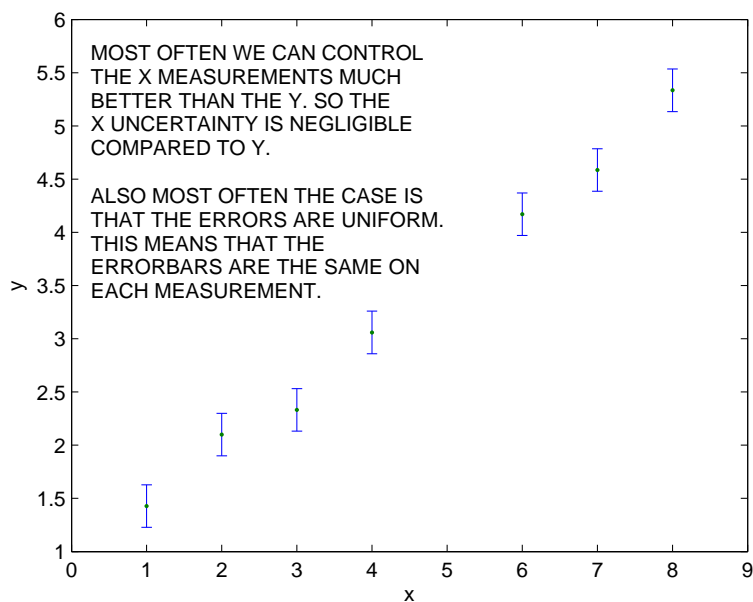
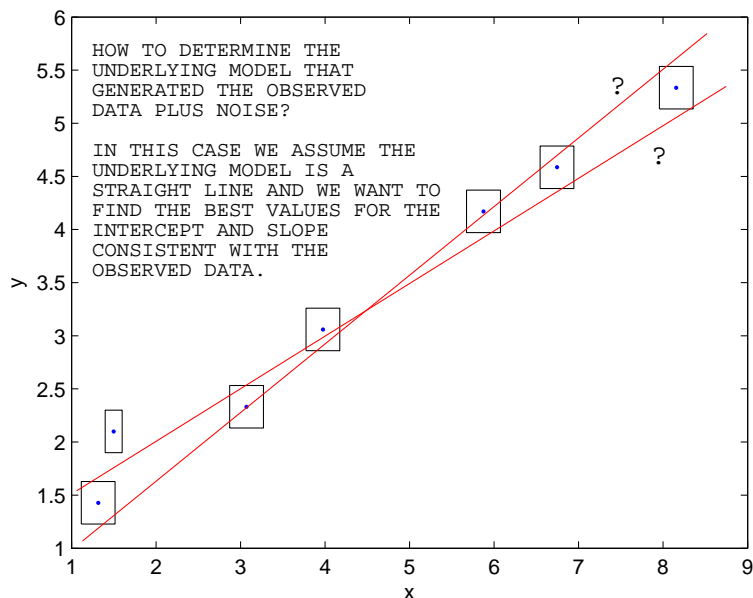



LEAST SQUARES FITTING



In the restricted case that the errors in x are negligible, the errors in y are uniform (Gaussian)  The **best** fit curve has

$$\sum_{i=0}^N \epsilon_i^2 = \text{minimum}$$

where ϵ_i are now the offset of the observed data from the data predicted by the best model. The offsets, or **data misfits** are not the same as the errors of observation, except if the model is correct.

Suppose we want to least squares fit a straight line to some data

$$y = ax + b$$

then we want to know what choices for a and b result in the smallest $\sum_{i=0}^N \epsilon_i^2$? Or, more generally

$$f(x) = a_1 f_1(x) + a_2 f_2(x) + \cdots + a_M f_M(x)$$

so there are M parameters and M known functions $f_j(x)$. Accounting for the fact there are N observations

$$f(x_i) = a_1 f_1(x_i) + a_2 f_2(x_i) + \cdots + a_M f_M(x_i) + \epsilon_i \quad i = 1 \cdots N$$

This is really a set of equations

$$\begin{pmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_M(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_M(x_2) \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ f_i(x_N) & f_2(x_N) & \cdots & f_M(x_N) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ a_M \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \cdot \\ \cdot \\ \cdot \\ f(x_N) \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_N \end{pmatrix}$$

The big matrix is M columns by N rows and usually there are more observations than parameters so $N > M$.

