Fitting curves to data

Fitting curves to data is one of the most common things we can do when analysing our data. There are all manner of ways in which various curves can be fitted, leading to the question of which is the “best”, under a variety of circumstances. Curves are fit to data all the time in science, especially using linear regression. However, as we’ll see, it’s not always as straightforward as it seems.

By the end of this chapter, you should know how to use Matlab to fit basic curves, how selecting a model depends on the confidence you have in your data, some sources of common error and some of the potential pitfalls associated with computers fitting data blindly.

8.1 Model fitting vs. interpolation

When analysing data, we can use information that data implies to formulate mathematical models. These models rely on assumptions about the data, or about the data we have not collected. We may encounter situations in which there are different assumptions leading to different models. For example, as an influenza pandemic moves through a population, we could make assumptions about the heterogeneous mixing of the population being proportional to the local population density, or the urban vs. rural environment, or we could ignore the heterogeneity of the population altogether.

We may be faced with using collected data to determine unknown parameters in our model in a way that selects the curve from each model that “best fits” the data and then choose whichever resultant model is most appropriate for the particular situation under investigation.

A different case arises when the problem is so complex as to prevent the formulation of a model explaining the situation. For instance, a three-dimensional model for the spread of measles might involve partial differential equations for the movement of infectious droplets in three spatial dimensions, plus one temporal dimension. This will not only be enormously complicated, but the
equations may not even be solvable, so there is little hope for constructing a master model that can be solved and analysed analytically. Or there may be so many variables that one would not even attempt to construct an explicit model.

In such cases, experiments may have to be conducted to investigate the behaviour of the system. Then the experimental data can be used to predict the outcome, but only within the range of the data points collected.

The preceding discussion identifies three possible tasks when analysing a collection of data points:

1. Fitting a selected model type or types to the data. For example, applying a line of best fit to known data points.
2. Choosing the most appropriate model from competing types that have been fitted. For example, we may need to determine whether the best-fitting exponential model is a better model than the best-fitting least-squares model.
3. Making predictions from the collected data. This may involve interpolation (predicting in between known data points) or extrapolation (predicting a point outside the range of collected data).

In Task 1, the precise meaning of “best” model must be identified and the resulting mathematical problem resolved. In Task 2, a criterion is needed for computing models of different types. In Task 3, criteria must be established for determining how to make predictions in between the observed data points or outside the ranges of what we know.

In Tasks 1 and 2, a relationship of a particular type is strongly suspected and the modeller is willing to accept some deviation between the model and the collected data points in order to have a model that satisfactorily explains the situation under investigation.

However, in the third task, when interpolating, the modeller is strongly guided by the data that have been carefully collected and analysed and a curve is sought that captures the trend of the data in order to predict in between the data points. Less reliably, we may want to predict outside the range of known data points. However, as we’ll see in the lab, curves of best fit are very good for interpolation, but not nearly so good for extrapolation, so we have to be careful.

In all situations the modeller may ultimately want to make predictions from the model. However, the modeller tends to emphasize the proposed models over the data when model fitting, whereas she places greater confidence in the collected data when interpolating and attaches less significance to the form of the model.

For example, suppose we have data from the 1918 influenza pandemic in Philadelphia shown in Figure 8.1. The x axis represents time in days and the y axis represents the number of fatal cases. To make predictions based solely upon this data, we could use a technique such as spline interpolation (which we will study in the next chapter) in order to pass a smooth curve through the
points (see Figure 8.2). In this case, the interpolating curve passes through the data points and captures the trend of the behaviour over the range of observations.

**Fig. 8.1.** Data from the 1918 influenza pandemic.

**Fig. 8.2.** Interpolating the data using a smooth curve.
However, we know that data isn’t always perfect, so a curve that passes precisely through every data point may actually be less useful than a curve that misses them all, but captures the ‘trend’.

