1 Linear Least Squares

I will try to be consistent in notation, with

- n being the number of data points, and
- $m < n$ being the number of parameters in a model function.

We are interested in solving an inconsistent set of equations, $\mathbf{Ax} = \mathbf{b}$, where

$\mathbf{A}$ is an $n \times m$-matrix with components $a_{i,j}$,
$\mathbf{x}$ is an m-vector with $m < n$ and components x_i,
$\mathbf{b}$ is an n-vector with components b_i.

Notice that if $m = n$ this could be solved (providing a solution exists) using Gaussian elimination.

This system is inconsistent (has no solution) since the number of unknown parameters m is less than the number of equations n and it is unlikely that $\mathbf{x}$ can be chosen such that $\mathbf{Ax} = \mathbf{b}$.

If $m = n$, we would want to determine $\mathbf{x}$ such that $\|\mathbf{Ax}-\mathbf{b}\|^2 = 0$ (where we will work with the Euclidean norm).

Goal: If $m < n$, we would want to determine $\mathbf{x}$ such that $\|\mathbf{Ax} - \mathbf{b}\|^2$ is minimized.

Notation: Given $\mathbf{A} = [a_{i,j}]_{n\times m}$, we define the transpose of $\mathbf{A}$ as $\mathbf{A}^T = [a_{j,i}]_{m\times n}$.

Useful results from linear algebra:

$$(\mathbf{A}+\mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$$
$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$
$$\|\mathbf{x}-\mathbf{y}\|^2 = \sum_{i=1}^{n}(x_i - y_i)^2 = (\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})$$
$$i\text{th row of } \mathbf{Ax} \text{ is } \sum_{k=1}^{n} a_{i,k}x_k$$
$$i\text{th row of } \mathbf{A^T b} \text{ is } \sum_{k=1}^{n} a_{k,i}b_k$$

Definition: Two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$ are orthogonal if $\mathbf{x}^T\mathbf{y} = 0$

I want to attack this in a different manner than the text, using directional derivatives.

$$\begin{aligned}\|\mathbf{Ax}-\mathbf{b}\|^2 &= (\mathbf{Ax}-\mathbf{b})^T(\mathbf{Ax}-\mathbf{b})\\ &= ((\mathbf{Ax})^T - \mathbf{b}^T)(\mathbf{Ax}-\mathbf{b})\\ &= (\mathbf{x}^T\mathbf{A}^T - \mathbf{b}^T)(\mathbf{Ax}-\mathbf{b})\\ &= \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - \mathbf{b}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}\end{aligned}$$

Now, each term in the above is a scalar, since the norm is a scalar. For this to be a minimum, the directional derivative in any direction $\mathbf{u}$ must be zero (from Calculus III).

$$D_{\mathbf{u}}\|\mathbf{Ax}-\mathbf{b}\|^2 = \frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - \mathbf{b}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}\right)\cdot\mathbf{u} = 0$$

where[1]

$$\frac{\partial \alpha}{\partial \mathbf{x}} = \left(\frac{\partial \alpha}{\partial x_1} \cdots \frac{\partial \alpha}{\partial x_j} \cdots \frac{\partial \alpha}{\partial x_m} \right)^T$$

Let's work out each of these derivatives in turn, starting with the easy ones, and looking at the jth row $\frac{\partial \alpha}{\partial x_j}$.

We will use $\frac{\partial x_i}{\partial x_j} = \delta_{ij}$.

$$\alpha = \mathbf{b}^T\mathbf{b} = \sum_{k=1}^{n} b_i^2$$

$$\frac{\partial}{\partial x_j}(\alpha) = 0 \qquad \longrightarrow \qquad \frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^T\mathbf{b}) = \mathbf{0}$$

$$\alpha = \mathbf{x}^T\mathbf{A}^T\mathbf{b} = \sum_{i=1}^{n} x_i \sum_{k=1}^{n} a_{k,i} b_k$$

$$\frac{\partial}{\partial x_j}(\alpha) = \sum_{k=1}^{n} a_{k,j} b_k = \sum_{k=1}^{n} (a_{j,k})^T b_k \quad (j\text{th row of } \mathbf{A}^T\mathbf{b}) \qquad \longrightarrow \qquad \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T\mathbf{A}^T\mathbf{b}) = \mathbf{A}^T\mathbf{b}$$

