Basic tutorial on Curve Fitting

M. M. Bandi[1]

[1]*Nonlinear and Non-equilibrium Physics Unit, OIST Graduate University, 1919-1 Tancha, Onna-son, Okinawa, 904-0495 Japan*

(Dated: February 13, 2023)

These notes present the basics of Curve fitting. You will find these methods useful no matter what branch of science you work in because we deal with data in all areas of science.

INTRODUCTION

The process of constructing an approximate curve $y = f(x)$, which fit best to a given discrete set of data points (x_i, y_i), $i = 1, 2, 3, ..., n$ is called curve fitting. Curve fitting and interpolation are closely associated procedures. In interpolation, the fitted function should pass through all given data points, whereas curve fitting methodologically fits a unique curve to the data points, which may or may not lie on the fitted curve. The difference between interpolation and curve fitting; while attempting to fit a linear function is illustrated in the adjoining figure reffig1.

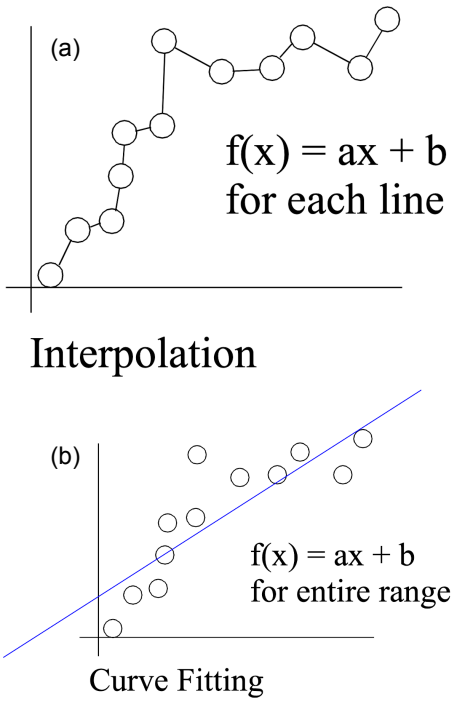

FIG. 1: (a) Interpolation and (b) Curve fitting of an arbitrary data set

Interpolation: The method connects all the data points. If the data is reliable, i.e. we know that the values are error-free in all respects, we can plot it and connect the dots. This is a piece-wise linear interpolation, and has limited use as a general function $f(x)$. Since its really a group of small $f(x)$s, connecting one point to the next doesn't really work well for data that has built in random errors (scatter).

Curve fitting Captures the trend in the data by assigning a single function across the entire range. The example in fig. 1 uses a straight line function. A straight line is described generically by $f(x) = ax + b$. Our **goal** is to identify the coefficients "a" and "b" such that $f(x)$ "fits" the data well. But what does it mean to require