Suppose that in studying the data the modeller makes assumptions leading to the expectation of a quadratic model, or parabola, of the form \( y = C_1 x^2 + C_2 x + C_3 \). In this case, the data of Figure 8.1 would be used to determine the arbitrary constants \( C_1, C_2 \) and \( C_3 \) in order to select the “best” parabola. See Figure 8.3. The fact that the parabola may deviate from some or all of the data points would be of no concern. Outside the range of data points, the curves may vary significantly; e.g. in the vicinity of \( x_5 \), the predictions made by the curves in Figures 8.2 and 8.3 are quite different.

![Fig. 8.3. Fitting a parabola to the data points.](image)

Of course, we may find it necessary to both fit a model and to interpolate in the same problem. The best-fitting model of a given type may prove to be unwieldy or even impossible for subsequent analysis involving operations like integration or differentiation. In such situations, the model may have to be replaced with an interpolating curve (such as a polynomial) that is more readily differentiated or integrated.

For example, a step function used to model the sudden onset of a pandemic might be replaced by a trigonometric approximation to facilitate subsequent analysis. In these instances, we want the interpolating curve to approximate closely the essential characteristics of the function it replaces.
8.2 Sources of error in the modelling process

When fitting models to data or interpolating, we need to examine the modelling process in order to ascertain where errors can arise. If error considerations are neglected, undue confidence may be placed in intermediate results, causing faulty decisions in subsequent steps. Our goal is to ensure that all parts of the modelling process are computationally compatible and to consider the effects of cumulative errors likely to exist from previous steps.

We can classify errors under the following category scheme:

1. Formulation error
2. Truncation error
3. Round-off error

A formulation error is an error resulting from the assumption that certain variables are negligible or from simplifications arising in describing interrelationships among the variables in the various submodels. For example, if we ignore the spatial heterogeneity of a population as an influenza pandemic sweeps through, we may be neglecting important relationships among individuals that facilitate or prevent the transmission of the disease. Formulation errors are present in even the best models. As George Box famously said, “All models are wrong, but some are useful.”

Truncation errors are those errors attributable to the numerical method used to solve a mathematical problem. For example, we may find it necessary to approximate $e^x$ with a polynomial representation obtained from the power series

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \cdots.$$ 

An error will be introduced when the series is truncated (i.e. only a finite number of terms are included) to produce the polynomial. In fact, every time your calculator or computer calculates an exponential (or a sine or cosine or a logarithm), it is using just such a finite polynomial representation, complete with truncation error.

Round-off error refers to any error caused by using a finite digit machine for computation. Since all numbers cannot be represented exactly using only finite representations, we must always expect round-off errors to be present. For example, consider a calculator or computer that uses 8-digit arithmetic. Then the number $1/3$ is represented by 0.33333333 so that $3 \times 1/3 = 0.99999999$, rather than the actual value 1. The error of $10^{-8}$ is due to round-off. The ideal real number 1/3 is an infinite string of decimal digits 0.3333333..., but any calculator or computer can do arithmetic only with numbers having finite precision. When many arithmetic operations are performed in succession, each with its own round-off, the accumulated effect of round-off can significantly alter the numbers that are supposed to be the answer. Round-off is just one
of the things we have to live with – and be aware of – when we use computing machines.

*Measurement errors* are caused by imprecision in the data collection. This imprecision may include such diverse things as human errors in recording or reporting the data, or the actual physical limitations of the laboratory equipment. For example, considerable measurement error would be expected in the data reflecting the spread of influenza through a population, since we can’t measure everyone, fatalities caused by influenza are often attributed to other factors and the speed of spread may outpace the ability to measure symptoms.

### 8.3 Visual fitting with the original data

Suppose we want to fit the model \( y = ax + b \) to the data shown in Figure 8.4. How might the constants \( a \) and \( b \) be chosen to determine the line that “best fits” the data? Generally, when more than two data points exist, all of them cannot be expected to lie exactly along a straight line, even if such a line accurately models the relationship between the two variables \( x \) and \( y \). That is, ordinarily there will be some vertical discrepancy (residuals) between some of the data points and any particular line being considered. We refer to these vertical discrepancies as *absolute deviations*.

