Numerical Methods ITGS219

Curves of Best Fit

By: Zahra A. Elashaal

5.7 Curves of Best Fit

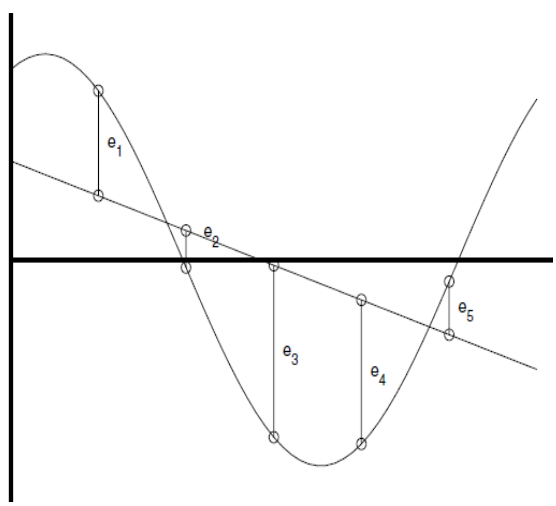
In all of the examples so far we have forced the curves to go through all the data points. We now relax that requirement.

We shall start with a straight line and optimize the coefficients used to define it. The straight line is given by $f_L(x) = ax + b$.

Let us assume that the **error** at a certain point x_j is given by $(f_L(x_j) - f_j)^2$, so the **total error** is

$$\text{Error: } e = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (f_L(x_i) - f_i)^2$$

$$= \sum_{i=1}^{N} (ax_i + b - f_i)^2 .$$



We wish to minimize this expression by choosing *a* and *b* accordingly.

Curves of Best Fit … cont.

$$e = \sum_{i=1}^{N} (ax_i + b - f_i)^2 .$$

In order to determine the actual values of a and b we **differentiate** with respect to each one and set the **result** equal to **zero**.

We need to be very careful at this point, and this is included **purely** for interest:

$$\frac{\partial e}{\partial a} = \sum_{i=1}^{N} 2x_i \left(ax_i + b - f_i\right) = 0 \quad \text{and} \quad \frac{\partial e}{\partial b} = \sum_{i=1}^{N} 2\left(ax_i + b - f_i\right) = 0.$$

These equations can be manipulated to give the simultaneous equations

$$a\sum_{i=1}^{N} x_i^2 + b\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i f_i \qquad a\sum_{i=1}^{N} x_i + bN = \sum_{i=1}^{N} f_i .$$

$$\begin{pmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & N \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} x_i f_i \\ \sum_{i=1}^{N} f_i \end{pmatrix} \Rightarrow \begin{cases} a = \dfrac{N\sum_{i=1}^{N} x_i f_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} f_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2} = \dfrac{N\sum x_i f_i - \sum x_i \sum f_i}{N\sum x_i^2 - \left(\sum x_i\right)^2} \\ b = \dfrac{\sum_{i=1}^{N} f_i}{N} - \mathrm{a}\dfrac{\sum_{i=1}^{N} x_i}{N} = \dfrac{\sum f_i}{N} - \mathrm{a}\dfrac{\sum x_i}{N} = \bar{f} - a\bar{x} \end{cases}$$

Curves of Best Fit … Example.

Example: Fit a straight line to the x and f values in the next table:

and then find the value of the function at $x=3.5$.

x_i	1.1	2	3.2	4	4.5	5.2
f_i	0.5	2.5	2	4	3.5	6.5

Solution: $f_L(x) = a\,x + b$ and $N=6$ $\quad e = \sum_{i=1}^{N} (ax_i + b - f_i)^2 .$

$$a = \frac{N\sum x_i f_i - \sum x_i \sum f_i}{N\sum x_i^2 - \left(\sum x_i\right)^2}$$

