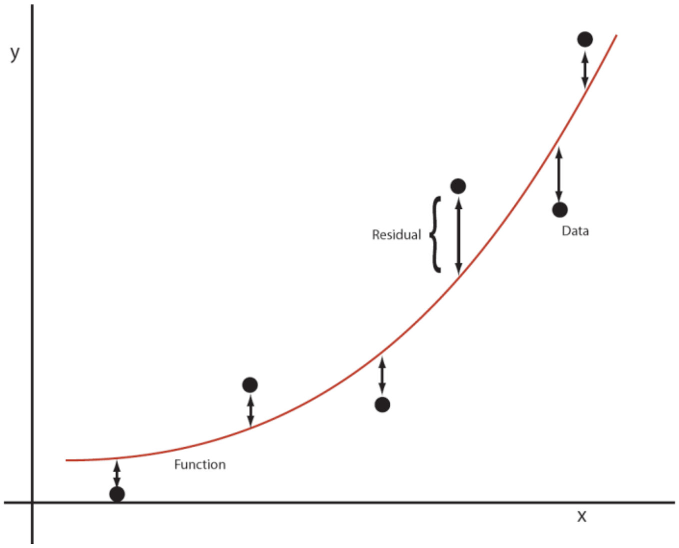
A Brief Introduction to Fitting Data to Rate Laws

For some biological processes, the underlying mechanisms may be less important than the response itself. This is particularly true of gene expression models and enzyme catalysis. Often, simply the expression rate /Transcription Factor response is sufficient to know without being burdened by the complexities of the underlying chemistry. In such situations it is enough to fit the response data to some suitable function and treat the fitted function as a representation of the process. This is a tried and tested technique in engineering. This approach is also eminently suitable for synthetic biology where one can construct in vivo component parts which have measurable inputs and outputs.

The question is how does one go about fitting experimental data to a specific mathematical function? The classic approach is to minimize the sums of squares of the residuals. The figure below illustrates a function and a set of data we wish to fit to the function. Just as in linear regression, the quality of the fit is measured by how much each data point deviates from the function curve, the so-called **residuals**. To get an overall measure of the fit, the residuals are squared to remove any negative values and then summed. Algorithms are then devised that adjust the parameters in the function in order to minimize the residual sum of squares (RSS).

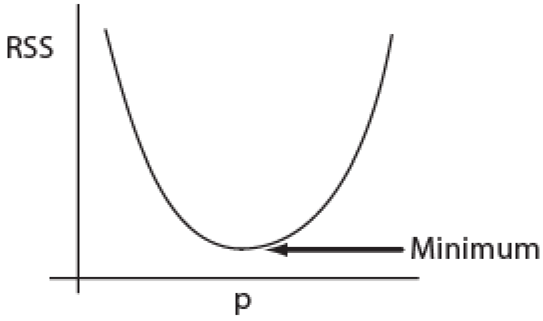


Minimize:

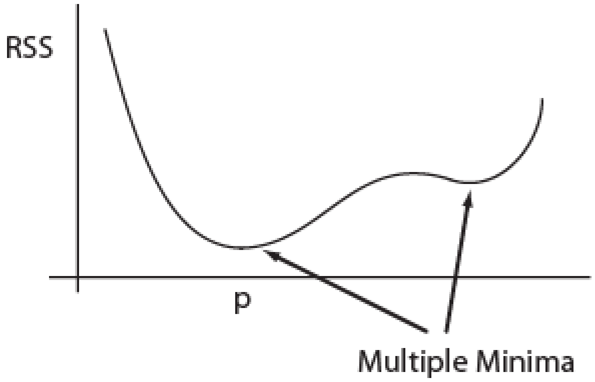
$$RSS = \sum_{i=1}^{n} \left(y_i - f(x_i, p)\right)^2$$

Where n is equal to the number of data points, $y_i$is the value of the i^{th}data point and $f(x_i, p)$ is the value of the function at the i^{th}position. p is the list of parameters that influence the shape of the function.

The residual sums of squares has a minimum where the fitting function is poised exactly midway between all the data points.



In other words we want to find a value for p that minimizes the value of the RSS. Unfortunately, finding the minimum is a little bit more troublesome that it might appear. The biggest problem is that the shape of the RSS/p curve shown above is usually much more complicated and rather than having one minimum, it will often have multiple minima.