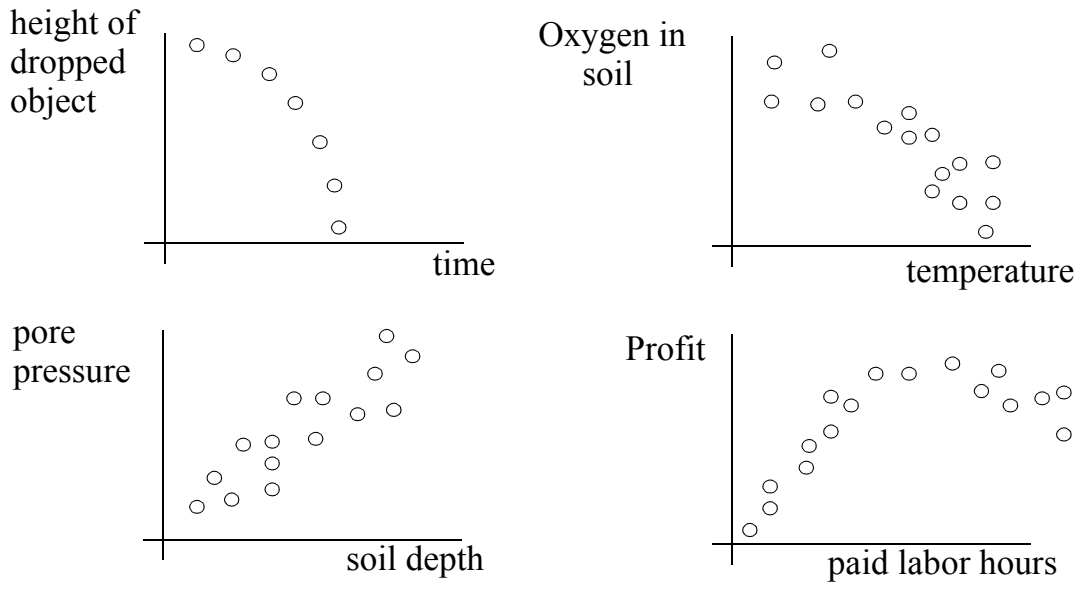

FIG. 2: Examples of data sets that can be fit to an as yet unknown function $f(x)$.

that $f(x)$ fits the data well? This is a very subjective remark and lacks the quantitative rigor we expect to bring to a principled approach to curve fitting. We will sharpen the meaning of this term in due course and cast it in quantitative terms shortly.

Let us start by considering some other examples of arbitrary data sets that we can fit a function to in fig. 2. Presumably, each of these data sets was an experimental measurement. Is a straight line suitable for each of these cases? Certainly Not. But we are not stuck with just straight line fits. We will start with straight lines and then expand the concept to work our way towards an extended class of functional forms.

In fig. 2, we have some idea of the expected functional form for say, the height of a dropped object versus the time as shown in fig 2a. Elementary mechanics teaches us this curve follows a parabolic form $h = -\frac{1}{2}gt^2$ where the negative sign denotes we have chosen the height as positive upwards and negative going downwards, $g = 9.8\text{m/s2}$ is the acceleration due to gravity, h is the height from which the object is dropped and t is the time for the said object to hit the ground. But for other measurements (fig 2b-d), we may or may not have a prior idea of what functional form to expect for the data, although by looking at the plots themselves we can form a subjective opinion of what that functional form might look like. For instance, the pore pressure seems to increase linearly with soil depth, where as the oxygen content in soil seems to decrease approximately linearly with temperature, I say approximately because from the scatter in the data, its difficult to tell whether its linear or not, which doesn't seem to be the case for pore pressure. Finally for profit against paid labor hours

its clearly nonlinear, the profits seem to increase linearly early on and then either saturate or even decrease marginally when the paid labor hours increase beyond a certain point. Whereas this does intuitively make sense – a company's profits cannot increase forever with the number of hours they pay a laborer to work because the laborer's physical exertion takes a toll after some hours, there are human limits to how long a person can continuously work.

LINEAR CURVE FITTING (LINEAR REGRESSION)

Given the general form of a straight line

$$f(x) = ax + b \tag{1}$$

how can we pick the coefficients "a" and "b" that best fits the line to the data? First question: What makes a particular straight line "good" fit? You see, we run smack into the question we asked above about the subjective nature of the word "well." Let us consider the two plots in fig. 3

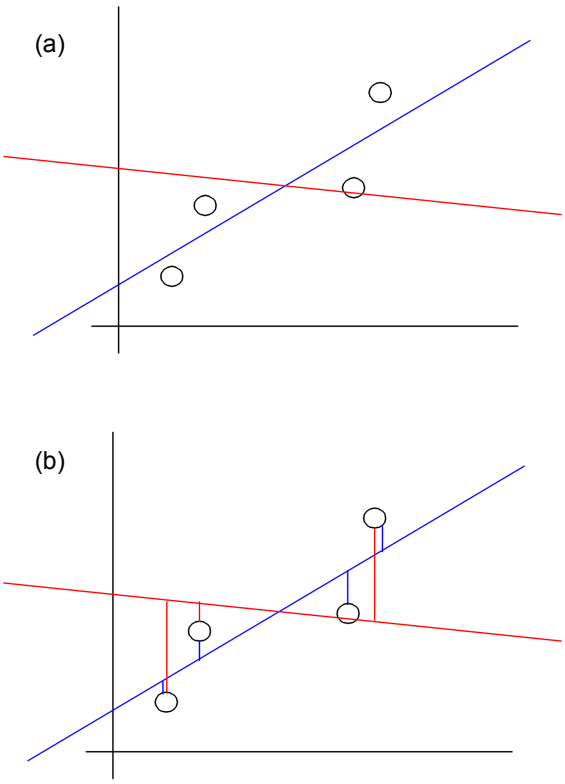

FIG. 3: (a) Two potential linear fits (red and blue solid lines) around a scattered set of data points (black circles) and (b) the distance from each data point to each of the linear fits

In fig. 3a we have two solid lines, one blue and one red, which provide putative linear fits to the scattered data points (black circles). Intuitively, it seems as though the blue line fits the data better than the red line. Can we set this intuition of ours on a quantitative basis?

