Least Squares Solutions

Before we consider the general case, consider the following problem:

Given p data points, $\{(x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p)\}$, find the line $y = mx + b$ that **best** fits the data.

The first question we must answer is what we mean by "best". In this example, we will assume that the data does not actually fit on a line due to errors in the $y-$values- the $x-$values are precise.

Therefore, given a particular m, b, we will define the error to be:

$$E(m, b) = (y_1 - (mx_1 + b))^2 + (y_2 - (mx_2 + b))^2 + \ldots + (y_p - (mx_p + b))^2$$

That is, this is the "sum of squares" error between our *desired* output, $y_1, \ldots, y_p$, and our *model output*, $mx_i + b$. The error can be written compactly as:

$$E(m, b) = \sum_{k=1}^{p} (y_k - mx_k - b)^2$$

and note that E depends only on m, b (the x's and y's are numbers that are known already).

We can find our solution from Calculus- in fact, we could plot the graph of E using Maple. From Calculus, we know that to compute the minimum of E, we set the partial derivatives to zero and solve for m, b:

$$\frac{\partial E}{\partial m} = 2(y_1 - mx_1 - b)(-x_1) + 2(y_2 - mx_2 - b)(-x_2) + \ldots + 2(y_p - mx_p - b)(-x_p)$$

or more compactly:

$$\frac{\partial E}{\partial m} = \sum_{k=1}^{p} 2(y_k - mx_k - b)(-x_k) = \sum_{k=1}^{p} 2(-y_k x_k + mx_k^2 + bx_k) = 2\left(-\sum_{k=1}^{p} y_k x_k + m\sum_{k=1}^{p} x_k^2 + b\sum_{k=1}^{p} x_k\right)$$

Setting this to zero we could write this as (the expressions in parentheses are numbers computed from the data):

$$m\left(\sum_{k=1}^{p} x_k^2\right) + b\left(\sum_{k=1}^{p} x_k\right) = \left(\sum_{k=1}^{p} x_k y_k\right)$$

Similarly, compute $\frac{\partial E}{\partial b}$ and set it equal to zero:

$$\frac{\partial E}{\partial b} = \sum_{k=1}^{p} 2(y_k - mx_k - b)(-1) = 2\left(-\sum_{k=1}^{p} y_k + m\sum_{k=1}^{p} x_k + b\sum_{k=1}^{p} 1\right) = 0$$

so that

$$m\left(\sum_{k=1}^{p} x_k\right) + b \cdot p = \left(\sum_{k=1}^{p} y_k\right)$$

Now we have two equations in two unknowns. Writing these equations in matrix form, and simplifying our summation notation, we get:

$$\begin{bmatrix} \sum x_k^2 & \sum x_k \\ \sum x_k & p \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum x_k y_k \\ \sum y_k \end{bmatrix}$$

We can also write the equation in terms of linear algebra. In this setting, the data can be written as p equations in m, b which can be translated into a matrix-vector equation:

$$\begin{array}{rl} mx_1 + b & = y_1 \\ mx_2 + b & = y_2 \\ \vdots & \vdots \\ mx_p + b & = y_p \end{array} \quad \Rightarrow \quad \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_p & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} \text{ or } A\mathbf{x} = \mathbf{b}$$

To motivate what we do next, consider that $\mathbf{b}$ is not contained in the columnspace of A (if it were, a solution would exist to $A\mathbf{x} = \mathbf{b}$), therefore we can write $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$, where $\mathbf{b}_1 \in \text{Col}(A)$ and $\mathbf{b}_2 \in \text{Null}(A^T)$.

$$A\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad A^TA\mathbf{x} = A^T\mathbf{b} = A^T\mathbf{b}_1 + A^T\mathbf{b}_2 = A^T\mathbf{b}_1 \quad \Rightarrow \quad A^TA\mathbf{x} = A^T\mathbf{b}$$

The middle equality shows that multiplication by A^T will remove that part of $\mathbf{b}$ that is in the nullspace of A^T, and the last equality is *the normal equation.* Note that this is a 2×2 system of equations in m, b. Rewriting it, we get:

$$A^TA = \begin{bmatrix} x_1 & x_2 & \dots & x_p \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_p & 1 \end{bmatrix} = \begin{bmatrix} \sum x_k^2 & \sum x_k \\ \sum x_k & p \end{bmatrix}$$

Thus, the normal equations are exactly the same as the equations we got from setting the partial derivatives of our error to zero.

To see why this is so, consider that the matrix equation had no solution, $\mathbf{x}$. Therefore, we are trying to find an approximate solution, $\hat{\mathbf{x}}$, so that the difference between the given vector $\mathbf{b}$ and our approximate $\hat{\mathbf{b}} = A\hat{\mathbf{x}}$ is as small as possible. That is,

Find $\hat{\mathbf{x}}$ so that we minimize the following: $\|\mathbf{y} - A\mathbf{x}\|^2$

If you were to write this expression out componentwise, you would see that it is exactly $E(m, b)$.

We can generalize this technique to other models. Here is the primary definition:

Definition: A model equation is *linear* if it is linear in its parameters.

EXAMPLES (the vector $\mathbf{c}$ will represent model parameters):

$$f(x, \mathbf{c}) = c_0 + c_1 x + c_2 x^2 + \ldots + c_n x^n$$

$$f(x, y, \mathbf{c}) = c_1 \mathrm{e}^{x} + c_2 \mathrm{e}^{-y}$$

$$f(x, \mathbf{c}) = c_1 \sin(x) + c_2 \cos(x) + c_3 \sin(2x) + c_4 \cos(2x)$$

NOT EXAMPLES (these are nonlinear models):

$$f(x, \mathbf{c}) = c_1 \sin(c_2 x) + c_3 \cos(c_4 x)$$

$$f(x, \mathbf{c}) = c_1 \mathrm{e}^{c_2 x} + c_3 \mathrm{e}^{-c_4 x}$$

If you are using a linear model, then you can always express the problem as a matrix-vector equation with the parameter vector $\mathbf{c}$ as the unknown. To be more specific, if you have k parameters to solve for, and p data points, then the matrix A will be $p \times k$. Here are some examples using the ordered pairs $(1, -1), (2, 0), (3, 1), (-1, 0)$:

If $f(x, \mathbf{c}) = c_0 + c_1 x + c_2 x^2$, then construct $A\mathbf{c} = \mathbf{y}$, where:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & -1 & 1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

If $f(x, \mathbf{c}) = c_0 + c_1 \sin(x) + c_2 \cos(x)$, then construct $A\mathbf{c} = \mathbf{y}$, where:

$$A = \begin{bmatrix} 1 & \sin(1) & \cos(1) \\ 1 & \sin(2) & \cos(2) \\ 1 & \sin(3) & \cos(3) \\ 1 & \sin(-1) & \cos(-1) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Using Matlab notation, these last two examples would be:

```
x=[1; 2; 3; -1]; y=[-1; 0; 1; 0];
A=[ones(4,1), x, x.^2];
c=A\y;
%Plot the results using the model over extra pts
X=linspace(-1.5,3.5);
Y=c(1)+c(2)*X+c(3)*X.^2;
plot(x,y,'r*',X,Y);

%Second Model:
A=[ones(4,1), sin(x), cos(x)];
c=A\y;
%Plot the results using the model over extra pts
X=linspace(-1.5,3.5);
Y=c(1)+c(2)*sin(X)+c(3)*cos(X);
plot(x,y,'r*',X,Y);
```