![Fig. 8.4. Each data point is thought of as an interval of confidence.](image)

For the “best fitting” line we might want to try to minimise the sum of these absolute deviations leading to the model depicted in Figure 8.5. While
we may be successful at minimising the sum of the absolute deviations, the absolute deviation from individual points may be quite large. For example, while points A, B and C are quite close to the fitted line, point D is some considerable distance from it. If we have confidence in the accuracy of this data point, there would be concern for the predictions made from the fitted line near this point.

As an alternative, suppose a line is selected that minimises the largest deviation from any point. In this case, we would have the line shown in Figure 8.6. In this case, no point is exactly on the line, but no point is too far from it either.

Although these visual methods for fitting a line to data points may appear imprecise, the methods are often quite compatible with the accuracy of the modelling process itself. The grossness of the assumptions and the imprecision involved in the data collection may not warrant a more sophisticated analysis. In such situations, the blind application of more analytical methods may lead to models far less appropriate than one obtained graphically.

Furthermore, a visual inspection of the model fitted graphically to the data immediate gives an impression of how good the fit is and where it appears to fit well. Unfortunately, these important considerations are often overlooked in problems with large amounts of data analytically fitted via computer codes. Since the model-fitting portion of the modelling process appears to be more...
precise and analytic than some of the other steps, there is a tendency to place undue faith in the numerical computations.

8.4 Transforming the data

Most of us are limited visually to fitting only lines. So to graphically fit other curves as models, we have to transform the data. For example, consider the data shown in Table 8.1 of new cases of HIV infections detected in 1981. The data is plotted in Figure 8.7

<table>
<thead>
<tr>
<th>Month</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Cases</td>
<td>51</td>
<td>179</td>
<td>370</td>
<td>1207</td>
</tr>
</tbody>
</table>

Table 8.1. Collected data for new HIV infections in 1981.

We may suspect that the relationship is exponential, i.e. of the form $y = Ce^x$, where $x$ is the time in months since the beginning of the survey and $y$ is the number of cases. Thus, if we plot $y$ versus $e^x$, we should obtain approximately a straight line. See Figure 8.8. Since the plotted data points do lie approximately along a line that projects through the origin, we conclude that the assumed proportionality is reasonable. From the figure, the slope of the line is approximated as...
An alternative technique involves taking the logarithm of each side of the equation $y = Ce^x$ to obtain
Fitting curves to data

\[ \ln y = \ln \left[ Ce^x \right] \]

\[ \ln y = \ln C + \ln e^x \quad \text{(since } \ln(ab) = \ln(a) + \ln(b)) \]

\[ \ln y = \ln C + x \quad \text{(remember } \ln \text{ and } e \text{ are inverses).} \]

Note that this expression is an equation of a line in the variables \( \ln y \) and \( x \). The number \( \ln C \) is the intercept when \( x = 0 \). The transformed data are shown in Table 8.2 and plotted in a “semi-log” plot, Figure 8.9.

<table>
<thead>
<tr>
<th>( x )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ln y )</td>
<td>3.932</td>
<td>5.167</td>
<td>5.914</td>
<td>7.096</td>
</tr>
</tbody>
</table>

Table 8.2. Transformed data

![Figure 8.9. Plot of \( \ln y \) versus \( x \) for the transformed data.](image)

From Figure 8.9 we can determine that the intercept \( \ln C \) is 2.9776, giving \( C = e^{2.9776} \approx 19.6 \). So which \( C \) is the right one? Answer: probably neither. Which \( C \) do we have more faith in? Answer: definitely the second one! It uses more data points (using all four to determine the line of best fit and hence the intercept), whereas the first \( C \) only uses two data points to determine the slope. And we can see from Figure 8.8 that this isn’t going to be the exact slope of the fitted line anyway. Of course, a smarter approach to this would be to use linear regression to calculate the slope of the best-fit line.

So does that mean we should always transform our data into something where we can fit a line, if that’s possible? Well... not necessarily. As we’ll
8.4 Transforming the data

see in the next section, this too has the potential to mislead. It’s a tough curve-fitting world out there.