$$\alpha = \mathbf{b}^T\mathbf{A}\mathbf{x} = \sum_{i=1}^{n} b_i \sum_{k=1}^{n} a_{i,k} x_k$$

$$\frac{\partial}{\partial x_j}(\alpha) = \sum_{i=1}^{n} b_i a_{i,j} = \sum_{i=1}^{n} (a_{j,i})^T b_i \quad (j\text{th row of } \mathbf{A}^T\mathbf{b}) \qquad \longrightarrow \qquad \frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^T\mathbf{A}\mathbf{x}) = \mathbf{A}^T\mathbf{b}$$

$$\alpha = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = (\mathbf{A}\mathbf{x})^T\mathbf{A}\mathbf{x} = \left(\sum_{k=1}^{n} a_{i,k} x_k\right)\left(\sum_{q=1}^{n} a_{i,q} x_q\right) \quad (\text{product of } i\text{th row of } \mathbf{A}\mathbf{x})$$

$$\begin{aligned}
\frac{\partial}{\partial x_j}(\alpha) &= a_{i,j}\left(\sum_{q=1}^{n} a_{i,q} x_q\right) + \left(\sum_{k=1}^{n} a_{i,k} x_k\right) a_{i,j} \quad (\text{product rule of derivatives}) \\
&= 2\sum_{k=1}^{n} a_{i,j} a_{i,k} x_k \\
&= 2\sum_{k=1}^{n} (a_{j,i})^T a_{i,k} x_k = 2 \quad (j\text{th row of } \mathbf{A}^T\mathbf{A}\mathbf{x}) \qquad \longrightarrow \qquad \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}) = 2\mathbf{A}^T\mathbf{A}\mathbf{x}
\end{aligned}$$

Substituting these results back, we find that

$$\left(2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b}\right) \cdot \mathbf{u} = 0$$

[1]Note this is the *denominator layout* convention which you can learn more about here: http://en.wikipedia.org/wiki/Matrix_calculus

and for this to be true for all directions $\mathbf{u}$, we must have

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b} \qquad \text{(the normal equations)}$$

The solution $\mathbf{x}$ to the normal equations will minimize the value of the residual $\|\mathbf{r}\| = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$.

You can solve the normal equations using Guassian elimination, or finding an inverse of a matrix:

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$$

where $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ is the pseudoinverse of the matrix $\mathbf{A}$.

Fitting Models to Data

If we are given $1 \le i \le n$ data points (x_i, y_i) and have a linear model we wish to fit

$$f(x) = \sum_{j=1}^{m} c_j f_j(x)$$

where $m < n$ then we can use the normal equations to determine the values of the parameters c_j that best fit the data. "Best fit" is taken to mean minimizing the quantity $\|\mathbf{A}\mathbf{c} - \mathbf{b}\|$, where substituting the data points into the model results in the equations:

$$f(x_i) = y_i$$

which is written as $\mathbf{A}\mathbf{c} = \mathbf{b}$.

You should note that this process is actually quite flexible, and you can construct whatever model might best fit the data, even something like:

$$f(x, y) = c_1 + c_2 \cos x + c_3 \sin y + c_4 xy$$

You simply use this equation to create the associated normal equations for the related system $\mathbf{A}\mathbf{c} = \mathbf{b}$. The process is linear in terms of being linear in the parameters c_i that are being determined, not in terms of a linear model function f.

Section 4.2 shows quite a few models, so make sure to read it.

Improvements When Solving The Normal Equations Fail: QR Algorithm

Sometimes solving the normal equations fails, since the Gaussian elimination is not successful. In those cases it is useful to employ the QR factorization.

Gram-Schmidt Orthogonalization

I think the text switched the meaning of m and n here, so I will maintain my idea that n is the number of data points at $m < n$ is the number of parameters (this is how the *Mathematica* notebook is set up).

Gram-Schmidt orthogonalization is used to take a set of m linearly independent vectors and create a new set of m vectors that is orthogonalized (each vector is perpendicular to all the other vectors). The algorithm is fairly straightforward to element, and is discussed in the *Mathematica* file. The full QR decomposition is

$$\mathbf{A}_{[n\times m]} = \mathbf{Q}_{[n\times n]}\mathbf{R}_{[n\times m]}$$

where $\mathbf{Q}^T = \mathbf{Q}^{-1}$ (this is why we need the full decomposition, so the inverse matrix exists since the inverse is only defined for square matrices).

