2$$

Normal Equations

- From $\partial J/\partial\beta_j = 0$,

$$\sum_{i=0}^{m} 2\left[y_i - \sum_{j=1}^{n} \beta_j g_j(t_i)\right] g_k(t_i) = 0$$

$$\sum_{j=1}^{n} \beta_j \left\{\sum_{i=0}^{m} g_j(t_i) g_k(t_i)\right\} = \sum_{i=0}^{m} y_i g_k(t_i)$$

$$\mathbf{X}^{\mathbf{T}}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^{\mathbf{T}}\mathbf{y} \qquad \boldsymbol{\beta} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\,\mathbf{X}^{\mathbf{T}}\mathbf{y}$$

Quasi-linear Models

- Log-linear

$$f(t) = \beta \exp(\lambda t)$$

$$\rightarrow \ln f = \ln \beta + \lambda t$$

$$f(t) = \beta c^t$$

$$\rightarrow \ln f = \ln \beta + t \ln c$$

- Log-polynomial

$$(\beta_1 t^{n-1} + \cdot\cdot\cdot + \beta_{n-1} t + \beta_n)\cdot\log(\alpha t)$$

$$= \gamma_1 t^{n-1} + \cdot\cdot\cdot + \gamma_{n-1} t + \gamma_n$$

$$+(\beta_1 t^{n-1} + \cdot\cdot\cdot + \beta_{n-1} t + \beta_n)\cdot\log(t)$$

Nonlinear Models

- Rational functions

$$f(t) \approx \frac{\beta_1 t^{n-1} + \cdots + \beta_{n-1} t + \beta_n}{\alpha_1 t^{n-1} + \cdots + \alpha_{n-1} t + \alpha_n}$$

- Exponentials

$$f(t) \approx \beta_1 \exp(-\lambda_1 t) + \cdots + \beta_n \exp(-\lambda_n t)$$

- Gaussians

$$f(t) \approx \beta_1 \exp\left[-\left(\frac{t-\mu_1}{\sigma_1}\right)^2\right] + \cdots + \beta_n \exp\left[-\left(\frac{t-\mu_n}{\sigma_n}\right)^2\right]$$

Nonlinear Models

- $\beta\sin(\alpha t)$, $\beta\cos(\alpha t)$
- $\beta t\cdot\sin(\alpha t)$, $\beta t\cdot\cos(\alpha t)$
- Polynomial $\times$ [$\sin(\alpha t)$ or $\cos(\alpha t)$]

- Parts of circle, ellipse, parabola or hyperbola

$$\beta\sqrt{\text{polynomial }(\alpha)}$$

Nonlinear Least Squares

- Wen Shen Example 8.4

$$y(t) = at \cdot \sin(bt)$$

  - Error function

$$J(a,b) = \sum_{i=0}^{m} [y_i - at_i \cdot \sin(bt_i)]^2$$

  - At minimum, $\partial J/\partial a = \partial J/\partial b = 0$:

$$2\sum_{i=0}^{m} [y_i - at_i \cdot \sin(bt_i)] \cdot [t_i \cdot \sin(bt_i)] = 0$$

$$2\sum_{i=0}^{m} [y_i - at_i \cdot \sin(bt_i)] \cdot \left[at_i^2 \cdot \cos(bt_i)\right] = 0$$

Nonlinear Least Squares

- Wen Shen Example 8.4
  - Newton method
    - Find a, b such that $\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \partial J/\partial a \\ \partial J/\partial b \end{pmatrix} = 0$

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \boldsymbol{\delta} \qquad \text{where } \mathbf{x}_n = \begin{pmatrix} a_n \\ b_n \end{pmatrix}$$
$$\mathbf{H}_J(\mathbf{x}_n)\boldsymbol{\delta} = \mathbf{f}(\mathbf{x}_n)$$

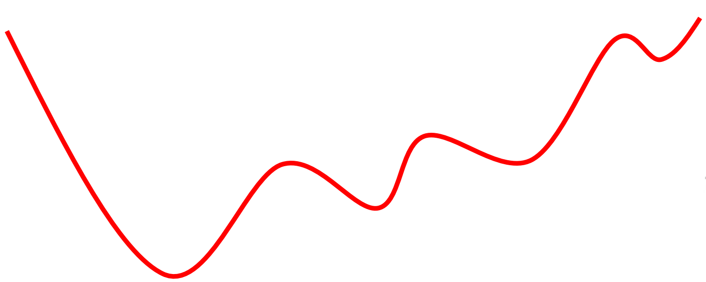
    - $\mathbf{H}_J$: Hessian (matrix) of J (= Jacobian of $\mathbf{f}$)