A shorthand for the above is

$$f_{ij} \cdot a_j = d_i + \epsilon_i$$

or better

$$F \cdot A = D + E$$

where E is the misfit of the model to the data, or the error vector

We would like to solve this equation for A such that

$$E = \sum_i^N \epsilon_i^2 \quad \text{is a minimum}$$

To find the minimum of a function we differentiate wrt a variable and set that equal to zero.

$$\frac{\partial E}{\partial a_k} = 0 \quad k = 1, 2 \dots M$$

or

$$\sum_i^N 2\epsilon_i \frac{\partial \epsilon_i}{\partial a_k} = 0 \quad k = 1 \dots M$$

but

$$\epsilon_i = \sum_{j=1}^m f_{ij} a_{jk} - d_i$$

so

$$\begin{aligned} \frac{\partial \epsilon_i}{\partial a_k} &= \sum_{j=1}^m f_{ij} \frac{\partial a_j}{\partial a_k} - \frac{\partial d_i}{\partial a_k} \\ &= f_{ik} \quad \text{because} \quad \frac{\partial a_j}{\partial a_k} = \delta_{jk} \quad \& \quad \frac{\partial d_i}{\partial a_k} = 0 \end{aligned}$$

so the condition for a minimum becomes

$$\sum_{i=1}^N \epsilon_i f_{ik} = 0 \quad k = 1 \dots M$$

or, substituting for ϵ

$$\epsilon_i = (f_{ij}a_j - d_i)$$

or

$$(F \cdot A - D) \cdot F = 0$$

$$F \cdot A \cdot F = D \cdot F$$

multiplying through by the transpose of F

$$F^T F \cdot A \cdot F = F^T \cdot D \cdot F$$

$$A \cdot F = (F^T F)^{-1} F^T D \cdot F$$

and so

$$A = (F^T F)^{-1} F^T D$$

is the choice for A that minimizes the sum of squares of the data misfits.

matlab has a shorthand for the above

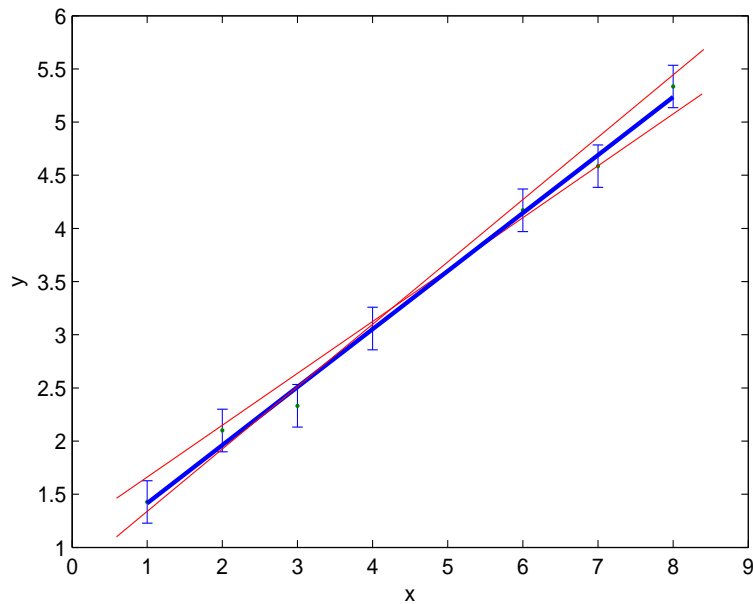
$$A = (F^T F)^{-1} F^T D = F \backslash D$$

as long as the dimensions are consistent, matlab will interpret the backslash in this way.

The data as predicted by the best model are then

$$d_{pre} = \cancel{Am} FA$$

Going back to the original figure, the best line is the bold line. Some other possibilities, that fit the observed data but not with as small a misfit as the best model are also shown.



