CS3220 Lecture Notes: Linear least squares

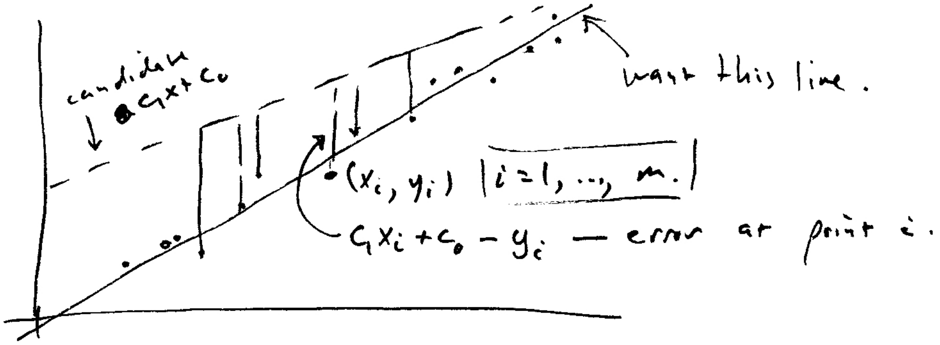
Steve Marschner
Cornell University

9 March 2009

One of the big applications of linear systems, right up there with Newton-type iterations, is solving least squares problems. Generally, these are problems where we want to fit a *model* to some *data* but we don't expect an exact match. This is unlike the problems we solved earlier in the course having to do with computing interpolating polynomials for a set of points, where we insisted on an exact match at all data points.

1 Linear regression

The canonical starting point for least squares is *linear regression*: you have data that you think would fit a linear model, except that it's become contaminated by some random errors (maybe they are rounding errors, or maybe these are physical measurements with their own uncertainty).


candidate c_1x+c_0
want this line.
(x_i, y_i) | $i=1, \dots, m.$
$c_1 x_i + c_0 - y_i$ — error at point i.

We want to minimize the distances from the points to the lines. I'll assume the x coordinates of the points are exact, so the distance is measured along the y direction. The difference between the model and the data,

$$r_i = c_1 x_i + c_o - y_i,$$

is called the "residual error" or just "residual" for point i. It's clear we are in the business of balancing these errors against one another. If we think of the residuals all together as a residual vector,

$$\mathbf{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix}.$$

Balancing errors against one another now boils down to choosing how to define the "size" of **r**—that is, the choice of norm. For instance, minimizing $\|\mathbf{r}\|_1$ corresponds to minimizing the sum-total deviation of the data from the model (close points count the same as as far points); $\|\mathbf{r}\|_\infty$ corresponds to maximum error (only the farthest point matters); and $\|\mathbf{r}\|_2$ is somewhere in between. The 2-norm is the universally popular choice—partly because it can be proven to be optimal under certain assumptions about the nature of the errors, but usually there is an ulterior motive: the 2-norm is mathematically convenient, as we'll see in a moment. Linear least squares (LLS) problems, in which the residuals are linear functions of the parameters, are especially easy to solve.

For the line fitting problem (the "linear" in LLS does not refer to this line!) we have:

$$E(c_1, c_0) = \|\mathbf{r}\|_2^2 = \sum_{i=1}^{m} (c_1 x_i + c_0 - y_i)^2$$

The error is a sum of squares (hence "least squares"). I've taken the liberty of defining the error to be the *square* of the 2-norm, since the smallest-norm residual also has the smallest squared norm.

To minimize, look for a stationary point (in this setting, the solution will turn out to be a minimum) by setting the two partial derivatives of E to zero:

$$\frac{\partial E}{\partial c_1} = \sum_{i=1}^{m} 2x_i \left(c_1 x_i + c_0 - y_i\right) = 2\left[c_i \sum_i x_i^2 + c_0 \sum_i x_i - \sum_i x_i y_i\right] = 0$$
$$\frac{\partial E}{\partial c_0} = \sum_{i=1}^{m} 2\left(c_1 x_i + c_0 - y_i\right) = 2\left[c_i \sum_i x_i + c_0 m - \sum_i y_i\right] = 0$$

We can recognize this as a 2x2 linear system in the variables c_1 and c_0! That we know how to solve already.