1) Consider the distance between the data and points on the line as we do in fig. 3b, the vertical lines from the fit to each data point.

2) Add up the length of all the red and blue vertical lines.

3) The sum of these vertical lines is an expression of the "error" between data and fitted line. The shorter the distance between the black circles and fit lines, the smaller the value of the sum.

4) The one fit line that provides a minimum error is then the "best" straight line fit.

But there's a hitch with our procedure. A data point that is above or below a fit line has a ver-

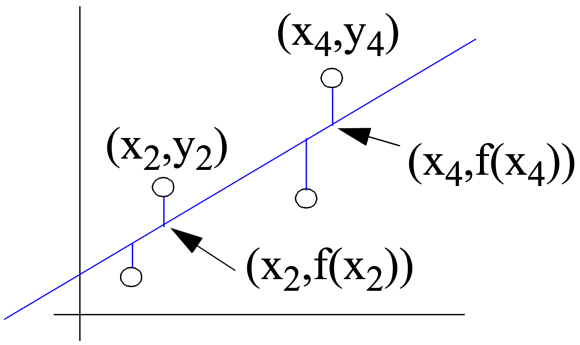

FIG. 4:

tical line denoting the error, but if we adopt an *ad hoc* principle that the vertical line above fit line is positive and below is negative, the sum of vertical lines can again be small, or even sum to zero. But the error remains even though our method would suggest its zero. Let us improve on our first attempt.

Quantifying error in a curve fit

Assumptions:

1) Positive or negative error (data point above or below the line) has the same sign.

2) Weight greater errors more heavily.

Both requirements can be achieved by **squaring** the distance.

1) Denote the data point by values (x, y).

2) Denote points on the fitted line as $(x, f(x))$.

3) Then we define the error as:

$$\epsilon = \sum (d_i)^2 = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 + f(x_4))^2 \quad (2)$$

4) Our fit is a straight line, so now substitute $f(x) = ax + b$

$$\epsilon = \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} (y_1 - (ax_i + b))^2 \quad (3)$$

where N is the total number of data points.

5) Using this method, we notice the "best" line has **minimum error** between the fitted line and the data points.

This is known as the least squares approach, since we minimize the square of the error:

$$\text{Minimize } \epsilon = \sum_{i=1}^{N} (y_1 - (ax_i + b))^2 \quad (4)$$

Its calculus time now, to find the minimum of a function. We know a derivative describes a slope and zero slope is an extremum. To assure ourselves the extremum is in fact a minimum, we must calculate the second order derivative, which, if positive, tells us its a minimum. We won't worry with the second-order derivative here for sake of simplicity and simply assume the existence of an extremum (1st order derivative) as being a minimum. In truth, the maximum is often, but not always, unbounded in these error estimations.

Let us take the derivative of the error with respect to a and b, set each to zero

$$\begin{aligned} \frac{\partial \epsilon}{\partial a} &= -2 \sum_{i=1}^{N} x_i (y_i - a x_i - b) = 0 \\ \frac{\partial \epsilon}{\partial b} &= -2 \sum_{i=1}^{N} (y_i - a x_i - b) = 0 \end{aligned} \quad (5)$$

Solve for the a and b such that the pair of equations in Eq. 5 both equal zero.

TABLE I: Least squares fit sample dataset.

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	1.5	3.0	4.5	6.0	7.5

Re-write these two equations as

$$\begin{aligned} a\sum x_i^2 + b\sum x_i &= \sum(x_i y_i) \\ a\sum x_i + b \times N &= \sum y_i \end{aligned} \tag{6}$$

put these in matrix form

$$\begin{bmatrix} N & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum(x_i y_i) \end{bmatrix} \tag{7}$$

What is unknown? We have the data points (x_i, y_i) for $i = 1, ...N$, so we have al lthe summation terms in the matrix. So the unknowns are a and b. Good news! We already know how to solve this problem. Remember Gaussian elimination or Row reduction in matrix algebra?

$$A = \begin{bmatrix} N & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}, X = \begin{bmatrix} b \\ a \end{bmatrix}, B = \begin{bmatrix} \sum y_i \\ \sum(x_i y_i) \end{bmatrix} \tag{8}$$

so $AX = B$. By inverting this matrix, the coefficients a and b are solved for, i.e. $X = A^{-1}B$. Please note that A, B and X are not the same as a, b and x.