8.4.1 Regression coefficients can be misleading

A similar transformation can be performed on a variety of other curves to produce linear relationships among the resulting transformed variables. For example, if \( y = x^a \), then

\[
\ln y = \ln(x^a) \\
\ln y = a \ln x
\]

(remember \( \ln(b^c) = c \ln(b) \)) is a linear relationship in the transformed variables \( \ln y \) and \( \ln x \), giving us a “log-log” transformation with slope \( a \).

For example, consider the data in Table 8.3, where \( x \) represents the number of avian influenza infections detected and \( y \) represents the number of birds that must be culled to contain the disease.

<table>
<thead>
<tr>
<th>( x )</th>
<th>3</th>
<th>7</th>
<th>20</th>
<th>148</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y )</td>
<td>8</td>
<td>65</td>
<td>549</td>
<td>36300</td>
</tr>
</tbody>
</table>

Table 8.3. Initial infections and cull size for an avian influenza outbreak.

If you use any basic linear regression package (a simple calculator will do), you’ll find that a line of best fit could be applied, with \( r = 0.9956 \). Seems pretty good, right? But if we plot the data, as in Figure 8.10, you can see that the fit isn’t nearly so good as we might suspect. (Indeed, the inset shows just how far off the line is for the first three data points, due to the scaling!) This is one of the dangers of relying on computers to do the work for us: we can easily be fooled into believing results that are misleading or outright wrong.

If instead we suspect a relationship of the form \( y = x^a \) and plot \( \ln y \) versus \( \ln x \), as in Figure 8.11, then we find \( r = 0.9995 \) and the slope of the line of best fit is 2.1496, suggesting that \( a \approx 2.1 \).

Finally, we can check this curve against the original data, by plotting the curve \( y = x^{2.1} \) against the original data, as in Figure 8.12. This looks a lot better than Figure 8.10. It’s at this point, by fitting the curve to the original data, that we make our decisions about which curve is the best.

Note that the linear regression \( r \) values weren’t terribly helpful here. Be careful: there’s a tendency for scientists to put all their faith in the \( r \) value they’ve calculated (even for very small data sets like the ones we used) and many are willing to accept \( r \) values a lot lower than the ones seen here.
8.4.2 Transformations can also be misleading

At this point, we must make an important observation. Suppose we do invoke a transformation and plot ln\(y\) versus \(x\), as in Figure 8.12 and find the line that successfully minimises the sum of the absolute deviations of the transformed data points. The line then determines ln\(C\), which in turn produces the
8.4 Transforming the data

Fig. 8.12. Comparing the exponential curve with the original avian influenza data. Inset: a rescaling of the first three data points.

proportionality constant $C$. Although it is not obvious, the resulting model $y = Ce^x$ is not the member of the family of exponential curves of the form $ke^x$ that minimises the sum of the absolute deviations from the original data points (when we plot $y$ versus $x$).

That is, the line may be the best fit in the transformed data, but it doesn’t follow that the corresponding curve is necessarily the best fit in the original.

When transformations of the form $y = \ln x$ are made, the distance concept is distorted. While a fit that is compatible with the inherent limitations of a graphical analysis may be obtained, we must be aware of this distortion and, crucially, verify the model using the graph from which it is intended to make predictions: namely the $y$ versus $x$ graph in the original data, rather than the graph of the transformed variables.

For example, consider the data plotted in Figure 8.13, which might represent biannual outbreaks of influenza. Suppose we have reason to believe the data are expected to fit a model of the form $y = Ce^{1/x}$. We want to choose the “best” $C$ that fits this. Using a logarithmic transformation as before, we find

$$\ln y = \ln \left( Ce^{1/x} \right)$$

$$\ln y = \ln C + \ln \left( e^{1/x} \right) \quad \text{(since } \ln(ab) = \ln a + \ln b)$$

$$\ln y = \frac{1}{x} + \ln C \quad \text{(since } \ln \text{ and } e \text{ are inverses).}$$
A plot of the points $\ln y$ versus $1/x$ based on the original data is shown in Figure 8.14. Note from the figure how the transformation distorts the distances between the original data points and squeezes them all together. Consequently, if a straight line is made to fit the transformed data plotted in Figure 8.14, the absolute deviations appear relatively small (that is, small compared on the Figure 8.14 scale rather than on the Figure 8.13 scale).

