Mathematical Tripos Part IB: Easter Term 2003

Numerical Analysis – Lecture 7

7 Linear least squares

7.1 Statement of the problem

Consider the problem of finding a vector $c \in \mathbb{R}^n$ such that $Ac = y$ where a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $y \in \mathbb{R}^m$ are given and $m > n$. When there are more equations than unknowns, the system $Ac = y$ is called *overdetermined*. In general, an overdetermined system has no solution, but we would like to have Ac and y close in a sense. Choosing the Euclidean distance $\|z\| = (\sum_{i=1}^m z_i^2)^{1/2}$ as a measure of closeness, one obtains

Problem 7.1 (Least squares in $\mathbb{R}^m$) Given $A \in \mathbb{R}^{m\times n}$ and $y \in \mathbb{R}^m$, find

$$c^* = \arg\min_{c\in\mathbb{R}^n} \|Ac - y\|,$$

i.e., find (the argument) $c^* \in \mathbb{R}^n$ which minimizes (the functional) $\|Ac - y\|$.

Discussion 7.2 Problem of this from occur frequently when we collect m observations (x_i, y_i) (which, typically, are prone to measurement error) and wish to exploit them to form an n-variable linear model

$$f(x_i) \approx y_i, \quad i = \overline{1,m}, \qquad f(x) = c_1\phi_1(x) + c_2\phi_2(x) + \cdots + c_n\phi_n(x),$$

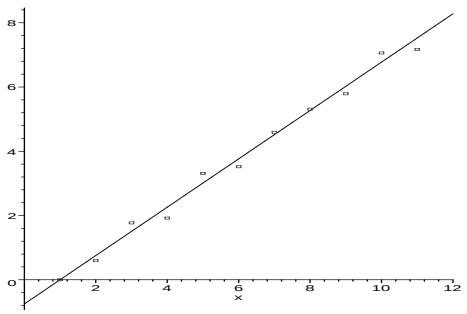
(e.g., trying to put some planet observations on, or near, an ellipse). That is, we want to determine particular values $c_i = c_i^*$ such that $f(x_i)$ would fit (y_i) in a certain way

$$Ac = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & & \vdots \\ \phi_1(x_n) & \cdots & \phi_n(x_n) \\ \vdots & & \vdots \\ \phi_1(x_m) & \cdots & \phi_n(x_m) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \\ \vdots \\ f(x_m) \end{bmatrix} \approx \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \vdots \\ y_m \end{bmatrix} = y.$$

Of course, there will be many ways of doing this, but it is customary to determine c_i^*'s so as to minimize the sum of squares of the deviation, i.e., to minimize

$$\sum_{i=1}^m [f(x_i) - y_i]^2 = \|Ac - y\|^2,$$

for this leads to a *linear* system of equations for determination of the (c_i^*).



x_i	y_i
1	0.00
2	0.60
3	1.77
4	1.92
5	3.31
6	3.52
7	4.59
8	5.31
9	5.79
10	7.06
11	7.17

Fig 7.1 Least squares straight line data fitting

Theorem 7.3 *The vector $c^* \in \mathbb{R}^n$ is a solution to the least squares* $\Leftrightarrow$ $A^T(Ac^* - y) = 0$.

Proof. If c^* is a solution then it minimizes

$$F(c) = \|Ac - y\|^2 = (Ac - y, Ac - y) = c^T A^T Ac - 2c^T A^T y + y^T y,$$

hence the gradient $\nabla F = [\frac{\partial F}{\partial c_1}, \cdots, \frac{\partial F}{\partial c_n}]$ must vanish at $c = c^*$. But

$$\tfrac{1}{2}\nabla F(c) = A^T Ac - A^T y, \quad \text{hence} \quad A^T(Ac^* - y) = 0.$$

Conversely, if $A^T(Ac^* - y) = 0$, then, for any $c' \in \mathbb{R}^n$, with $c := c' - c^*$, we find that

$$\begin{aligned} \|Ac' - y\|^2 &= (Ac + [Ac^* - y], Ac + [Ac^* - y]) \\ &= (Ac^* - y, Ac^* - y) + 2(A^T[Ac^* - y], c) + (Ac, Ac) \\ &= \|Ac^* - y\|^2 + \|Ac\|^2 \geq \|Ac^* - y\|^2, \end{aligned}$$

i.e., c^* is indeed the least squares solution. ∎

Corollary 7.4 *Optimality of c^** $\Leftrightarrow$ *the vector $Ac^* - y$ is orthogonal to all columns of A.*

7.2 Normal equations

Thus, we may find optimal c^* by solving the $n \times n$ system

$$A^T Ac^* = A^T y.$$

This system is called the *normal* equations, $A^T A$ is the *Gram* matrix, c^* is the *normal* solution.