Let us test this with an example using the dataset presented in Table I First we find values for all the summation terms $N = 6$. $\sum x_i = 7.4, \sum y_i = 22.5, \sum x_i^2 = 13.75, \sum x_i y_i = 41.25$. Now plugging into the matrix form gives us:

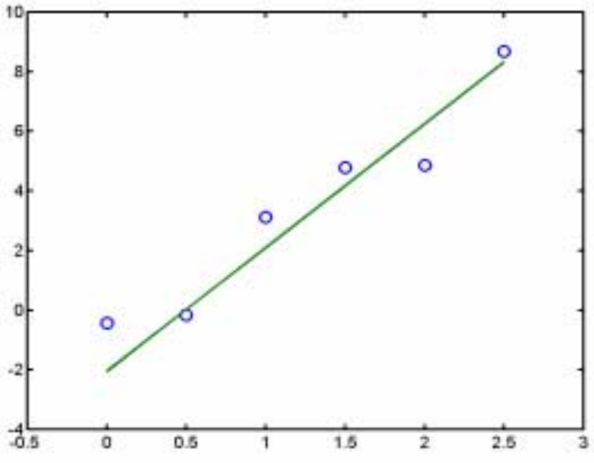

FIG. 5:

$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix} \tag{9}$$

Please note we are using $\sum x_i^2$ and not $(\sum x_i)^2$.

$$\begin{bmatrix} b \\ a \end{bmatrix} = \mathrm{I}nv \begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} 22.5 \\ 41.25 \end{bmatrix} \tag{10}$$

or use Gaussian elimination. The solution is $\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \implies f(x) = 3x + 0.$

This fits the data exactly, i.e. the error is zero. Usually this is not the outcome, instead we normally have data that does not exactly fit a straight line. Here's an example with some "noisy" data.

$x = [0 \;\; 0.5 \;\; 1 \;\; 1.5 \;\; 2 \;\; 2.5]$,

$y = [-0.4326 \;\; 0.1656 \;\; 3.1253 \;\; 4.7877 \;\; 4.8535 \;\; 8.6909]$

$$\begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix},$$

$$\begin{bmatrix} b \\ a \end{bmatrix} = \mathrm{I}nv \begin{bmatrix} 6 & 7.5 \\ 7.5 & 13.75 \end{bmatrix} \begin{bmatrix} 20.8593 \\ 41.6584 \end{bmatrix}$$

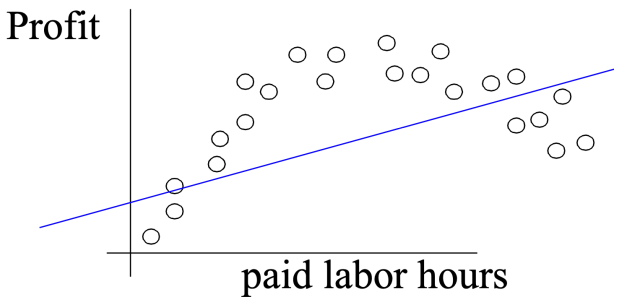

FIG. 6: Case of linear fit poorly describing data.

$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} -0.975 \\ 3.561 \end{bmatrix}$ so our fit is $f(x) = 3.561x - 0.975$. The plot is shown in fig. 5.

Thus far, we have looked at data sets that do fit a straight line. What do we do when a straight line is not suitable for a data set as shown in fig. 6? The straight line linear fit will not predict the diminishing returns in profits with increasing paid labor hours shown by the data.

We started the linear curve fit by choosing a generic form of the straight line $f(x) = ax + b$. This is just one kind of a function. There are an infinite number of generic forms we could choose from for almost any shape we want. Let us start with a simple extension of the linear regression concept and recall the examples of sampled data from fig. 2.

