Least square fit of a line/polynomial to input/output data

Prashant Dave

Chemical Engg.,
Indian Institute of Technology Bombay

Jan, 2012

Outline

Today's focus

- Scilab is free.

Today's focus

- Scilab is free.
- Matrix/loops syntax is same as for Matlab.

Today's focus

- Scilab is free.
- Matrix/loops syntax is same as for Matlab.
- Scilab provides all basic and many advanced tools.

Today's focus

- Scilab is free.
- Matrix/loops syntax is same as for Matlab.
- Scilab provides all basic and many advanced tools.
- Today: best fit: line and polynomial : reglin command

Linear fit

Given n samples of (x, y) pairs:
x_i and y_i for $i = 1, \ldots, n$, we expect following equation is satisfied

$$y_i = a_1 x_i + a_0 \qquad \text{for } i = 1, \ldots, n \tag{1}$$

for some constants a_1 and a_0.

Linear fit

Given n samples of (x, y) pairs:
x_i and y_i for $i = 1, \ldots, n$, we expect following equation is satisfied

$$y_i = a_1 x_i + a_0 \qquad \text{for } i = 1, \ldots, n \tag{1}$$

for some constants a_1 and a_0.
x: independent variable (exactly known),
y: dependent variable (some error in measuring it)
x_i and y_i fall on some line with slope a_1 and 'y-intercept'$=a_0$.
The 'line fit' problem:
Find these constants a_1 and a_0.
'Best' fit?

Best fit

The true relationship is $y_i = a_{0a} + a_{1a}x_i$, but due to noise (for example in measurements), the available x_i, y_i pairs will not satisfy the equation exactly.

Best fit

The true relationship is $y_i = a_{0a} + a_{1a}x_i$, but due to noise (for example in measurements), the available x_i, y_i pairs will not satisfy the equation exactly.
Least-square-fit problem:
Given n samples of (x_i, y_i) pairs,

Best fit

The true relationship is $y_i = a_{0a} + a_{1a}x_i$, but due to noise (for example in measurements), the available x_i, y_i pairs will not satisfy the equation exactly.

Least-square-fit problem:

Given n samples of (x_i, y_i) pairs,

find constants a_1 and a_0 such that the 'total square error'

$$\sum_{i=1}^{n}(y_i - a_1x_i - a_0)^2 \tag{2}$$

is least.

Scilab Tool: reglin

[a1,a0,sig]=reglin(x,y)

- x: $1 \times n$ vector (for n data points)
- y: $1 \times n$ vector (for n data points)
- a1: slope, a0: intercept
- sig: standard deviation of fit error: lower is “better”

Straight line fit example

Generate data using known (actual) values of a0 and a1.
Add noise to dependent variable.
Using noisy data, estimate a0 and a1.

1. True data generation: $y = 5 + 2x$ for $x = 0 : 10$.
2. Noise addition: $y = y + e$ where e is normally distributed noise with mean 0 and standard deviation 2.
3. Least squares fit: $[a1, a0, sig] = reglin(x, y)$.
4. Plot: (xi,yi) pairs, true (noise free) line, fitted line

Noise generation

Generate a vector of length n from a normal distribution with mean a and standard deviation b.

Noise generation

Generate a vector of length n from a normal distribution with mean a and standard deviation b.

1. rand('seed',10): get repeatable random numbers by initializing seed.

Noise generation

Generate a vector of length n from a normal distribution with mean a and standard deviation b.

1. rand('seed',10): get repeatable random numbers by initializing seed.
2. rand('normal'): generate from a normal distribution.

Noise generation

Generate a vector of length n from a normal distribution with mean a and standard deviation b.

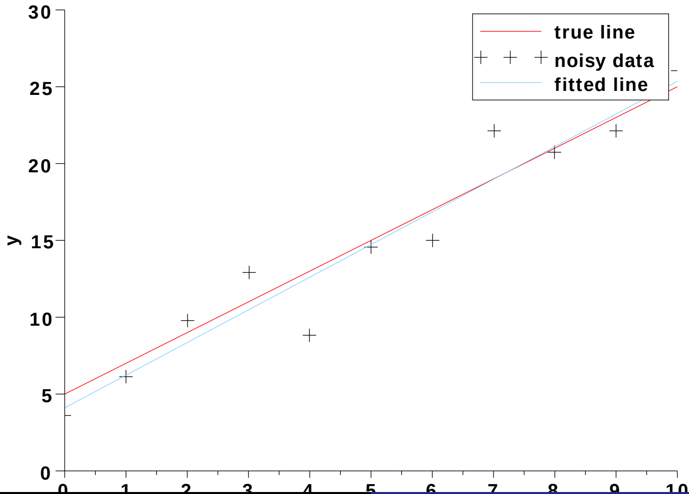
1. rand('seed',10): get repeatable random numbers by initializing seed.
2. rand('normal'): generate from a normal distribution.
3. rand(x): generate a vector of same length as x.

Noise generation

Generate a vector of length n from a normal distribution with mean a and standard deviation b.