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}^{-1}\mathbf{Q} = I$$

Once we have the full QR decomposition of $\mathbf{A}$ we do linear least squares on the $m \times n$ system $\mathbf{Ac} = \mathbf{b}$ for which we wish $\|\mathbf{Ac} - \mathbf{b}\|^2$ to be a minimum in the following manner:

$$\begin{aligned}
&\mathbf{Ac} = \mathbf{b} \\
&\mathbf{QRc} = \mathbf{b} \\
&\mathbf{Q}^T\mathbf{QRc} = \mathbf{Q}^T\mathbf{b} && \text{(left multiply by } \mathbf{Q}^T) \\
&\mathbf{Rc} = \mathbf{Q}^T\mathbf{b} && \text{(since } \mathbf{Q}^T\mathbf{Q} = I) \\
&\widehat{\mathbf{R}}\mathbf{c} = \widehat{\mathbf{d}}
\end{aligned}$$

where $\widehat{\mathbf{R}}$ is the upper $m \times m$ part of $\mathbf{R}$ and $\widehat{\mathbf{d}}$ is the upper m entries of $\mathbf{Q}^T\mathbf{b}$.

Solve this system for the model parameters $\mathbf{c}$.

2 Nonlinear Regression: Gauss-Newton Method

If the model function does not linearly depend on the parameters, then the previous linear algebra based methods will not work. We can examine the problem instead from the point of view of minimizing the error by taking the gradient.

General Set Up

We have n data points (x_i, y_i), $i = 1, 2, \ldots, n$ which we want to fit to a model function which has m adjustable parameters α_j, , $j = 1, 2, \ldots, m$: $f(x) = f(x; \alpha_1, \ldots, \alpha_n)$.

We actually have a great deal of choice in what type of function we want to minimize. It can be anything that will measure the relation of the data to the model function. The vector which compares the data to the model function at each point is given by

$$\begin{pmatrix} y_1 - f(x_1; \alpha_1, \ldots, \alpha_m) \\ y_2 - f(x_2; \alpha_1, \ldots, \alpha_m) \\ \vdots \\ y_n - f(x_n; \alpha_1, \ldots, \alpha_m) \end{pmatrix}$$

We can minimize this vector based on a variety of different norms:

$$\begin{array}{rl}
l_1 \text{ norm:} & \displaystyle\sum_{i=1}^{n} |y_i - f(x_i;\alpha_1,\dots,\alpha_m)| \\
l_p \text{ norm:} & \displaystyle\left(\sum_{i=1}^{n} (y_i - f(x_i;\alpha_1,\dots,\alpha_m))^p\right)^{1/p} \\
l_\infty \text{ norm:} & \displaystyle\max_{i=1,..,n} (y_i - f(x_i;\alpha_1,\dots,\alpha_m))
\end{array}$$

What is typically done is that the l_2 norm is used, since it is the Euclidean space norm, and the *square* of the norm is minimized:

$$\text{minimize } E(\alpha_1,\dots,\alpha_m) = \sum_{i=1}^{n} (y_i - f(x_i;\alpha_1,\dots,\alpha_m))^2$$

Minimizing is just a multivariable unconstrained minimization procedure, which yields the system of equations

$$0 = \sum_{i=1}^{n} (y_i - f(x_i;\alpha_1,\dots,\alpha_m)) \left(\frac{\partial}{\partial \alpha_k} f(x_i;\alpha_1,\dots,\alpha_m)\right), \quad k = 1,\dots,m \tag{1}$$

which must be solved for the m unknowns α_j. You can solve the equations numerically using Newton's method for systems.