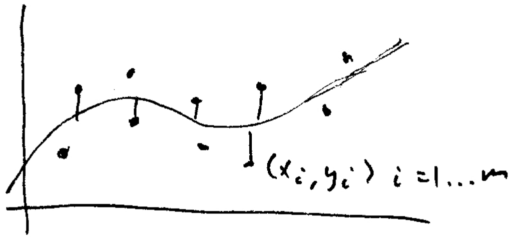
In matrix form, this is

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & m \end{bmatrix} \begin{bmatrix} c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix} \tag{1}$$

So this LLS problem reduced to a linear system—easy! There is a unique solution (unless the matrix is singular), it is easy to find, and it is always a minimum.

2 Cubic regression

Let's do this again but with a polynomial model this time: I want to fit a cubic to m data points, with $m > 4$, so unlike earlier in the course we don't expect an exact solution.



We saw before that fitting high degree polynomials to a lot of points is a dicey business; if we want a smooth curve it's better to use a low-degree model and accept some discrepancy.

Now the problem looks like this:

$$r_i = c_3 x_i^3 + c_2 x_i^2 + c_1 x_i + c_0 - y_i$$

and the system that results from setting the four partial derivatives to zero is:

$$E(c_3, \ldots, c_0) = \sum_{i=1}^{m} \left(c_3 x_i^3 + c_2 x_i^2 + c_1 x_i + c_0 - y_i\right)^2$$

$$\begin{aligned}
\frac{\partial E}{\partial c_3} &= 2 \sum_{i=1}^{m} x_i^3 \left(c_3 x_i^3 + c_2 x_i^2 + c_1 x_i + c_0 - y_i\right) \\
&= 2 \left[c_3 \sum x_i^6 + c_2 \sum x_i^5 + c_1 \sum x_i^4 + c_0 \sum x_i^3 - \sum x_i^3 y_i\right] = 0 \\
\frac{\partial E}{\partial c_2} &= 2 \left[c_3 \sum x_i^5 + c_2 \sum x_i^4 + c_1 \sum x_i^3 + c_0 \sum x_i^2 - \sum x_i^2 y_i\right] = 0 \\
\frac{\partial E}{\partial c_1} &= 2 \left[c_3 \sum x_i^4 + c_2 \sum x_i^3 + c_1 \sum x_i^2 + c_0 \sum x_i - \sum x_i y_i\right] = 0 \\
\frac{\partial E}{\partial c_0} &= 2 \left[c_3 \sum x_i^3 + c_2 \sum x_i^2 + c_1 \sum x_i + c_0 m - \sum y_i\right] = 0
\end{aligned}$$

Lo and behold, it is another linear system, this time a 4 by 4 system. The higher powers of x didn't enter in at all: it is only the values of these functions at the data points that matter, and the minimization is with respect to the c_is, which always appear linearly. (Note, by the way, the copy of the matrix for linear regression sitting in the lower right corner of this system).

In this case the system looks like this:

$$\begin{bmatrix} \sum x_i^6 & \sum x_i^5 & \sum x_i^4 & \sum x_i^3 \\ \sum x_i^5 & \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^4 & \sum x_i^3 & \sum x_i^2 & \sum x_i^1 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i^1 & \sum x_i^0 \end{bmatrix} \begin{bmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \sum x_i^3 y_i \\ \sum x_i^2 y_i \\ \sum x_i^1 y_i \\ \sum x_i^0 y_i \end{bmatrix} \tag{2}$$

3 Normal equations

Each of these two systems can be expressed more compactly. First the linear regression system: the original problem is

$$\min_{\mathbf{c}} \|\mathbf{A}\mathbf{c} - \mathbf{y}\|_2^2$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{x} & \mathbf{1} \end{bmatrix}$$
$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_0 \end{bmatrix}$$

and $\mathbf{x}$ (resp. $\mathbf{y}$) is the vector of all the x_is (resp. y_is) and $\mathbf{1}$ is a vector of all ones (ones(m,1)).

With these definitions, you can easily verify that the linear system in (1) is

$$\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{y}.$$

The same is also true of the larger system in (2). In that system:

$$\mathbf{A} = \begin{bmatrix} \mathbf{v}_3 & \mathbf{v}_2 & \mathbf{x} & \mathbf{1} \end{bmatrix}$$
$$\mathbf{c} = \begin{bmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

where, in Matlab notation, $\mathbf{v}_j = \mathbf{x}$.* j. You can easily verify that the linear system that resulted from setting the partials to zero in that case also has the form $\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{y}$.

This linear system is known as the *normal equations* for the least squares system $\mathbf{A}\mathbf{c} \approx \mathbf{y}$.

Sources

- Our textbook: Cheney & Kincaid, *Numerical Mathematics and Computing*, 6e. Section 12.1.