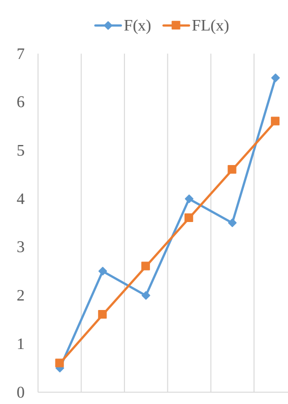
$$a = \frac{6 * 77.5 - 20 * 19}{6 * 78.74 - 400} = 1.1734$$

$$b = \frac{\sum f_i}{N} - \mathrm{a}\frac{\sum x_i}{N} = \bar{f} - a\bar{x}$$

$$b = \frac{19}{6} - 1.1734 * \frac{20}{6} = -0.7446$$

$f_L(x) = 1.1734\,x - 0.7446$

	x_i	f_i	x_i^2	$x_i f_i$
	1.1	0.5	1.21	0.55
	2	2.5	4	5
	3.2	2	10.24	6.4
	4	4	16	16
	4.5	3.5	20.25	15.75
	5.2	6.5	27.04	33.8
Σ	**20**	**19**	**78.74**	**77.5**



$f_L(3.5) = 1.1734 * 3.5 - 0.7446 = 3.3623$

Curves of Best Fit … Example.

Example: Fit a straight line to the given data regarding x as the independent variable.

x_i	1	2	3	4	5	6
y_i	1200	900	600	200	110	50

Sol. Let the straight line obtained from the given data by
$y = a\,x + b$ ………… (1)

Then the normal equations are:

$$a\sum_{i=1}^{N} x_i + bN = \sum_{i=1}^{N} f_i. \qquad a\sum_{i=1}^{N} x_i^2 + b\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i f_i$$

a Σx+ N b = Σy ………… (2)

a Σx^2 + b Σx = Σxy ………… (3)

x	y	x^2	xy
1	1200	1	1200
2	900	4	1800
3	600	9	1800
4	200	16	800
5	110	25	550
6	50	36	300
$\sum x = 21$	$\sum y = 3060$	$\sum x^2 = 91$	$\sum xy = 6450$

Putting all values in the equations (2) and (3), we get

21a + 6b = 3060
91a + 21b = 6450

Solving these equations, we get
a = −243.42 and b =1361.97

Hence the fitted equation is y = − 243.42x +1361.97. Ans.

Curves of Best Fit … Example.

Example: Fit a straight line to the x and y values shown in the table:

x_i	1	2	3	4	5	6	7
y_i	0.5	2.5	2	4	3.5	6	5.5

Try to complete the steps of this example to get the following solution:

$$\sum x_i = 28 \quad \sum y_i = 24.0 \qquad \bar{x} = \frac{28}{7} = 4$$

$$\sum x_i^2 = 140 \quad \sum x_i y_i = 119.5 \qquad \bar{y} = \frac{24}{7} = 3.428571$$

complete the Solution

$$a = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{N\sum x_i^2 - (\sum x_i)^2} = \frac{7*119.5-28*24}{7*140-28^2}$$
= 0.839285

$$b = \frac{\sum y_i}{N} - a\frac{\sum x_i}{N} = \bar{y} - a\bar{x}$$
=3.428571 - 0.839285 * 4 = 0.071431

Curves of Best Fit … cont.

the file **data.dat** contains:

x	*f(x)*
0.00	1.00000000
0.40	1.03936428
0.80	-0.06498473
1.20	-2.44823335
1.60	-4.94458639
2.00	-4.82980938

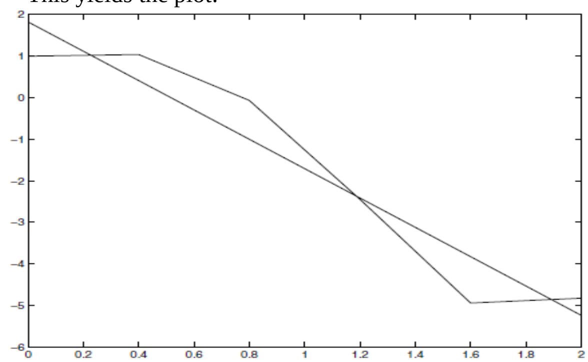
We could solve these equations by hand but we shall exploit Matlab for this purpose, so that we solve the matrix form of the equation:

$$\begin{pmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & N \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} x_i f_i \\ \sum_{i=1}^{N} f_i \end{pmatrix}$$

This is coded as:

```
load data.dat;
x = data(:,1); f=data(:,2);
N = length(x);
A = ([sum(x.^2)  sum(x);  sum(x)  N]);
rhs = ([sum(x.*f);  sum(f)]);
vect = inv(A)*rhs;
a = vect(1); b = vect(2);
fss = a*x+b;
plot(x,f,'r',x,fss,'b')
```

This yields the plot:



As you can see the curve is a reasonable approximation to the points but does not pass through many (if any) of the actual grid points. This method can be extended to assume other forms of data.

