General Least Squares

We now derive the least-squares solution, and illustrate that it provides the best numerical estimate for the constants. Suppose we have an equation of the form:

$$A_{i1}C_1 + A_{i2}C_2 + A_{i3}C_3 + A_{i4}C_4 + ... + A_{iK}C_K = D_i$$

Where A_{ij} is a coefficient to a constant , C_j is a constant and D_i is the driving term. K is the number of undetermined constants. For a series of data points, we can imagine the following set of equations where $N > K$:

$$\begin{array}{c} A_{11}C_1 + A_{12}C_2 + A_{13}C_3 + A_{14}C_4 + ... + A_{1K}C_K = D_1 \\ A_{21}C_1 + A_{22}C_2 + A_{23}C_3 + A_{24}C_4 + ... + A_{2K}C_K = D_2 \\ A_{31}C_1 + A_{32}C_2 + A_{33}C_3 + A_{34}C_4 + ... + A_{3K}C_K = D_3 \\ . \\ . \\ . \\ A_{N1}C_1 + A_{N2}C_2 + A_{N3}C_3 + A_{N4}C_4 + ... + A_{NK}C_K = D_N \end{array}$$

This set of equations can be written using subscript notation:

$$A_{ij}C_j = D_i$$

where i ranges from 1 to N, and j ranges from 1 to K.

We define a residual, R, which is the difference between the actual value D, and the value computed using some estimate of the constants, C_j. A single equation becomes

$$A_{i1}C_1 + A_{i2}C_2 + A_{i3}C_3 + A_{i4}C_4 + ... + A_{iK}C_K - D_i = R_i$$

R_i, D_i, and C_j must all have the same dimensions. This implies that A_{ij} is dimensionless. For the entire multilayer the equations become, in subscript notation:

$$A_{ij}\,C_j - D_i = R_i \tag{1}$$

An estimate of the total error is the sum of the R values. However, individual R values may have different signs, so an unbiased measure of the error, M, is the sum of the squares of the residuals, R.

$$M = \sum_{i=1}^{N} R_i^2 = R_1^2 + R_2^2 + R_3^2 + ... + R_N^2 = R_pR_p$$

The total error, M, is minimized when $\partial M/\partial C_j$ is zero. For the differentiation, the actual values of constants, C_j, need not be known. Taking the derivative of M with respect to the first constant, C_1, gives

$$\frac{\partial M}{\partial C_1} = 0 = \frac{\partial R_1^2}{\partial C_1} + \frac{\partial R_2^2}{\partial C_1} + \frac{\partial R_3^2}{\partial C_1} + ... + \frac{\partial R_N^2}{\partial C_1}$$

$$\frac{\partial R_1^2}{\partial C_1} = \frac{\partial}{\partial C_1}\left(A_{11}C_1 + A_{12}C_2 + A_{13}C_3 + ... + A_{1(4k)}C_{4k} - D_1\right)^2$$

$$= 2A_{11}(A_{1j}C_j - D_1)$$

Similar expressions can be derived for the other terms:

$$\frac{\partial R_2^2}{\partial C_1} = \ = 2A_{21}(A_{2j}C_j - D_2), \quad \frac{\partial R_3^2}{\partial C_1} = \ = 2A_{31}(A_{3j}C_j - D_3), \quad \text{etc.}$$

Summing terms,

$$\frac{\partial M}{\partial C_1} = \ \ 2A_{i1}(A_{ij}C_j - D_i)$$

In general,

$$\frac{\partial M}{\partial C_p} = \ \ 2A_{ip}(A_{ij}C_j - D_i)$$

The error is minimized when $\frac{\partial M}{\partial C_p}$ is zero:

$$2A_{ip}(A_{ij}C_j - D_i) = 0$$

This can be written in the following form:

$$A_{ip}A_{ij}C_j - A_{ip}D_i = 0$$

The subscripts *p* and *j* range from 1 to *K*, the number of constants, and the subscript *i* ranges from 1 to *N*, the number of matching equations. This equation gives the following square matrix:

$$\left|\begin{array}{l} A_{i1}A_{i1} \quad A_{i1}A_{i2} \ldots A_{i1}A_{iK} \\ A_{i2}A_{i1} \quad A_{i2}A_{i2} \ldots A_{i2}A_{iK} \\ A_{i3}A_{i1} \quad A_{i3}A_{i2} \ldots A_{i3}A_{iK} \\ . \\ . \\ . \\ A_{iK}A_{i1} \quad A_{iK}A_{i2} \ldots A_{iK}A_{iK} \end{array}\right| \left|\begin{array}{c} C_1 \\ C_2 \\ C_3 \\ . \\ . \\ . \\ C_K \end{array}\right| = \left|\begin{array}{c} A_{i1}\,D_i \\ A_{i2}\,D_i \\ A_{i3}\,D_i \\ . \\ . \\ . \\ A_{iK}\,D_i \end{array}\right|$$

Comparing the cell (2,1) (row 2, column 1) with cell (1,2), and cell (4*k*,1) with cell (1,4*k*), it can be seen that they are identical. This indicates that the matrix is diagonally symmetric. For any cell in the matrix, the only repeated subscript is *i*, the row counter. Thus, for any cell in the matrix, we sum on *i*. This indicates that the matrix can be generated by manipulating only one row of the *A* matrix at a time, rather than multiplying two complete *A* matrices.

The constants, C_j, which minimize the residuals (eq. 1) can then be solved for by inverting the coefficient matrix. Computationally, it is unnecessary to completely invert the matrix, and we use a LU-decomposition with back substitution to solve for the constants.

From an examination of the above coefficient matrix we see that it is:

$$A^T\, A\, C = A^T\, D$$

Solving gives

$$C = (A^T\, A)^{-1}\, A^T\, D$$

Thus finding the coefficients of the general polynomial is a simple operation in Excel.