Example 7.5 The least squares approximation to the data plotted in Fig. 7.1 by a straight line

$$f(x) = c_1 + c_2 x \qquad (\phi_1(x) = 1, \quad \phi_2(x) = x)$$

results in the problem $\|Ac - y\|^2 \to \min$ which normal solution is given by

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \vdots & \vdots \\ 1 & 11 \end{bmatrix}, \quad \underbrace{\begin{bmatrix} 11 & 66 \\ 66 & 506 \end{bmatrix}}_{A^T A} \begin{bmatrix} c_1^* \\ c_2^* \end{bmatrix} = \underbrace{\begin{bmatrix} 41.04 \\ 328.05 \end{bmatrix}}_{A^T y} \quad \Rightarrow \quad \begin{bmatrix} c_1^* \\ c_2^* \end{bmatrix} = \begin{bmatrix} -0.7314 \\ 0.7437 \end{bmatrix}.$$

Discussion 7.6 However, the approach based on the normal solution has three disadvantages. Firstly, $A^T A$ might be singular, secondly sparse A might be replaced by a dense $A^T A$ and, finally, forming $A^T A$ might lead to loss of accuracy. Thus, suppose that our computer works in the IEEE arithmetic standard (≈ 15 significant digits) and let

$$A = \begin{bmatrix} 10^8 & -10^8 \\ 1 & 1 \end{bmatrix} \qquad \Rightarrow \quad A^T A = \begin{bmatrix} 10^{16} + 1 & -10^{16} + 1 \\ -10^{16} + 1 & 10^{16} + 1 \end{bmatrix} \approx 10^{16} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Given $y = [0, 2]^T$, the solution of $Ac = y$ is $[1, 1]^T$, however, our computer 'believes' that $A^T A$ is singular!

7.3 QR and least squares

An alternative is provided by the QR factorization. Suppose that $A = QR$, with an orthogonal $Q \in \mathbb{R}^{m\times m}$ and an upper triangular $R \in \mathbb{R}^{m\times n}$. Then, since orthogonal mapping is length-preserving, i.e., $\|Q^T v\|^2 = \|v\|^2$ for any $v \in \mathbb{R}^m$, we have

$$\|Ac - y\| = \|Q^T(Ac - y)\| = \|Rc - Q^T y\|$$

therefore we may seek $c \in \mathbb{R}^n$ that minimises $\|Rc - Q^T y\|$.

Suppose for simplicity that $\operatorname{rank} R = \operatorname{rank} A = n$. Then the bottom $m - n$ rows of R are zero. Therefore we find c by solving the (nonsingular) linear system given by the first n equations of

$$Rc = Q^T y.$$

Similar (although more complicated) algorithm applies when $\operatorname{rank} R \le n - 1$. Note, recalling our former remark, that we don't require Q explicitly and need to evaluate only $Q^T y$.

7.4 Examples

Example 7.7 We wish to find $c \in \mathbb{R}^3$ that minimizes $\|Ac - y\|$, where

$$A = \tfrac{1}{2}\begin{bmatrix} 1 & 3 & 6 \\ 1 & 1 & 2 \\ 1 & 3 & 4 \\ 1 & 1 & 0 \end{bmatrix}, \qquad y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

We QR-factorize

$$A = \tfrac{1}{2}\begin{bmatrix} 1 & 3 & 6 \\ 1 & 1 & 2 \\ 1 & 3 & 4 \\ 1 & 1 & 0 \end{bmatrix} = \underbrace{\tfrac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}}_{Q} \times \underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{R}.$$

We thus need to solve the least-squares problem

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{R} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \approx \underbrace{\tfrac{1}{2}\begin{bmatrix} 3 \\ 1 \\ 1 \\ -1 \end{bmatrix}}_{Q^T y}.$$

No matter what the c_i's are, we cannot approach the fourth component of the right-hand side, but all others can be matched. So, we solve the 'top' square system

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \tfrac{1}{2}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \quad \Rightarrow \quad c^* = \tfrac{1}{2}\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \quad \|Ac^* - y\| = \tfrac{1}{2}.$$

7.5 Exercises

7.1. Let

$$A = \begin{bmatrix} 3 & 4 & 7 & -2 \\ 5 & 4 & 9 & 3 \\ 1 & -1 & 0 & 3 \\ 1 & -1 & 0 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 11 \\ 29 \\ 16 \\ 10 \end{bmatrix}.$$

Calculate the QR factorization of A by using Householder reflections. In this case A is singular and you should choose Q so that the last row of R is zero. Hence identify all the least squares solutions of the inconsistent system $Ax = b$, where we require $\boldsymbol{x}$ to minimize $\|Ax - b\|_2$. Verify that all the solutions give the same vector of residuals $Ax - b$, and that this vector is orthogonal to the columns of A. There is no need to calculate the elements of Q explicitly.