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

Dr. Madhu N. Belur

Control & Computing group
Department of Electrical Engineering
Indian Institute of Technology Bombay
Email: belur@ee.iitb.ac.in

30th Nov, 2010

Outline

1. Scilab
2. Least squares
3. Scilab commands

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, z) pairs:
x_i and z_i for $i = 1, \ldots, n$, we expect following equation is satisfied

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

for some constants a_1 and a_0.

Linear fit

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

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

for some constants a_1 and a_0.
x_i and z_i fall on some line with slope a_1 and 'z-intercept'$=a_0$.
The 'line fit' problem:
Find these constants a_1 and a_0.
'Best' fit?

Best fit

Real situtation:
Equation (1) will not be satisfied 'exactly'.
Least-square-fit problem:
Given n samples of (x_i, z_i) pairs,

Best fit

Real situtation:
Equation (1) will not be satisfied 'exactly'.
Least-square-fit problem:
Given n samples of (x_i, z_i) pairs,
find constants a_1 and a_0 such that the 'total square error'

$$\sum_{i=1}^{n} (z_i - a_1 x_i - a_0)^2 \qquad (2)$$

is least.

Best fit

Real situtation:
Equation (1) will not be satisfied 'exactly'.
Least-square-fit problem:
Given n samples of (x_i, z_i) pairs,
find constants a_1 and a_0 such that the 'total square error'

$$\sum_{i=1}^{n}(z_i - a_1 x_i - a_0)^2 \tag{2}$$

is least.
Least square error

Higher order polynomial best fit

Least-square-fit problem (of higher order):

Higher order polynomial best fit

Least-square-fit problem (of higher order):
Suppose we expect z_i satisfies the following equation:

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

Higher order polynomial best fit

Least-square-fit problem (of higher order):
Suppose we expect z_i satisfies the following equation:

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

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

Higher order polynomial best fit

Least-square-fit problem (of higher order):
Suppose we expect z_i satisfies the following equation:

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

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

Higher order polynomial best fit

Least-square-fit problem (of higher order):
Suppose we expect z_i satisfies the following equation:

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

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

Higher order polynomial best fit

Least-square-fit problem (of higher order):
Suppose we expect z_i satisfies the following equation:

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

Points (x_i, z_i) are sitting on a parabola.
Problem (more generally):
Given n samples of (x_i, z_i) pairs and some choice of degree d.
Find constants $a_d, \ldots a_1$ and a_0 such that the 'total square error'

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

is least.

Higher order polynomial best fit

Least-square-fit problem (of higher order):
Suppose we expect z_i satisfies the following equation:

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

Points (x_i, z_i) are sitting on a parabola.
Problem (more generally):
Given n samples of (x_i, z_i) pairs and some choice of degree d.
Find constants $a_d, \dots a_1$ and a_0 such that the 'total square error'

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

is least.

reglin

Linear regression problem.
The unknowns a_i enter the problem linearly.

reglin

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.)
[a1, a0] = reglin(x,z)
where x and z are vectors with same number of columns.

$$\sum_i (z_i - a_1 x_i - a_0)^2$$

is minimized.

reglin

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.)
[a1, a0] = reglin(x,z)
where x and z are vectors with same number of columns.

$$\sum_i (z_i - a_1 x_i - a_0)^2$$

is minimized.
[a1, a0, sig] = reglin(x,z)
sig : standard deviation of the residual.

reglin

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.)
[a1, a0] = reglin(x,z)
where x and z are vectors with same number of columns.

$$\sum_i (z_i - a_1 x_i - a_0)^2$$

is minimized.
[a1, a0, sig] = reglin(x,z)
sig : standard deviation of the residual.
(Smaller sig means better fit.)

More than one independent variables

Suppose z depends on independent variables x_1, x_2, etc.
[a1,a0]=reglin(x,z)
where x and z are matrix/vector with same number of columns.

More than one independent variables

Suppose z depends on independent variables x_1, x_2, etc.
[a1,a0]=reglin(x,z)
where x and z are matrix/vector with same number of columns.
(but x has many rows.)

More than one independent variables

Suppose z depends on independent variables x_1, x_2, etc.
[a1,a0]=reglin(x,z)
where x and z are matrix/vector with same number of columns.
(but x has many rows.)
If x has more than one rows:
components in a1 = number of rows of x
(number of independent variables.)

Exercise plan

Generate data using known (actual) values of a0 and a1.