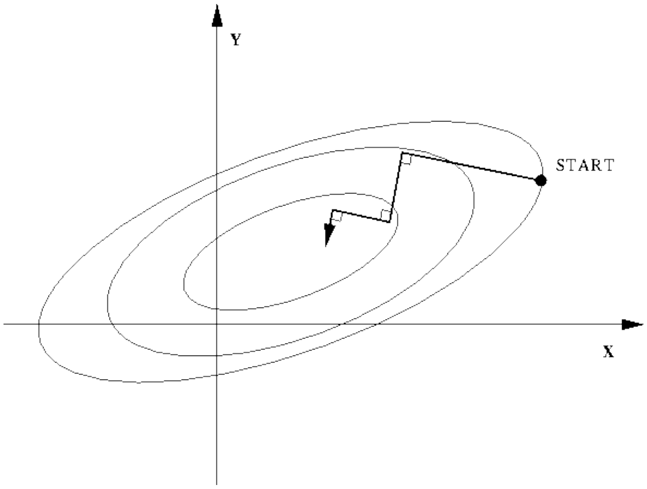


In addition, many problems will have more than one parameter so that the curve is actually a surface, possibly in high dimensions.

One of the central efforts in curve fitting is to find the so-called **global minimum** of this surface since this will represent the best fit. The simplest way to find the minimum is to move down the steepest portion of the surface until we reach a point where the surface starts climbing. The formula for this technique is:

$$x_{i+1} = x_i - \ h \ \frac{\partial f}{\partial x}$$

and is called **gradient or steepest descent.** The h factor can be adjusted to change the rate at which we proceed down the surface and ensures that we don't travel past the minimum.



Steepest Descent: From http://trond.hjorteland.com/thesis/node26.html

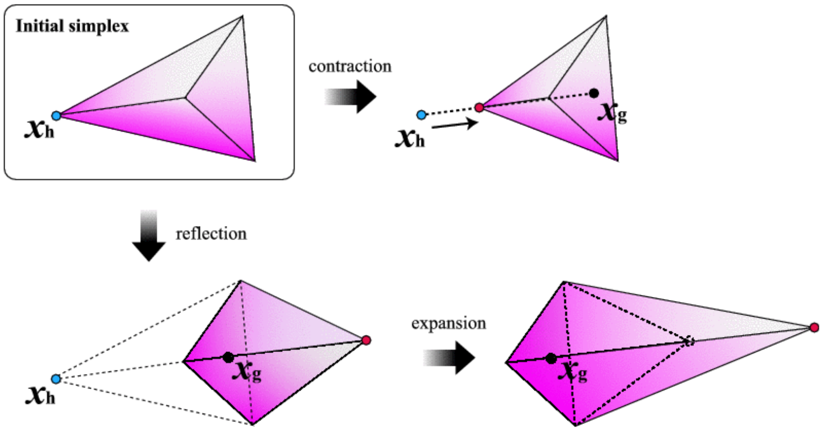
In practice the method does not work well because the search surface tends to be quite complicated and will often end up in a local minimum. In addition convergence can be quick slow as it tends to zig-zag across the surface which is particularly noticeable is the surface has a valley like shape. Finally, near the bottom of the minimum where the slope is likely to be shallow, steepest descent tends to move very slowly.

Luckily there are available quite a number of methods that can search complicated search spaces. Some of the most important include:

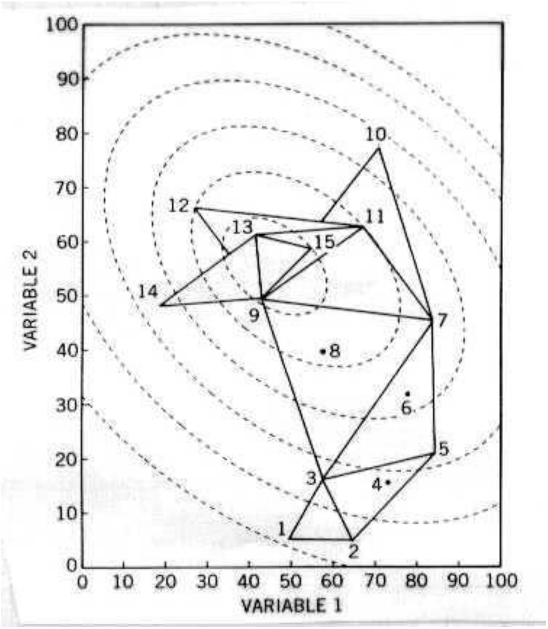
Levenberg-Marquardt: This method combines two ways to move towards the minimum. When far from the minimum it uses gradient decent to rapidly move down the surface. However, as it approaches the minimum it switches to what is called the Gauss-Newton method. This method takes into account the curvature of the surface and allows the method to speed up its descent as it approaches the minimum.

