

## The Least Squares Method

### Data fitting by a straight line

Given the data:  $m+1$  points  $(x_0, y_0), (x_1, y_1), \dots, (x_m, y_m)$ . Suppose there are some reasons to believe that the underlying function is linear. Then we would like to use a straight line

$$y = ax + b$$

to fit the data. Usually we cannot find  $a$  and  $b$  such that the line passes through all  $m+1$  points. But we can require all points be as “close” to the line as possible. One of typical approaches is to solve the following optimization problem

$$\min_{a,b} \phi(a, b), \quad \phi(a, b) \equiv \sum_{k=0}^m (ax_k + b - y_k)^2.$$

This optimization problem is called a *least squares* problem.

From calculus, the conditions that

$$\frac{\partial \phi}{\partial a} = 0, \quad \frac{\partial \phi}{\partial b} = 0$$

are necessary at the minimum. Then from the expression of  $\phi(a, b)$ , we can easily obtain

$$\begin{bmatrix} \sum_{k=0}^m x_k^2 & \sum_{k=0}^m x_k \\ \sum_{k=0}^m x_k & m+1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^m x_k y_k \\ \sum_{k=0}^m y_k \end{bmatrix}.$$

The above equations are called the *normal equations*, which can easily be solved. In fact, it is easy to show

$$\begin{aligned} a &= \left[ (m+1) \sum_{k=0}^m x_k y_k - \left( \sum_{k=0}^m x_k \right) \left( \sum_{k=0}^m y_k \right) \right] / \left[ (m+1) \sum_{k=0}^m x_k^2 - \left( \sum_{k=0}^m x_k \right)^2 \right], \\ b &= \left[ \left( \sum_{k=0}^m x_k^2 \right) \left( \sum_{k=0}^m y_k \right) - \left( \sum_{k=0}^m x_k \right) \left( \sum_{k=0}^m x_k y_k \right) \right] / \left[ (m+1) \sum_{k=0}^m x_k^2 - \left( \sum_{k=0}^m x_k \right)^2 \right]. \end{aligned}$$

### Data fitting by a general linear families of functions

Suppose the data are thought to conform to a relationship like

$$y = \sum_{j=0}^n c_j g_j(x),$$

where the functions  $g_0, g_1, \dots, g_n$  (called basis functions) are known. Then we can determine the coefficients  $c_0, c_1, \dots, c_n$  by solving the least squares problem

$$\min_{c_0, c_1, \dots, c_n} \phi(c_0, c_1, \dots, c_n), \quad \phi(c_0, c_1, \dots, c_n) \equiv \sum_{k=0}^m \left( \sum_{j=0}^n c_j g_j(x_k) - y_k \right)^2.$$

Define

$$A = \begin{bmatrix} g_0(x_0) & g_1(x_0) & \cdots & g_n(x_0) \\ g_0(x_1) & g_1(x_1) & \cdots & g_n(x_1) \\ \vdots & \vdots & \cdots & \vdots \\ g_0(x_m) & g_1(x_m) & \cdots & g_n(x_m) \end{bmatrix}, \quad c = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_m \end{bmatrix}$$

So the least squares problem can also be written

$$\min_c \|Ac - y\|_2^2.$$

From the expression of  $\phi$ , we obtain

$$\frac{\partial \phi}{\partial c_i} = \sum_{k=0}^m 2 \left[ \sum_{j=0}^n c_j g_j(x_k) - y_k \right] g_i(x_k), \quad i = 0, 1, \dots, n.$$

In order to find the minimizer, we set  $\frac{\partial \phi}{\partial c_i} = 0$  for  $i = 0, 1, \dots, n$ , leading to the normal equations:

$$\sum_{j=0}^n \left[ \sum_{k=0}^m g_i(x_k) g_j(x_k) \right] c_j = \sum_{k=0}^m g_i(x_k) y_k, \quad i = 0, 1, \dots, n.$$

It is easy to verify that the above equations can be rewritten

$$A^T A c = A^T y. \tag{1}$$

It can be shown that (1) always has a solution, and the solution is unique if the columns of  $A$  are linearly independent. It can also be shown that if  $c$  satisfies (1), it must be the least squares solution. In principle, (1) can be solved by Gaussian elimination with no pivoting (GENP). In fact, better algorithms (which will be taught in COMP 540) are available for solving  $c$ .

**Remarks:** A special case is  $g_j(x) = x^j$  for  $j = 0, 1, \dots, n$ . If  $n = m$ , then the least square problem is just the polynomial interpolation problem.

MATLAB tools: built-in command for finding  $c$ : `c = A \ y`