Nonlinear regression

In the following questions, you will see that in some cases, nonlinear regression is not substantially different to how linear regression can be performed. The purpose of the following questions is to understand how the objective function is obtained, how the normal equations are obtained, and how they can be solved to find the values of the coefficients.

1. Consider the following set of data:

i	1	2	3	4
x_i	-4	-2	2	5
y_i	-1	0	2	3

   Your aim is to fit these data points to a quadratic function of the form $f(x) = a_0 + a_1x + a_2x^2$ using least squares regression.

   (a) Write down the objective function that needs minimising.

   (b) Implement the objective function of (a) in MATLAB and then use `fminsearch` function to approximate the coefficients.

   (c) Use the general normal equations from Section 2.3 of the reading notes to write down the system of linear equations that can be solved to find the coefficients. Express the system in matrix form, and use MATLAB to solve it.

   (d) Write down matrices $\mathbf{A}$, $\mathbf{a}$ and $\mathbf{y}$ for which solving $\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y}$ finds the coefficients. Implement them in MATLAB and solve the system of equations to find the coefficients.

2. The system of normal equations for fitting a polynomial of degree m can be written as

$$\sum_{j=0}^{m}\left(a_j\sum_{i=1}^{n}x_i^{j+k}\right) = \sum_{i=1}^{n}y_ix_i^k, \quad k \in \{0, \dots, m\}.$$

   Show that this system has the form $\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y}$, where $\mathbf{A} = \begin{pmatrix}\mathbf{1} & \mathbf{x} & \dots & \mathbf{x}^m\end{pmatrix}$ and $\mathbf{x}^k$ is the column vector $\begin{pmatrix}x_1^k & \dots & x_n^k\end{pmatrix}^T$.

For the remaining questions, use the file `workshop4_data.csv` available on the LMS. Take note of which column corresponds to which variable and take note of which variables are used in each question.

3. Define suitable matrices in MATLAB and then apply suitable operations to find coefficients for the quadratic function $y = a_0 + a_1x + a_2x^2$ that best fit the data in `workshop4_data.csv`. Construct an appropriately labelled figure that includes both a scatter plot of the data points and the graph of the quadratic function you found.

4. Define suitable matrices in MATLAB and then apply suitable operations to find coefficients for the function $r = a_0\exp(q) + a_1\sin(q) + a_2\frac{1}{q}$ that best fit the data in `workshop4_data.csv`. Construct an appropriately labelled figure that includes both a scatter plot of the data points and the graph of the function you found.

Linearisation

For many nonlinear functions, such as $f(x) = ae^{bx}$, the best fit cannot be found exactly using the algebraic techniques of the previous questions. Below, you will see how linearisation can be applied in some cases to transform the data into a form that is suitable for the exact methods. In Question 8, you will see a comparison of linearisation with the corresponding unlinearised solution.

5. Show that fitting an exponential function of the form $f(x) = ae^{bx}$ to data of the form (x_i, y_i) corresponds to fitting a linear function $l(x) = mx + c$ to the data $(x_i, \log(y_i))$. What is the relation between the constants?

6. Using linearisation, construct suitable matrices in MATLAB to fit an exponential function $s = ae^{bq}$ to the data in `workshop4_data.csv`. Construct an appropriately labelled figure that includes both a scatter plot of the data points and the graph of the function you found.

7. Using linearisation, construct suitable matrices in MATLAB to fit a power function $s = aq^b$ to the data in `workshop4_data.csv`. Construct an appropriately labelled figure that includes both a scatter plot of the data points and the graph of the function you found.

8. Consider the following data:

i	1	2	3	4	5	6	7
x_i	0.28	0.76	0.93	1.88	3.03	4.73	4.90
y_i	1.62	1.22	1.80	1.03	1.17	0.70	0.22

We wish to fit to a power model $y = ae^{bx}$ using least squares regression.

(a) Write down the objective function that needs minimisation *without* using linearisation. Implement this objective function in MATLAB and then find a minimiser using `fminsearch`.

(b) Write down the objective function that needs minimisation using linearisation. Find a minimiser of this objective function.

(c) Construct a figure that includes a scatter plot of the data points, and the graph of each function you found in parts (a) and (b).

More MATLAB

9. Consider the `solveQuadratic` function of Workshop 1. The function was designed to handle a single quadratic at a time. Instead, imagine you had a table of quadratic equations, represented by a matrix `M`; e.g.,

```
M = [2 4 1; 4 1 1; 1 2 5];
```

which represents three quadratic equations, $2x^2 + 4x + 1 = 0$, $4x^2 + x + 1 = 0$ and $x^2 + 2x + 5 = 0$.

Write a function `solveQuadratics` that takes a single matrix `table` as input, and returns a column matrix whose entry in row `i` is the solution to the quadratic equation corresponding to row `i` of `table`.

10. Write the following MATLAB functions:

(a) A function `swapRows` that takes as input a matrix `M` and two row numbers `i` and `j`. It returns the matrix that results from swapping rows `i` and `j` in `M`.

(b) A function `multiplyRow` that takes as input a matrix `M`, a row number `i`, and a constant `c`. It returns the matrix that results from multiplying row `i` of `M` by `c`.

(c) A function `addRow` that takes as input a matrix `M`, two row numbers `i` and `j`, and a constant `c`. It returns the matrix that results from adding `c` times row `i` to row `j` in `M`.

(d) A function `rowLeader` that takes as input a matrix `M` and a row number `i`. If row `i` has a row leader, it returns the column number of the row leader in row `i`. Otherwise, it returns the number 0.

(e) A function `pivot` that takes as input a matrix `M`, a row number `i`, and a column number `j`. If `M(i,j)` is non-zero, it returns the matrix that results from rescaling that row so that entry `M(i,j)` becomes 1, and then performs the row operations that set out all other entries in column `j` to 0. If `M(i,j)` is zero, it returns the input matrix `M` unchanged.

11. Use the functions in Question 10 to perform Gaussian elimination on the matrices of Workshop 3.

12. **Challenge**: use your functions in Question 10 to write a MATLAB function `gauss` that performs Gaussian elimination on a matrix `M`.