There are many variants that use the steepness of the surface to find the minimum and the Levenberg-Marquardt is probably the most efficient of these. However, all these methods suffer from an inability to find the global minimum since they very easily get trapped in local minima.

Simplex or Nelder-Mead Method: Although the simplex method has the same problem with getting stuck in local minima as the previously mentioned methods, it is a very appealing geometric approach. The method uses the idea of a simplex, an object that in an N dimension space has N+1 vertices. On a two dimensional surface the simplex is a triangle. The idea is that each vertex (point on the simplex) samples a piece of the surface. By comparing the different points of contact the algorithm changes the shape of the simplex, either through contraction, expansion or reflection, so that the simplex moves towards the minimum. The simplex is adept at moving quickly down steep slopes and squeezing through narrow valleys. When it gets close to the minimum it will shrink down and trigger a stopping decision when it can no longer improve.



From http://mikilab.doshisha.ac.jp/dia/research/report/2006/0806/006/report20060806006.html



From http://oregonstate.edu/instruct/ch490/lessons/lesson10.htm

Simulated Annealing: Simulated annealing is a method that follows a procedure analogous to the cooling of a liquid to produce nice well formed crystals. If a solution is cooled quickly, the crystals that are formed are likely to be small and irregular. If however the liquid is cooled slowly then large well formed crystals form. Simulated annealing uses a similar approach. Initially the method tumbles around the search space, this allows the method to search a wide area without getting stuck in a local minimum. As the ‘temperature’ of the method is lowered, the method becomes trapped in the lowest minima that it has reached. The method therefore avoids setting at the first minimum point it finds and is more likely to find the global minimum.

Genetic Algorithm: Genetic algorithms use a population of candidate solutions. Such solutions, which includes the parameter values in the function being fitted, are encoded as binary strings but other representations are also possible. Each individual in the population has a fitness associated with it which indicates how close it is to the minimum, computed using the bit string. Populations move through generations and at each generation, individuals with a low fitness are culled while those with a high fitness are kept and used as breeding stock to create the next generation. The method may employ a variety of means to mutate individuals, including simple bits flips in the genomes or more elaborate recombination of individuals to form hybrid solutions. The method has proved to be quite successful although it can sometimes be slow to converge. However, its virtue, like simulated annealing, is that it is capable of sampling a wide area of the surface being minimized and is thus more likely to find the global minimum.

Data fitting Software

It is my experience that data fitting software that is easy to use is not common. Tools such as Matlab and Excel can, after some effort, be used to fit data to functions, however there are some easier and less painful data fitting tools available online including:

The Colby fitter at:

http://www.colby.edu/chemistry/PChem/scripts/lsfitpl.html

The Colby fitter has the limitation that you have to choose a function from a limited selection (21 function types). In addition only a maximum of three parameters can be fitted at once. However it is easy to use and it will plot the final fit which is an advantage.

Another online data fitter is the ‘Curve Fitter’ at

http://statpages.org/nonlin.html

This is more versatile in that is allows the user to select the function they wish to fit and offers many more parameters that can be fitted.

Finally, I will mention a tool called JSim. This tool runs from the desktop and allows a user to fit an arbitrary function to data. We will use this tool in class.

Class Exercise

A text file is provided on the Wiki that contains artificially generated data. The data comprises two columns, the first column is nominally labeled TF1 or TF2 and the second column GFP florescence. Your task is to fit this data to a suitablel equation using the Nelder and Mead method found in JSim and to report estimates for the Hill coefficient, Vmax and Km constant. A JSim template is provided to fit the first data set but you must construct your own for the second data set and decide on a suitable fitting function.