1. rand('seed',10): get repeatable random numbers by initializing seed.
2. rand('normal'): generate from a normal distribution.
3. rand(x): generate a vector of same length as x.
4. a+b*rand(x): generate with mean a and standard deviation b.

Plots for example



Higher order polynomial least-square fit

Suppose we expect y_i satisfies the following equation:

$$y_i = a_2 x_i^2 + a_1 x_i + a_0$$

Higher order polynomial least-square fit

Suppose we expect y_i satisfies the following equation:

$$y_i = a_2 x_i^2 + a_1 x_i + a_0$$

Points (x_i, y_i) are sitting on a parabola.

Higher order polynomial least-square fit

Suppose we expect y_i satisfies the following equation:

$$y_i = a_2 x_i^2 + a_1 x_i + a_0$$

Points (x_i, y_i) are sitting on a parabola.
Problem (more generally):

Higher order polynomial least-square fit

Suppose we expect y_i satisfies the following equation:

$$y_i = a_2 x_i^2 + a_1 x_i + a_0$$

Points (x_i, y_i) are sitting on a parabola.
Problem (more generally):
Given n samples of (x_i, y_i) pairs and some choice of degree d.

$$y_i = a_d x_i^d + a_{d-1} x_i^{d-1} + \ldots + a_1 x_i + a_0$$

Higher order polynomial least-square fit

Suppose we expect y_i satisfies the following equation:

$$y_i = a_2 x_i^2 + a_1 x_i + a_0$$

Points (x_i, y_i) are sitting on a parabola.
Problem (more generally):
Given n samples of (x_i, y_i) pairs and some choice of degree d.

$$y_i = a_d x_i^d + a_{d-1} x_i^{d-1} + ... + a_1 x_i + a_0$$

Find constants $a_d, \ldots a_1$ and a_0 such that the 'total square error'

$$\sum_{i=1}^{n} (a_d x_i^d + a_{d-1} x_i^{d-1} + \cdots + a_1 x_i + a_0 - y_i)^2 \qquad (3)$$

is least.

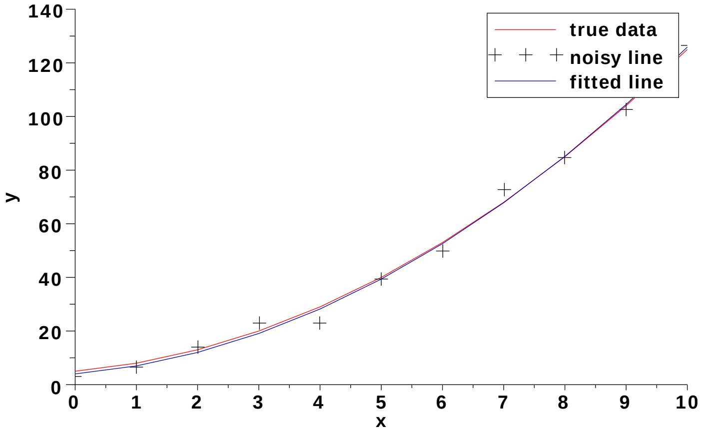
Still a linear regression problem

The unknowns a_i enter the problem linearly.

Still a linear regression problem

The unknowns a_i enter the problem linearly.
(i.e. a_i's are not getting squared, or multiplied to each other.)
[slopes, intercept] = reglin(X,y)
where $X = [x; x^2]$: a matrix with two regressors (one in each row)
y: a row vector with same number of columns as X.
slopes: the coefficients a1,a2
intercept: the coefficient a0
sig : standard deviation of the residual.

Second order fit example



More than one independent variables

Suppose y depends on independent variables x_1, x_2, etc.

$$y_i = a_0 + a_1 x_{1i} + a_2 x_{2i} + \ldots + a_p x_{pi}$$

A multiple linear regression problem (coefficients a_i still appear linearly)

More than one independent variables

Suppose y depends on independent variables x_1, x_2, etc.

$$y_i = a_0 + a_1 x_{1i} + a_2 x_{2i} + \ldots + a_p x_{pi}$$

A multiple linear regression problem (coefficients a_i still appear linearly)
[slopes,intercept]=reglin(X,y)
where X and y are matrix/vector with same number of columns.

More than one independent variables

Suppose y depends on independent variables x_1, x_2, etc.

$$y_i = a_0 + a_1 x_{1i} + a_2 x_{2i} + \ldots + a_p x_{pi}$$

A multiple linear regression problem (coefficients a_i still appear linearly)
[slopes,intercept]=reglin(X,y)
where X and y are matrix/vector with same number of columns.
(but X has many rows.)
Components in slopes = number of rows of X
(number of independent variables.)

Nonlinear Least Squares

The parameters to be estimated appear non-linearly in the model:
$y = f(x)$
Example, $y_i = a/(b + x_i)$

Nonlinear Least Squares

The parameters to be estimated appear non-linearly in the model:
$y = f(x)$
Example, $y_i = a/(b + x_i)$

- Want to choose parameters so as to minimize $\sum_{i=1}^{n}(y_i - f(x_i))^2$.
- Analytical solution usually not available.
- Use a numerical optimization technique.
- Scilab functions: lsqrsolve, leastsq (front end to optim function)

Thank You