Least Squares Curve Fitting & Gaussian Elimination

Eric Angle
August 15, 2013

1 Least Squares Curve Fitting

We aim to approximate a given function $y(x)$ over an interval x_1 to x_2 by the function $f(x, a_i)$,

$$y(x) = f(x, a_i) + \epsilon(x),$$

where a_i are parameters and $\epsilon(x)$ is the error. To optimize the approximation, we minimize a suitably chosen norm of the error[1],

$$U = \int_{x_1}^{x_2} dx\ \epsilon^2(x),$$

by setting to zero $\partial U/\partial a_i$:

$$0 = \frac{\partial U}{\partial a_i} = 2\int_{x_1}^{x_2} dx\ \epsilon\ \frac{\partial \epsilon}{\partial a_i} = -2\int_{x_1}^{x_2} dx\,(y - f)\,\frac{\partial f}{\partial a_i} \quad \Rightarrow \quad \int_{x_1}^{x_2} dx\ f\ \frac{\partial f}{\partial a_i} = \int_{x_1}^{x_2} dx\ y\ \frac{\partial f}{\partial a_i}. \tag{1}$$

2 The Linear Case

If $f(x, a_i)$ is linear in a_i,

$$f(x, a_i) = \sum_i a_i g_i(x), \tag{2}$$

for some functions g_i, equation 1 reduces to the linear system of equations

$$\sum_j \left(\int_{x_1}^{x_2} dx\ g_i\ g_j\right) a_j = \int_{x_1}^{x_2} dx\ g_i\ y \quad \text{or} \quad \sum_j m_{ij} a_j = b_i,$$

where

$$m_{ij} = m_{ji} = \int_{x_1}^{x_2} dx\ g_i\ g_j \quad \text{and} \quad b_i = \int_{x_1}^{x_2} dx\ g_i\ y. \tag{3}$$

3 Polynomial Least Squares

If in equation 2 we choose

$$g_i(x) = x^i, \quad \text{with} \quad i = 0, 1, ..., n,$$

so that we approximate y by an n degree polynomial, equation 3 gives

$$m_{ij} = \int_{x_1}^{x_2} dx\ x^{i+j} = \frac{1}{i+j+1}\left(x_2^{i+j+1} - x_1^{i+j+1}\right) \quad \text{and} \quad b_i = \int_{x_1}^{x_2} dx\ x^i\ y(x). \tag{4}$$

4 An Example

We will henceforth consider the specific example $y(x) = \sin x$, $x_1 = 0$, and $x_2 = 1$. Equation 4 then reduces to

$$m_{ij} = \frac{1}{i+j+1} = h_{(i+1)(j+1)} \quad \text{and} \quad b_i = \int_0^1 dx\ x^i \sin x, \tag{5}$$

[1] Replace $\int_{x_1}^{x_2} dx$ with $\sum_{x_k=x_1}^{x_2}$ and x with x_k for the discrete case, where we're interested in a set of points x_i.

where $h_{ij} = 1/(i+j-1)$ is the Hilbert matrix. We can develop a recursive solution for b_i in equation 5 by defining (here, $j = \sqrt{-1}$)

$$\begin{aligned} c_i + jb_i = d_i &= \int_0^1 dx\ x^i e^{jx} = \int_0^1 dx\ x^i \cos x + j\int_0^1 dx\ x^i \sin x \\ &= -j\int_0^1 dx \left[\frac{d}{dx}\left(x^i e^{jx}\right) - ix^{i-1}e^{jx}\right] \\ &= -je^j + jid_{i-1} \\ &= \sin(1) - ib_{i-1} + j\left[-\cos(1) + ic_{i-1}\right]. \end{aligned}$$

This gives

$$b_i = -\cos(1) + ic_{i-1} = -\cos(1) + i\left[\sin(1) - (i-1)\,b_{i-2}\right] = -\cos(1) + i\sin(1) - i(i-1)\,b_{i-2}, \tag{6}$$

which determines b_i for $i > 1$, with

$$b_0 = \int_0^1 dx\ \sin x = 1 - \cos(1) \quad \text{and} \quad b_1 = \int_0^1 dx\ x\sin x = \int_0^1 dx \left[\cos x - \frac{d}{dx}(x\cos x)\right] = \sin(1) - \cos(1).$$

5 Programming Note

With m_{ij} in equation 5 and b_i in equation 6, it remains to solve for a_i in

$$\sum_{j=0}^{n} m_{ij}a_j = b_i.$$

For programming purposes, it will be convenient to define $M_{ij} = m_{(i-1)(i-1)} = h_{ij}$, $A_i = a_{i-1}$, and $B_i = b_{i-1}$, and solve for A_i in

$$\sum_{j=1}^{n+1} M_{ij}A_j = B_i. \tag{7}$$

Equation 6 must then be modified,

$$B_i = b_{i-1} = -\cos(1) + (i-1)\sin(1) - (i-1)(i-2)\,b_{i-3} = -\cos(1) + (i-1)\sin(1) - \left(i^2 - 3i + 2\right)B_{i-2},$$

where $B_1 = b_0 = 1 - \cos(1)$ and $B_2 = b_1 = \sin(1) - \cos(1)$.

