Curve Fitting Tutorial

1) Load the file "noisy_curves.mat" into MATLAB. The file contains one array of x-values and 8 arrays, each with a distinct set of corresponding y-values produced with noisy added to distinct functions. Using (and perhaps improving on) the methods in the file "curve_fitting.m" and the associated functions that calculated the sum of squared errors for particular curves, find the curve most likely to produce the data for each array of y-values (y1 to y8).

    Notes:

    a) You may want to run the fits a number of times, like 20, to ensure you have reached an optimal solution.

    b) You may also want to start some of the fits from values that seem more reasonable given the data. Sometimes, when curves are nested (e.g. 3rd to 4th order polynomials), using a solution from a simpler function can be a good initial estimate for the more complex function. This is not always helpful though.

2) Load the file "Bimodaldata.mat", which contains an array of x-values and a corresponding array of y-values.

    Are the data best fitted by a single Gaussian or the sum of two Gaussians? To answer this question, you will need to select a method to fit with the sum of two Gaussians (perhaps generate a function that calculates the sum of squared errors as with the single Gaussian fit provided) and use an appropriate penalty factor to prevent overfitting.

3) Load the file "binary_data.mat", which contains an array of x-values with a corresponding array of binary y-values. In this case, assume the data arise from a logistic function, such that for any value of x, the probability of a response ($y = 1$) is given by:

$$P(y = 1|x) = \frac{1}{1 + \exp\left[-\left(x - x_{1/2}\right)/\sigma_x\right]}.$$

    Find the most likely values of $x_{1/2}$ (the value of x where the probability of a response is ½) and of σ_x (the range over which responses increase from mostly zero to mostly one). To do this you will need to write a MATLAB code with initial values for the two parameters and use a minimization routine, such as fminsearch (all as in curve_fitting.m) and write a function that returns the negative of the log-likelihood of the data given the parameters of the function.