Applying this technique to Linear Regression

For now we are interested in fitting to a polynomial of degree m (which means we have $m+1$ parameters).

$$f(x;\alpha_0,\dots,\alpha_m) = \sum_{j=0}^{m} \alpha_j x^j$$

The system of equations we solve becomes

$$\begin{aligned}
0 &= \sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{m}\alpha_j x_i^j\right)\left(\frac{\partial}{\partial\alpha_k}\sum_{j=0}^{m}\alpha_j x_i^j\right), \quad k = 0,\dots,m \\
0 &= \sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{m}\alpha_j x_i^j\right)\left(\sum_{j=0}^{m}\delta_{jk} x_i^j\right), \quad k = 0,\dots,m \\
0 &= \sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{m}\alpha_j x_i^j\right) x_i^k, \quad k = 0,\dots,m \\
0 &= \sum_{i=1}^{n}\left(y_i x_i^k - \sum_{j=0}^{m}\alpha_j x_i^j x_i^k\right), \quad k = 0,\dots,m \\
\sum_{j=0}^{m}\alpha_j \sum_{i=1}^{n} x_i^{j+k} &= \sum_{i=1}^{n} y_i x_i^k, \quad k = 0,\dots,m
\end{aligned}$$

This is a system of $m+1$ equations in the $m+1$ unknowns α_j. It is a linear system, and can be solved using Cramer's rule, resulting in the fits you may have seen before (especially for $f(x) = \alpha_0 + \alpha_1 x$).

3 Aside: Orthogonal Polynomials and Approximating Functions

Now, we consider having a function $f(x)$ and fitting a curve $P(x;\alpha_0,\ldots,\alpha_n)$ to the function.

We want to minimize the error between the two, and again choose the least squares approximation:

$$\text{minimize } E(\alpha_0,\ldots,\alpha_n) = \int_a^b w(t)\Big(f(t) - P(t;\alpha_0,\ldots,\alpha_n)\Big)^2\,dt$$

where $w(t)$ is a weight function. The weight function gives greater weight to certain sections of the region that it is defined on.

Minimizing this equation is just a multivariable unconstrained minimization procedure, which yields

$$\frac{\partial E(\alpha_0,\ldots,\alpha_n)}{\partial \alpha_k} = 0 = 2\int_a^b w(t)\Big(f(t) - P(t;\alpha_0,\ldots,\alpha_n)\Big)\frac{\partial}{\partial \alpha_k}P(t;\alpha_0,\ldots,\alpha_n)\,dt, \quad k=0,\ldots,n$$

We choose the function $P(t;\alpha_0,\ldots,\alpha_n) = \sum_{j=0}^n \alpha_j\phi_j(t)$, where $\phi_j(t)$ is a set of linearly independent functions on $[a,b]$. Therefore, we get

$$\begin{aligned}
0 &= \int_a^b w(t)\Big(f(t) - \sum_{j=0}^n \alpha_j\phi_j(t)\Big)\frac{\partial}{\partial \alpha_k}\sum_{j=0}^n \alpha_j\phi_j(t)\,dt, \quad k=0,\ldots,n\\
0 &= \int_a^b w(t)\Big(f(t) - \sum_{j=0}^n \alpha_j\phi_j(t)\Big)\sum_{j=0}^n \delta_{kj}\phi_j(t)\,dt, \quad k=0,\ldots,n\\
0 &= \int_a^b w(t)\Big(f(t) - \sum_{j=0}^n \alpha_j\phi_j(t)\Big)\phi_k(t)\,dt, \quad k=0,\ldots,n\\
\int_a^b w(t)f(t)\phi_k(t)\,dt &= \sum_{j=0}^n \alpha_j\int_a^b w(t)\phi_j(t)\phi_k(t)\,dt, \quad k=0,\ldots,n
\end{aligned}$$

If the functions $\phi_k(t)$ can be chosen so that they are orthogonal,

$$\int_a^b w(t)\phi_j(t)\phi_k(t)\,dt = \begin{cases} 0 & j\neq k\\ \beta_j & j=k\end{cases}$$

then we get

$$\begin{aligned}
\int_a^b w(t)f(t)\phi_k(t)\,dt &= \sum_{j=0}^n \alpha_j\int_a^b w(t)\phi_j(t)\phi_k(t)\,dt, \quad k=0,\ldots,n\\
&= \sum_{j=0}^n \alpha_j\delta_{jk}\beta_j, \quad k=0,\ldots,n\\
&= \alpha_k\beta_k, \quad k=0,\ldots,n\\
\alpha_k &= \frac{1}{\beta_k}\int_a^b w(t)f(t)\phi_k(t)\,dt
\end{aligned}$$

and we have determined the α_j which minimize the function.

The Gram-Schmidt process describes how to construct orthogonal polynomials given a specific weight function $w(x)$.