The solutions A_i from equation 7 will generate an approximation

$$\sin(x) \approx \sum_{i=1}^{n+1} A_i x^{i-1}$$

from $x = 0$ to 1.

6 Gaussian Elimination

We aim to numerically solve equation 7 by implementing Gaussian elimination with and without partial pivoting. The algorithms for those methods are as follows.

6.1 Gaussian Elimination without Pivoting

Gaussian elimination without pivoting is implemented as follows:

DO $k = 1$ to n

Loop through rows.
DO $i = k + 1$ to $n + 1$

$M(i,k) = M(i,k)/M(k,k)$ (This would be zero, so we'll store the ratio here.)
Loop through columns.
DO $j = k+1$ to $n+1$
$M(i,j) = M(i,j) - M(i,k)\,M(k,j)$
END DO
$B(i) = B(i) - M(i,k)\,B(k)$
END DO
END DO

6.2 Gaussian Elimination with Pivoting

Gaussian elimination with pivoting is implemented as follows:

DO $k = 1$ to n
Determine pivot element.
$maxvalue = 0$
DO $i = k$ to $n+1$
IF $M(i,k) > maxvalue$
$maxvalue = M(i,k)$
$maxindex = i$
END IF
END DO
Swap pivot row with top row.
IF $maxindex \neq k$
DO $j = 1$ to $n+1$
$swap = M(maxindex, j)$
$M(maxindex, j) = M(k,j)$
$M(k,j) = swap$
END DO
$swap = B(maxindex)$
$B(maxindex) = B(k)$
$B(k) = swap$
END IF
Insert Gaussian elimination without pivoting from section 6.1.
END DO

6.3 Substitution

Both Gaussian elimination methods are followed by the same substitution method:

$A(n+1) = B(n+1)/M(n+1, n+1)$
Loop backwards, starting at bottom row.
DO $k = 1$ to n
$i = n - k + 1$
$sum = B(i)$
DO $j = i+1$ to $n+1$
$sum = sum - M(i,j)\,A(j)$
END DO
$A(i) = sum/M(i,i)$
END DO

7 Error Analysis

We should have some idea of the number of digits that are meaningful in our results. For simplicity, consider the error δA due to an error δB in equation 7:

$$M(A+\delta A) = B + \delta B \quad \Rightarrow \quad M\delta A = \delta B \quad \Rightarrow \quad \delta A = M^{-1}\delta B \quad \Rightarrow \quad |\delta A| \leq \left|\left|M^{-1}\right|\right| |\delta B| .$$

This result, combined with

$$MA = B \quad \Rightarrow \quad ||M||\,|A| \geq |B| \quad \Rightarrow \quad \frac{1}{|A|} \geq \frac{||M||}{|B|},$$

yields

$$\frac{|\delta A|}{|A|} \leq \frac{\left|\left|M^{-1}\right|\right| |\delta B|}{|A|} \leq ||M||\left|\left|M^{-1}\right|\right| \frac{|\delta B|}{|B|} = K(M)\frac{|\delta B|}{|B|}. \tag{8}$$

Now, $|\delta A|/|A| = 10^{-d}$, where d is the number of meaningful digits in A, while $|\delta B|/|B| = \epsilon = 2^{-53} = 1.11 \times 10^{-16}$ using double precision. Using equation 8,

$$10^{-d} \approx K(M)\,\epsilon \quad \Rightarrow \quad d \approx \log_{10}(1/\epsilon) - \log_{10} K(M) .$$

Therefore, no digits are meaningful when $K(M) \approx 1/\epsilon = 9 \times 10^{15}$. Below is a table of condition numbers $K(M(n+1)\times(n+1))$ and meaningful digits d for polynomial degrees n.

n	$K(M)$	d
1	1.9×10^{1}	33.8
2	5.2×10^{2}	30.5
3	1.6×10^{4}	27.1
4	4.8×10^{5}	23.7
5	1.5×10^{7}	20.2
6	4.8×10^{8}	16.8
7	1.5×10^{10}	13.3
8	4.9×10^{11}	9.8
9	1.6×10^{13}	6.3
10	5.2×10^{14}	2.8
11	1.7×10^{16}	-0.6
12	5.6×10^{17}	-4.1

In practice, it was found that both Gaussian elimination methods fail at precisely $n = 10$.