Now we can calculate the misfit

$$\begin{aligned}
 E &= F \cdot A - D \\
 &= (F(F^T F)^{-1} F^T - I) \cdot D
 \end{aligned}$$

If the correct model has been chosen, that is, it accounts for all the variance in the data except that due to the errors of observation, then the data misfit will be the same as the errors of observation. ~~Since errors of observation are often Gaussian, (Gauss invented least squares)~~ least squares fitting implicitly assumes that the misfit is Gaussian, although the method will still produce reasonable results even if the misfit is not Gaussian. *More precisely, only with Gaussian errors, the Least-Squares inverse gives an unbiased solution*

This is important in the potential fields context because when we separate regional and residual by least squares we are assuming that the regional is the model and the misfit is the residual. Does it make any sense at all to expect that the residual will be Gaussian?

EXAMPLE 1

Fit a (1D) parabola to data

$$a_1 + a_2x + a_3x^2 = f$$

and so if we had data

$$\begin{pmatrix} x = \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \begin{pmatrix} d = \\ 1 \\ 6 \\ 17 \\ 32 \\ 58 \end{pmatrix}$$

the matrix expression would be

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 6 \\ 17 \\ 32 \\ 58 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{pmatrix}$$

and

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = (F^T F)^{-1} F^T \begin{pmatrix} 1 \\ 6 \\ 17 \\ 32 \\ 58 \end{pmatrix} = \begin{pmatrix} 0.024 \\ 0.078 \\ 0.274 \end{pmatrix}$$

EXAMPLE 2

Fitting a single sine wave with known frequency but unknown amplitude and phase.

Fitting a signal in noise

$$\begin{pmatrix} \sin\omega_o t_1 & \cos\omega_o t_1 \\ \sin\omega_o t_2 & \cos\omega_o t_2 \\ \vdots & \vdots \\ \sin\omega_o t_N & \cos\omega_o t_N \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} d(t_1) \\ d(t_2) \\ \vdots \\ d(t_N) \end{pmatrix} + \begin{pmatrix} \epsilon \end{pmatrix}$$

and so

$$\begin{pmatrix} a \\ b \end{pmatrix} = (F^T F)^{-1} F^T d$$

and the amplitude is

$$amp = (a^2 + b^2)^{1/2}$$

and the phase of the sine wave is

$$\phi = \text{Tan}^{-1}(b/a)$$

Note we could construct an entire Fourier transform in this way. The data would not have to be equispaced as with the DFT and FFT.

EXAMPLE 3

FITTING A PLANE IN TWO DIMENSIONS

This is the simplest regional residual separation. The equation of a plane is

$$ax + by + c = PLANE(x, y)$$

so the set up is

$$\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \cdot & \cdot & \\ x_N & y_N & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \cdot \\ d_N \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \cdot \\ \epsilon_N \end{pmatrix}$$

and this can be generalized to any polynomial surface.

In all these examples, the data were a linear function of the parameters. Suppose in the sinusoid fitting problem the frequency was unknown and we wanted to least squares fit for it. The data depend non-linearly on the frequency so we could not solve this in the above way. Non-linear least squares requires special tools that I don't have time to discuss here.

The equation for the magnetic anomaly of even the simplest structure - a dipole - is non-linear in the parameters as is the gravity anomaly of a sill, so there are many problems we will not be able to handle with linear least squares.

CONVOLUTION

Many of the operations we do with Fourier transforms can also be done with convolution filters. A FILTER operates on a data set by CONVOLVING the elements of the filter with the elements of the data.

Suppose you had a data set that consisted of a single spike at time T .

and a filter that responds to a spike by delaying it by $2\Delta T$ the issuing a single spike, waiting a further $2\Delta T$ issuing a second spike with half the amplitude of the first etc. This is like listening at a repeated echo between two cliffs. The filter coefficients are the amplitudes in response to a spike and the filter delays are the waits between spikes. You actually only know the filter coefficients, the delay is in arbitrary units that only become specific when the filter is applied to data sampled at some interval.

Suppose we had a filter whose coefficients are $f_1 f_2$ etc, and the data is just one spike of amplitude a .

$$\begin{array}{ccccccc}
 =af_o & af_1 & af_2 & \cdots af_n & \text{output} & & \\
 =T_o & T_o + \Delta T & T_o + 2\Delta T & \cdots T_o + n\Delta T & \text{time} & &
 \end{array}$$

Spatial filters used in gravity/mag usually extend in BOTH directions of the coordinate axes.