Is a straight line suitable for each of these cases? We already concluded it is not. Top left and bottom right don't look linear in trend, so why fit a straight line? There is no logical reason to do so. Let us then consider other options. There are a lot of functions with lots of different shapes that depend on coefficients. We can choose a form based on experience and trial & error. Let us develop a few options for nonlinear curve fitting. We will start with a simple extension to linear regression by moving to higher order polynomials.

POLYNOMIAL CURVE FITTING

Consider the general form for a polynomial of order j

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + ... a_j x^j + \sum_{k=1}^{j} a_k x^k \tag{11}$$

Just as was the case for linear regression, we ask how can we pick the coefficients that best fits the curve to the data? We can use the same idea: The curve fit that gives minimum error between data y and the fit $f(x)$ is the "best" fit.

Quantify the error for the two second order curves shown in fig. 7.

(1) Add up the length of all the red and blue vertical lines.

(2) Pick the curve with minimum total error.

Error - Least squares approach

The general expression for any error using the least squares approach is

$$\epsilon = \sum (d_i)^2 = (y_1 - f(x))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 - \tag{12}$$

where we want to minimize this error. Now substitute the form of our Eq. 10 into the general

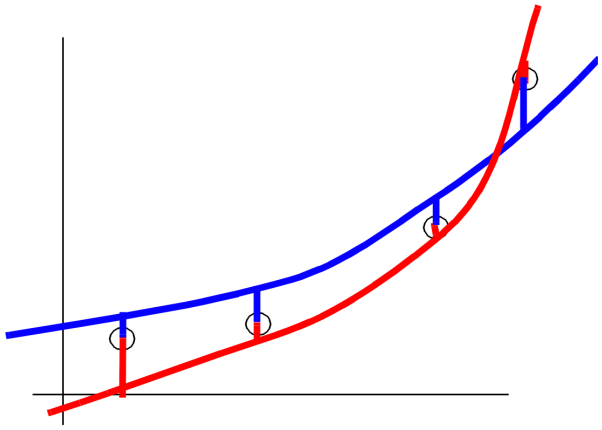

FIG. 7:

least squares error Eq. 12.

$$\epsilon = \sum_{i=1}^{N} \left(y_i - \left(a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + ... a_j x_i^j \right) \right) \tag{13}$$

where N is the number of data points given, i is the current data point being summer and j is the polynomial order. Re-writing Eq. 13

$$\epsilon = \sum_{i=1}^{N} \left(y_i - \left(a_0 - \sum_{k=1}^{j} a_k x^k \right) \right) \tag{14}$$

find the best line = minimize the error (squared distance) between line and data points. Find the set of coefficients a_k, A_0 so we can minimize Eq. 14.

To minimize Eq. 14, we take the derivative with respect to each coefficient $a_0, a_k \quad k = 1, ..., j$ set each to zero.

$\frac{\partial \epsilon}{\partial a_0} = -2 \sum_{i=1}^{N} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) = 0$

$\frac{\partial \epsilon}{\partial a_1} = -2 \sum_{i=1}^{N} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x = 0$

$\frac{\partial \epsilon}{\partial a_1} = -2 \sum_{i=1}^{N} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x^2 = 0$

.

.

$\frac{\partial \epsilon}{\partial a_j} = -2 \sum_{i=1}^{N} \left(y_i - \left(a_0 + \sum_{k=1}^{j} a_k x^k \right) \right) x^j = 0$

re-writing these $j+1$ equations, and putting them in matrix form yields

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & ... & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & ... & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & ... & \sum x_i^{j+2} \\ : & : & : & : & : \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & ... & \sum x_i^{j+j} \end{bmatrix}$$

where all summations above are over $i = 1, ..., N$. What is unknown? We have the data points (x_i, y_i) for $i = 1, ..., N$. We want $a_0, a_k \quad k = 1, ..., j$. But we already know how to solve this problem.

$$A = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & ... & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & ... & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & ... & \sum x_i^{j+2} \\ : & : & : & : & : \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & ... & \sum x_i^{j+j} \end{bmatrix},$$

$$X = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ : \\ a_j \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum (x_i^2 y_i) \\ : \\ \sum (x_i^j y_i) \end{bmatrix},$$

where all summations are over $i = 1, ..., N$ data points.