Exercise plan

Generate data using known (actual) values of a0 and a1.
Add noise.

Exercise plan

Generate data using known (actual) values of a0 and a1.
Add noise.
Use noisy data to estimate a0 and a1:

Random numbers

rand(3) generates a 1×1 random number (uniformly distributed between 0 and 1).

Random numbers

rand(3) generates a 1×1 random number (uniformly distributed between 0 and 1).
If P is an $n \times p$ matrix, then

Random numbers

rand(3) generates a 1×1 random number (uniformly distributed between 0 and 1).
If P is an $n \times p$ matrix, then
B=rand(P)
defines a random matrix B of the size of P.

Random numbers

rand(3) generates a 1×1 random number (uniformly distributed between 0 and 1).
If P is an $n \times p$ matrix, then
B=rand(P)
defines a random matrix B of the size of P.
(P is not overwritten. A new matrix B is defined.)

Exercises:

Generate data, add noise, and estimate the parameters back:
One independent variable (first).

a0a = 3;

a1a = 6;

(actual)

x = 1:10;

Exercises:

Generate data, add noise, and estimate the parameters back:
One independent variable (first).

a0a = 3;

a1a = 6;

(actual)

x = 1:10;

dev = 0.5;

Exercises:

Generate data, add noise, and estimate the parameters back:
One independent variable (first).

```
a0a = 3;
a1a = 6;
(actual)
x = 1:10;
dev = 0.5;
y = a1a*x + a0a + dev*(rand(x)-0.5)
```

Use this x and y to estimate a0 and a1
Compare a0 & a1 with actual a0a & a1a.

Exercises:

Generate data, add noise, and estimate the parameters back:
One independent variable (first).

```
a0a = 3;
a1a = 6;
(actual)
x = 1:10;
dev = 0.5;
y = a1a*x + a0a + dev*(rand(x)-0.5)
```

Use this x and y to estimate a0 and a1
Compare a0 & a1 with actual a0a & a1a.
For sig to compare with dev, note that sig is standard deviation (most relevant for normal distribution, and we used uniform distribution).

Fit 2nd order polynomial

Find a_2, a_1 and a_0 such that

$$y = a_1x + a_2x^2 + a_0$$

Suppose a0a = 3; a1a = 6; a2a = 2; (actual)

Fit 2nd order polynomial

Find a_2, a_1 and a_0 such that

$$y = a_1 x + a_2 x^2 + a_0$$

Suppose a0a = 3; a1a = 6; a2a = 2; (actual)
x = 1:10; dev = 0.5;

Fit 2nd order polynomial

Find a_2, a_1 and a_0 such that

$$y = a_1 x + a_2 x^2 + a_0$$

Suppose a0a = 3; a1a = 6; a2a = 2; (actual)
x = 1:10; dev = 0.5;
y = a1a∗x + a2a∗x•^2 + a0a + dev∗(rand(x)-0.5)
x = 1:10; x2 = x.^2;
X=[x;x2];
reglin(X,y)

Read/write csv files

csv $\equiv$ comma separated value

Read/write csv files

csv ≡ comma separated value

```
r = read_csv('lsquare_data.csv',ascii(9)); // read csv data into r
r = strsubst(r,',','.'); // string substitute
r = evstr(r); // convert string to numerical values
```

Read/write csv files

csv ≡ comma separated value

```
r = read_csv('lsquare_data.csv',ascii(9)); // read csv data into r
r = strsubst(r,',','.'); // string substitute
r = evstr(r); // convert string to numerical values

xval = r(1,:); //first row (time)
yval = r(2,:); //second row (displacement)
```

Displacement under gravity

Constant gravity: $g = 9.8\ m/s^2$ downwards.
Initial velocity: v_0
Initial displacement: x_0

$$x(t) = x_0 + v_0 t - \frac{1}{2} g t^2$$

Spoken tutorial?

More spoken tutorials?

Spoken tutorial?

More spoken tutorials?
For least square fit: and remaining topics:

Spoken tutorial?

More spoken tutorials?
For least square fit: and remaining topics:
Enthusiasts/volunteers required for spoken tutorials creation.

Spoken tutorial?

More spoken tutorials?

For least square fit: and remaining topics:

Enthusiasts/volunteers required for spoken tutorials creation.

Be in touch

Spoken tutorial?

More spoken tutorials?
For least square fit: and remaining topics:
Enthusiasts/volunteers required for spoken tutorials creation.
Be in touch