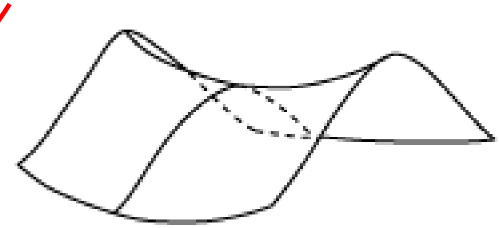
$$\mathbf{H}_J = \begin{pmatrix} \dfrac{\partial^2 J}{\partial a^2} & \dfrac{\partial^2 J}{\partial a \partial b} \\ \dfrac{\partial^2 J}{\partial a \partial b} & \dfrac{\partial^2 J}{\partial b^2} \end{pmatrix}$$

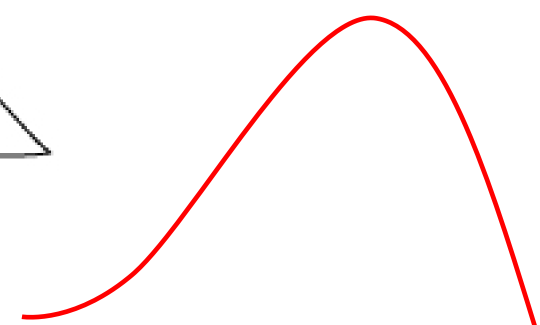
參考: Nonlinear Least Squares

- Nonlinear least squares methods are usually iterative and need initial values.

- Nonlinear least squares methods occasionally fail to find the true minimum.
  - Not convergent
  - A local minimum or a non-stationary point







参考: Separable Least Squares

- In case that linear parameters are explicitly distinguished from nonlinear parameters
  - Nonlinear parameters α's: nonlinear methods
  - Linear parameters *β's*: linear least squares
- Process
  1. Initial α's
  2. Linear least squares – *β's* and *J*
  3. Nonlinear least squares – varying α's to find the minimum *J* by repeating the procedure 2.

LS for Continuous Functions

– Wen Shen pp. 164~168

- Instead of a data set $\{x_i, y_i\}$, a function $f(x)$ on $[a, b]$ is given.
- Find a combination of simple functions to approximate to $f(x)$
  – Consider a combination of orthogonal basis functions $g_i(x)$ on $[a, b]$

$$g(x) = \sum_{i=1}^{n} c_i g_i(x)$$

  – Error function: $E(f, g) = \int_a^b [f(x) - g(x)]^2 \, dx$

LS for Continuous Functions

- Find a combination of simple functions to approximate to $f(x)$
  - Error function: $E(f,g) = \int_a^b [f(x) - g(x)]^2\, dx$

    → $E(c_1, c_2, \dots, c_n) = \int_a^b [f(x) - \sum_{i=1}^n c_i g_i(x)]^2\, dx$
  - Minimum condition: $\partial E / \partial c_i = 0$ for every c_i

$$\frac{\partial E}{\partial c_i} = -2 \int_a^b g_i(x) \left[f(x) - \sum_{j=1}^n c_j g_j(x) \right] dx = 0$$

$$\int_a^b g_i(x) f(x) dx - \int_a^b g_i(x) \sum_{j=1}^n c_j g_j(x)\, dx = 0$$

LS for Continuous Functions

- Find a combination of simple functions to approximate to $f(x)$

$$\sum_{j=1}^{n} c_j \int_a^b g_i(x) g_j(x) dx = \int_a^b g_i(x) f(x) dx$$

  – This is a matrix-vector equation

$$\mathbf{Ac} = \mathbf{b}$$

$$A_{ij} = \int_a^b g_i(x) g_j(x) dx$$

$$b_{\boldsymbol{i}} = \int_a^b g_i(x) f(x) dx$$

LS for Continuous Functions

- Orthogonal basis functions
  - Definition: basis functions $g_i(x)$ with the condition

$$\int_a^b g_i(x) g_j(x) dx = 0 \ \text{ if } i \neq j$$

  - Ex.) Legendre polynomials
  - Orthogonal basis functions make the matrix on the previous page diagonal.

$$\mathbf{Ac} = \mathbf{b} \quad \Longrightarrow \quad c_i = b_i / A_{ii}$$

References

- Wen Shen,
  An Introduction to Numerical Computation
- C. Moler,
  Numerical Computing with MATLAB
- Wikipedia
- C. T. Kelly,
  Iterative Methods for Optimization

Further Study

- **Gauss-Newton** method