Please note that no matter what the order j, we always get equations LINEAR with respect to the coefficients. This means we can use the previous solution method $AX = B$ which we then invert to solve the coefficients $X = A^{-1}B$.

Example 1

Fit a second order polynomial to the data presented in Table II.

TABLE II: Least squares fit sample dataset.

i	1	2	3	4	5	6
x	0	0.5	1.0	1.5	2.0	2.5
y	0	0.25	1.0	2.25	4.0	6.25

Since the order is 2 ($j = 2$), the matrix form to solve is

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}.$$

Now plug in the data from Table II. The answers we expect for the coefficients after looking at the data are: $\sum x_i = 7.5$, $\sum y_i = 13.75$, $\sum x_i^2 13.75$, $\sum x_i y_i = 28.125$, $\sum x_i^3 = 28.125$, $\sum x + i^2 y_i = 61.1875$, $\sum x_i^4 = 61.1875$.

$$\begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 13.75 \\ 28.125 \\ 61.1875 \end{bmatrix}$$ or you could use Gaussian elimination to obtain the solution to the coefficients.

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \implies f(x) = 0 + 0x + 1x^2.$$

This fits the data exactly, i.e. $f(x) = y$ since $y = x^2$.

Example 2

Now let us try some noisy data.

$x = [0 \quad 0.5 \quad 1 \quad 1.5 \quad 2 \quad 2.5]$, $y = [0.0674 \quad 0.9156 \quad 1.6253 \quad 3.0377 \quad 3.3535 \quad 7.9409]$.

The resulting system to solve is

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \mathrm{Inv} \begin{bmatrix} 6 & 7.5 & 13.75 \\ 7.5 & 13.75 & 28.125 \\ 13.75 & 28.125 & 61.1875 \end{bmatrix} \begin{bmatrix} 15.1093 \\ 32.2834 \\ 71.276 \end{bmatrix}.$$

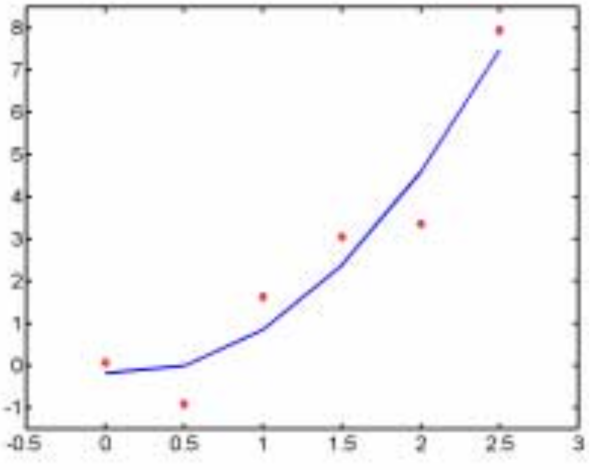

FIG. 8:

So our fitted second-order function is $f(x) = -0.1812 - 0.3221x + 1.3537x^2$.

Example 3: Data with three different fits.

In this example we are not sure which order will fit the data well, so we try three different polynomial orders. Note that linear regression or first order curve fitting is just the general polynomial form we just discussed, where we use $j = 1$. The results are presented in fig. 9, and we notice the second and sixth order fits look similar, but the sixth order fit has a 'squiggle' in it. We may not want that. This brings us to the question of overfitting or underfitting a curve.

OVERFIT/UNDERFIT

You might naively assume the higher the fit order, the more accurate the fit. This is not so. Indeed, as we saw in the third example in the previous section, the second order polynomial

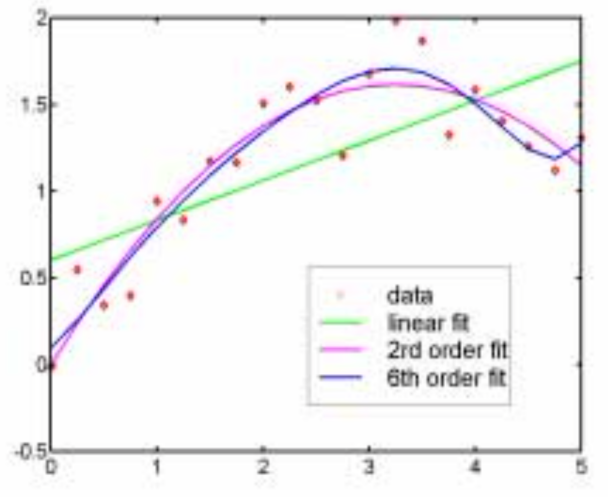

FIG. 9:

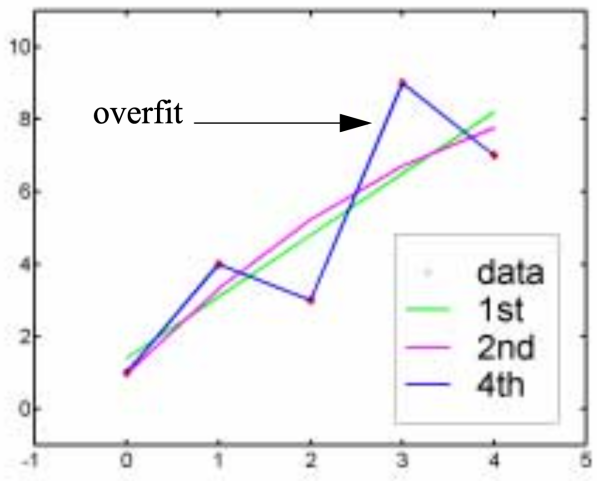

FIG. 10:

gave a better fit than the sixth order polynomial. Let us explore this a bit further to better understand the nuance here.

Overfit:

Overdoing the requirement for the fit to 'match' the data trend (order too hight). Polynomials become more 'squiggly' as their order increases. A 'squiggly' appearance comes from inflections in the function.

Consideration 1:
third order: 1 inflection point.
fourth order: 2 inflection points.
nth order: $n-2$ inflection points.

Consideration 2:
2 data points: linear touches each point.
3 data points: second order touches each point.
n data points: $(n-1)$ order polynomial touches each point.

So picking an order too high will always overfit the data, see for example the data in fig. 10.. As a general rule, pick a polynomial form at least several orders lower than the number of data points. Start with linear and add order until trends are matched.

Underfit

This occurs if the order is too low to capture obvious trends in the data, as we saw in fog. 6 where a linear fit clearly does not capture trends in the data.

As a general rule, view the data first, then select an order that reflects the inflections, etc. For the example in fig. 6, the data trend is obviously nonlinear, so the order of the polynomial fit has to be greater than 1. Secondly, we observe no obvious inflection points, so order less than 3 is recommended. Ergo, we ought to use

a second order polynomial fit.

SUMMARY

Let us conclude here, but this does not mean you've learned everything there is to learn about curve fitting. You've just scratched the surface. Let me leave you with an example. Will a polynomial of any order necessarily fit any set of data? The answer is unambiguously, No. Many phenomena do not follow a polynomial form. They may be, for example, exponential, or stretched exponential etc. For such cases you still need nonlinear curve fitting strategies but not polynomial fit. You should exercise your judgment and apply an appropriate curve fitting strategy. A simple way to get started is to apply the following consideration:

Plot the data in linear-linear, log-linear, linear-log, and log-log forms.

1) linear-linear plot denotes both X and Y axes are plotted in linear scale: a linear trend becomes obvious from this plot. If not linear, then count the inflection points in the data trend and weight against the total number of data points to determine whether a polynomial fit suffices.

2) Log-linear plot denotes Y axis in log scale and X axis in linear scale: An exponential function shows as a straight line in this plot.

3) Log-log plot denotes both X and Y axes in logarithmic scale: A power law function shows a straight line in this plot.

4) Linear-log plot denotes X axis in logarithmic and Y axis in linear scale: A logarithmic function shows a straight line in this plot.

Finally, many a time we are interested in the exponents of functions we fit the data to. For instance, if your data exhibits power-law behavior, of the form $f(x) = Ax^{\alpha}$, plotting it in log-log scale is tantamount to saying, you took a logarithm on both sides $\log [f(x) = \log A + \alpha \log x$. Notice that this looks similar to the equation of a line $y = mx + c$. Just as we determine the slope m by taking a derivative, we take a log derivative of the power-law $\frac{d\log [f(x)]}{d\log x} = \alpha$. But remember, derivatives are inherently noisy, so there are in fact a range of strategies to determine the exponent of a power-law. Similar rules apply for stretched exponentials etc. So, you can see there's plenty more to be learned.