Reverberations in a water layer (downward going pulses only) could be described by a filter whose coefficients are

$$1 \quad -e \quad +e^2 \quad -e^3 \dots$$

How to handle an input that is not a single spike? If data are sampled at discrete and equally spaced times then each sample is a spike and we can convolve each spike with the filter accounting for delays.

For example, suppose we had data

$$1 \quad 1/2 \quad 2$$

at time T_o , $T_o + \Delta T$, $T_o + 2\Delta T$ and filter coefficients

$$1 \quad 1/2 \quad 1/4$$

The response to the first data sample is

$$1 \quad 1/2 \quad 1/4$$

at times T_o , $T_o + \Delta T$, $T_o + 2\Delta T$ (the amplitude of the first data point is 1 and there is no delay for the first filter coefficient).

The response to the second data sample is

$$11/2 \quad 3/4 \quad 3/8$$

starting at $T_o + \Delta T$ and incrementing by ΔT .

The response to the third data sample is

$$2 \quad 1 \quad 1/2$$

starting at $T_o + 2\Delta T$. Adding all these up with allowance of delays, we have

$$1 \quad 2 \quad 3 \quad 13/8 \quad 1/2$$

starting at T_o of course.

Here is what we did

$$\begin{array}{cccc} \text{reflected order} & d_2 & d_1 & d_o \\ & & & f_o \quad f_1 \quad f_2 \end{array}$$

So the response is $f_o d_o$ at T_o

The response at $T_o + \delta T$ is

$$\begin{array}{cccc} \text{reflected order} & d_2 & d_1 & d_o \\ & & & f_o \quad f_1 \quad f_2 \end{array}$$

So the response is $f_o d_1 + f_1 d_o$

The response at $T_o + 2\Delta T$ is

$$\begin{array}{rcccc} \text{reflected order} & d_2 & d_1 & d_o \\ & f_o & f_1 & f_2 \end{array}$$

So the response is $f_o d_2 + f_1 d_1 + f_2 d_o$

The response at $T_o + 3\Delta T$ is

$$\begin{array}{rcccc} \text{reflected order} & d_2 & d_1 & d_o \\ & f_o & f_1 & f_2 \end{array}$$

So the response is $f_1 d_2 + f_2 d_1$

The response at $T_o + 4\Delta T$ is

$$\begin{array}{rcccc} \text{reflected order} & d_2 & d_1 & d_o \\ & f_o & f_1 & f_2 \end{array}$$

So the response is $f_2 d_2$

and response at $T_o + 5\Delta T$ is

$$\begin{array}{rcccc} \text{reflected order} & d_2 & d_1 & d_o \\ & f_o & f_1 & f_2 \end{array}$$

that is, there is no overlap, so no response.

A more concise notation is to write $y = f * d$ the output of f convolved with d .

$$y_t = \sum_{s=0}^m f_s d_{t-s} \quad \text{it is convenient to view the range of 's' as infinite.}$$

Where the minus sign reflects the reversal we did above.

or, for continuous data

$$f * g = \int_{-\infty}^{\infty} f(x)g(t-x)dx$$

There are some useful properties of convolutions

$$f * g = g * f \quad \text{commutative}$$

$$f * (g + h) = f * g + f * h \quad \text{distributive}$$

$$f * (g * h) = (f * g) * h \quad \text{associative}$$

CONVOLUTION IN THE OBSERVATION DOMAIN

IS EQUIVALENT TO MULTIPLICATION IN THE FREQUENCY DOMAIN

AND

CONVOLUTION IN THE FREQUENCY DOMAIN IS EQUIVALENT TO MUL-

TIPLICATION IN THE OBSERVATION DOMAIN

As a result of the convolution theorem we know that observing for a finite amount of time broadens the spectrum of a peak in the frequency domain.

Derivatives of convolutions are important

CONVOLUTIONS IN 2D

$$h(x, y) = \int \int f(s, t)g(x - s, y - t)dsdt$$

or

$$h_{n,m} = \sum_0^{k+r} \sum_0^{l+s} f_{ij}g_{n-i,m-j}$$

A 2-D filter might look like

$$\begin{pmatrix} f_{-11} & f_{01} & f_{11} \\ f_{-10} & f_{00} & f_{10} \\ f_{-1-1} & f_{0-1} & f_{1-1} \end{pmatrix}$$