Curves of Best Fit … cont.

We could have also used the Matlab command *polyfit* which automates the previous procedure. This is called using *[p, s] = polyfit(x,y,n)* and fits a polynomial of degree *n* for *y = y(x).*

- The coefficients for the polynomial are returned in *p* and the second variable *s* is associated with the structure and is used by other commands to assess the level of the error.

We can now use the command *polyval* to determine the value of the fitted polynomial at an intermediate point.

- This can be called simply as *y = polyval(p,x)* or in a more sophisticated form as

  [y, delta] = polyval(p,x,s), where delta is in some sense **representative** of the **error** and the input argument *s* is generated by *polyfit.*

- If we know more about a function we can use **higher-order approximations**, or can use **combinations of functions**.

Curves of Best Fit … cont.

Example 5.5 Given that a set of data is of the form $y = ax + be^{-x} + c$ state how one would determine the constants a, b and c.

As with the linear example we define the sum of the squares of the errors

$$e = \sum_{i=1}^{N} \left(ax_i + be^{-x_i} + c - f_i\right)^2 \qquad \text{Error: } e = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} \left(f_L(x_i) - f_i\right)^2$$

and seek the values of the constants which minimizes this expression. Again we construct the partial derivatives and equals by zeros:

$$\frac{\partial e}{\partial a} = \sum_{i=1}^{N} 2x_i \left(ax_i + be^{-x_i} + c - f_i\right), \quad \frac{\partial e}{\partial b} = \sum_{i=1}^{N} 2e^{-x_i} \left(ax_i + be^{-x_i} + c - f_i\right), \quad \frac{\partial e}{\partial c} = \sum_{i=1}^{N} 2\left(ax_i + be^{-x_i} + c - f_i\right).$$

These equations can then be combined into a matrix form to give:

$$\begin{pmatrix} \sum x_i^2 & \sum x_i e^{-x_i} & \sum x_i \\ \sum e^{-x_i} x_i & \sum e^{-2x_i} & \sum e^{-x_i} \\ \sum x_i & \sum e^{-x_i} & \sum 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum x_i f_i \\ \sum e^{-x_i} f_i \\ \sum f_i \end{pmatrix}$$

Quiz

Write a Matlab program to fine the values of a, b and c for the equation $y = ax + be^{-x} + c$ from the next matrix's.

Note: You can load the value of x_i and $f_i(x)$ from a file or by input them directly.

$$\begin{pmatrix} \sum x_i^2 & \sum x_i e^{-x_i} & \sum x_i \\ \sum e^{-x_i} x_i & \sum e^{-2x_i} & \sum e^{-x_i} \\ \sum x_i & \sum e^{-x_i} & \sum 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum x_i f_i \\ \sum e^{-x_i} f_i \\ \sum f_i \end{pmatrix}$$

the file **data.dat** contains:

x	*f(x)*
0.00	1.00000000
0.40	1.03936428
0.80	-0.06498473
1.20	-2.44823335
1.60	-4.94458639
2.00	-4.82980938

```
load data.dat;
x = data(:,1); f=data(:,2);
%N = length(x);
A = ([ sum(x.^2)          sum(x.*exp(-x))  sum(x) ; ...
       sum(exp(-x).*x)  sum(exp(-2*x))  sum(exp(-x)); ...
       sum(x)              sum(exp(-x))       sum(1)  ]);
rhs = ([sum(x.*f);  sum(exp(-x).*f) ;  sum(f)]);
vect = inv(A)*rhs;
a = vect(1); b = vect(2); c = vect(3);
fss = a*x + b*exp(-x) + c;
plot(x,f,'r',x,fss,'b')
```

Any Question?