This means that the model is a reasonably good fit for the transformed data, with $\ln C \approx -1.25$. It might not look so great to our eye, but the deviations are quite small, so a computer would tell us it’s a very good fit. We could then solve for $C$ and assume we’ve got the best $C$ (because we had the best $\ln C$).

But let’s test this out on the original data. If we plot the fitted model $y = Ce^{1/x}$ to the data in Figure 8.13, you would see that it fits the data relatively poorly, as shown in Figure 8.15. There are no biannual peaks to the fitted line and worse, the line would behave quite badly (heading up to infinity) in the vicinity of 0.

What’s gone wrong here? Answer: The data were never supposed to fit a model of the form $y = Ce^{1/x}$. But we wouldn’t know that from the transformation, which tells us that the fit is actually pretty good.

From this example, we can see that if we are not careful when using transformations, we can be tricked into selecting a relatively poor model. This realisation becomes especially important when comparing alternative models. Very serious errors can be introduced when selecting the best model unless all the comparisons are made with the original data. Otherwise, the choice of “best” model may be determined by a peculiarity of the transformation rather
than on the merits of the model and how well it fits the original data. While the danger of making transformations is evident in this example, it is easy to be fooled if we are not especially observant, since many computer codes fit models by first making a transformation.

Fig. 8.14. The transformed data points and a line of best fit.

Fig. 8.15. A plot of the curve $y = Ce^{1/x}$ based on the value $\ln C \approx -1.25$. 
8.5 Lab work

The problem.

In the data in Table 8.4, \( x \) is the diameter of a ponderosa pine in inches measured at chest height and \( y \) is a measure of volume: the number of board feet divided by 10. (Most lumber is sold by the board foot, which is equal to a board that is one foot long, one foot wide, and one inch thick.)

<table>
<thead>
<tr>
<th>x</th>
<th>17</th>
<th>19</th>
<th>20</th>
<th>22</th>
<th>23</th>
<th>25</th>
<th>28</th>
<th>31</th>
<th>32</th>
<th>33</th>
<th>36</th>
<th>37</th>
<th>38</th>
<th>39</th>
<th>41</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>19</td>
<td>25</td>
<td>32</td>
<td>51</td>
<td>57</td>
<td>71</td>
<td>113</td>
<td>141</td>
<td>123</td>
<td>187</td>
<td>192</td>
<td>205</td>
<td>252</td>
<td>259</td>
<td>294</td>
</tr>
</tbody>
</table>

Table 8.4. Diameter of a ponderosa pine versus volume.

- Use the `polyfit` command to fit polynomials of degree 1, 2, 3, 4, 6, 9 and 15 to the data. Which of these seem reasonable?
- Use the Basic Data fitting tool to fit same polynomials (except for 15) to the data. Which seems reasonable now?
- Use Matlab to plot the points and plot the natural cubic spline joining them.
- Make an appropriate transformation to fit the model \( y = ax^b \).
- Estimate the parameters \( a \) and \( b \) of the model.
- Plot your model against the original data. Which of the earlier polynomials does this most closely approximate?

The solution

First we need to enter the data. We’ll then use `polyfit` to evaluate a range of possible best-fit polynomials.

```matlab
x=[17 19 20 22 23 25 28 31 32 33 36 37 38 39 41];
y=[19 25 32 51 57 71 113 141 123 187 192 205 252 259 294];
n=input('Enter the degree of the polynomial you want to fit ');
p=polyfit(x,y,n);
f=polyval(p,x);
figure(1)
plot(x,y,'o') %This plots a 'clean' set of data for later use
figure(2)
plot(x,y,'o',x,f,'-')
```
Matlab is great for fitting data. If you keep Figure 1 open, then choose “Tools → Basic Fitting”, you have the option of fitting all manner of things to the data. Try fitting linear, quadratic, cubic etc polynomials to the data. How does this look? (You may want to type axis([15 45 0 350]) into the Matlab command window in order to rescale the axes.)

