Fitting Curves to Data using Nonlinear Regression

The following was adapted from *Fitting curves to data using nonlinear regression: a practical and nonmathematical review* by Harvey J. Motulsky and Lennart A. Ransnas. A link to the original document can be found on the BOSS page on nonlinear regression. The purpose of this document is to provide background information on what nonlinear regression is, how it works, and how to interpret the results. Additional resources are provided as examples that show how to perform nonlinear regression in MATLAB.

1 Why choose nonlinear regression?

Nonlinear regression is often ignored in statistics textbooks because its mathematical derivations can be extremely complex. Alternatives such as linear regression of transformed data are often presented. For example, the exponential decay function, $y = Ae^{-Bx}$, can be linearized by taking the logarithm of each side of the equation. The result, $ln(y) = ln(A) - Bx$, is a linear plot where the slope and intercept give the parameters of interest. This method is useful because it is straightforward and does not require a computer, but the results are not always statistically optimal. The transformation process can distort experimental errors. Thus, for some nonlinear data sets, nonlinear regression may be the best way to model your data.

2 How does nonlinear regression work?

Let's say there is an independent variable x and a dependent variable y and their relationship can be described by an equation that includes one or more parameters (constants). Nonlinear regression is a procedure that will determine the values for the parameters that minimizes the sum of the squares of the distances of the data points to the curve for the equation of interest. In other words, given the data point y and the corresponding point on the curve $\hat{y}$, nonlinear regression will minimize the sum of the their differences squared: SS=sum$[(y - \hat{y})^2]$. Note that the term $(y - \hat{y})$ is simply the residual for each data point. This type of method is referred to as a least-squares method and is only applicable if the uncertainty is normally distributed.

Computer algorithms are used to solve nonlinear regression problems. They work by iterating through values of the constants in the designated model until the residual sum of squares is minimized. Because of this, the programs have to be given an initial guess of the parameters from which to start their iterations. The initial guess is often based on rough calculations or intuition. For most cases, the algorithm should be able to determine the parameters regardless of the initial guess. However, it is possible for the algorithm to go in the wrong direction and never converge on a solution or for it to converge on the wrong solution. In short, be careful with your choice for the initial guess and always check your result to make sure it behaves reasonably.

3 Does the model fit the data?

The goal of analyzing the results of a nonlinear regression has four components: to determine how well the model fits the data, if the model fits the data better than an alternative model, how much uncertainty there is in the values of the parameters, and if the equation fits this data differently for a different set of data.

1. The first step in assessing how well the model fits the data (goodness of fit) is to graph the curve and the data points on the same plot. If the algorithm worked correctly, the distance between the curve and the data points should appear to be at a minimum. The distances can then be plotted in a residual plot, in which the distance between the data points and the curve are on the y axis and the corresponding independent values are on the x axis. If the residuals appear randomly distributed across the zero line, then the model is a good fit. If the residuals show a pattern, such as a sinusoid, or have long runs of the same sign, then the model is likely not a good fit.

   The goodness of fit can also be measured using quantitative measures. The r^2 value can be determined by using the equation: $r^2 = 1 - SS_{res}/SS_{tot}$, where SS_{res} is the sum of square of the residuals (the value that is minimized by the nonlinear regression procedure) and $SS_{tot} = \text{sum}[(y - \bar{y})^2]$ (the sum of

the squared differences between the data points and the average of the data points). It can be seen that if the value for SS_{res} is minimized, the value for r^2 will approach 1. Thus, the closer r^2 is to 1, the better the model is at fitting the data. r^2 is often referred to as the coefficient of determination and can be interpreted as the proportion of the variation in y that is explained by knowledge of x.

2. It is often useful to find confidence intervals on the calculated parameters. This is easy to do in MATLAB using the calculated residuals and Jacobian (see *Conducting a Nonlinear Fit Analysis in MATLAB* document for more information). The confidence bounds indicate that if the experimental procedure were repeated the parameters calculated from the new data would be within the bounds 95% of the time.

3. Sometimes it is not clear if one model will be a better fit for a given set of data than another. If the residuals and r^2 values for each model are similar, an F-test can be conducted to see which model is better. If both models have the same number of parameters, the formula for the F statistic is $F=SS_1/SS_2$, where SS_1 is the residual sum of squares for the first model and SS_2 is the residual sum of squares for the second model. There are $N - V$ degrees of freedom, where N is the number of data points and V is the number of parameters being estimated (one degree of freedom is lost per parameter estimated). The resulting F statistic can then be compared to an F-table to extract the p-value. Alternatively, the p-value can be computed using MATLAB's built in function 'fcdf'. To do this simply type P=1-fcdf(F,df_1,df_2), where F is the computed F-statistic, and df_1 and df_2 are the degrees of freedom of each model equation. **If the p-value is large (greater than α) then the first model is statistically better than the second. If the p-value is small (less than 1-α) then the second model is statistically better than the first.**

   If the models have different numbers of parameters, the formula becomes:

$$F = \frac{(SS_1 - SS_2)/(df_1 - df_2)}{SS_2/df_2}$$

   The sum of squares for each model and the degrees of freedom for each model are calculated as before (note the models will have different degrees of freedom for this case). Additionally, the first model must be the one with fewer parameters (i.e. the simpler one). Once again, the F-statistic and degrees of freedom can be used to determine the p-value. In this case, use df_1-df_2 and df_2 degrees of freedom when finding the p-value from a table or using the MATLAB . **A p-value less than α indicates that the more complex model (denominator of F-statistic) fits the data significantly better than the simpler model.** For an example, see *Using an F-test To Compare Two Models.*

4. Sometimes one wants to know if two sets of data fit the same general model. This can be accomplished through a relatively simple method. The idea is to first analyze the data separately and then analyze it combined (as if it was one large data set instead of two separate trials). If the separate fit is significantly better than the pooled fit than the two sets of data are not well described by the single curve. To accomplish this, first find the sum of squares and degrees of freedom for the separated data:

$$SS_{sep} = SS_1 + SS_2$$
$$df_{sep} = df_1 + df_2$$

   Where the SS_1 is the residual sum of squares for the model that fits the first set of data and SS_2 is the residual sum of squares for the model that fits the second set of data. Degrees of freedom are calculated by subtracting the number of parameters from the number of data points as before.

   Next, pool all of the data together and find the residual sum of squares and the degrees of freedom for a model that fits the entire data. Finally, calculate the F-statistic:

$$F = \frac{(SS_{pool} - SS_{sep})/(df_{pool} - df_{sep})}{SS_{sep}/df_{sep}}$$

   If the p-value is less than the significance level, the separate fit is much better than the pooled fit and the two data sets cannot be modeled with the same curve.