Fitting low order polynomials isn’t too bad, but higher order polynomials don’t seem to work very well. They’re a reasonable fit to the actual data, but outside the region we have data for, they’re an extremely bad fit.

While we’re here, let’s choose the “spline interpolant” option (again, you may want to rescale the axes). This fits a cubic spline to the data. We’ll learn more about the theory of cubic splines in the next chapter, but basically they’re a way to fit a smooth curve to the data that actually passes through all the data points, should we so desire it.

To transform the data, since we have exponents, the obvious transformation is to take the logarithm of both sides. Thus

\[ \ln y = \ln(ax^b) \]
\[ \ln y = \ln a + \ln x^b \]
\[ \ln y = \ln a + b \ln x. \]

Thus, if we plot \( \ln y \) versus \( \ln x \), then the intercept should be \( \ln a \) and the slope should be \( b \).

In Matlab the function \( \ln x \) is represented as log\( (x) \) (for historical reasons, “log” and “ln” are the same thing in mathematics, although confusingly they aren’t for most scientists), so you can type plot(log\( (x) \),log\( (y) \),’o’) directly into the Matlab command window. The ‘o’ means “plot the data as circles”. You could also use ‘*’, ‘+’, etc, but Matlab’s default is to join the data with lines, which we don’t want.

To apply a line of best fit, simply choose “Tools → Basic Fitting” and choose “linear”. Click the button and you’ll see the coefficients of the linear polynomial. Thus,

\[ \ln a = -5.7427 \]
\[ \Rightarrow a = e^{-5.7427} = 0.0032061 \]
\[ b = 3.0919. \]

We can thus plot our model versus the data by typing the following code into the command window:

\[ \text{plot}(x,y,’o’,x,0.0032061.*x.^(3.0919)) \]

Matlab can plot multiple things on top of each other. Every two entries (plus an optional descriptor) is a data set.

We can also see that the cubic polynomial was the closest to the best fit, assuming this model is accurate.
8.5.1 Exercises

Table 8.5 shows the mean number of *T. infestans* bugs collected per infected site, after community-wide insecticide spraying in 1992 to eliminate Chagas' disease in two rural areas in Argentina.

<table>
<thead>
<tr>
<th>Month</th>
<th>Amamá</th>
<th>Mercedes/Trinidad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct-93</td>
<td>2.7</td>
<td>1</td>
</tr>
<tr>
<td>Nov-94</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>May-95</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Nov 95</td>
<td>2.8</td>
<td>4.8</td>
</tr>
<tr>
<td>May-96</td>
<td>3.4</td>
<td>3.7</td>
</tr>
<tr>
<td>Nov-96</td>
<td>3.3</td>
<td>2.8</td>
</tr>
<tr>
<td>May-97</td>
<td>8.7</td>
<td>6.2</td>
</tr>
<tr>
<td>Nov-97</td>
<td>7.4</td>
<td>11.2</td>
</tr>
<tr>
<td>May-98</td>
<td>6.4</td>
<td>5.4</td>
</tr>
<tr>
<td>Nov-98</td>
<td>8.4</td>
<td>3.4</td>
</tr>
<tr>
<td>May-99</td>
<td>7.9</td>
<td>4.7</td>
</tr>
</tbody>
</table>

*Table 8.5. Mean number of bugs per infected site. (Data courtesy Ricardo Gürtler.)*

1. Plot the data for each site on two separate figures. (Watch out for the scaling on the x-axis; the October entry means you'll have to be careful.)
2. Fit some polynomials to each data set by modifying your polyfit program. Which polynomial seems the ‘best’ to you?
3. Use the “Basic Fitting” tools to fit your polynomials again. What does the behaviour around the edges tell you?
4. Fit a cubic spline to the data. Do you think this would be a better fit than a polynomial?
5. Do you have a reason to suspect the data might need to be transformed? If so, make an appropriate transformation and determine the parameters.
6. Is there a qualitative difference between the mean number of bugs per infested site in the two villages?
7. What would you predict would be the mean number of bugs per infected site in Nov 99?