# Matrix MLE for Linear Regression

**Joseph E. Gonzalez**

Some people have had some trouble with the linear algebra form of the MLE for multiple regression. I tried to find a nice online derivation but I could not find anything helpful. So I have decide to derive the matrix form for the MLE weights for linear regression under the assumption of Gaussian noise.

## The Model

Lets say we are given some set of data $X$ and $y$. The matrix $X$ has $n$ rows corresponding to each of the examples and $d$ columns corresponding to each of the $d$ features. The column vector $y$ consists has $n$ rows corresponding to each of the examples and 1 column. We want to "learn" the relationship between an individual feature vector $x$ and an outcome $y$. In some sense we want to learn the function $f : \mathbb{R}^d \to \mathbb{R}$ which satisfies:

$$y = f(x) \tag{1}$$

### Linear Models

There are many functions $f$ that we could chose from (I am sure you have some favorites). To simplify our computation and to impose some assumptions (which often aids in generalization) we will restrict $f$ to the class of linear functions. That is for a choice of weights $w$ we can express $f$ as:

$$f_w(x) = \sum_{j=1}^{d} w_j x_j \tag{2}$$

### Nonlinear Features

Often people find this assumption to restrictive. We can permit a more complex class of functions by creating new (nonlinear) features from the original features $x_j$. For example:

$$f_w(x) = \sum_{j=1}^{d} w_j x_j + \sum_{j=d+1}^{2d} w_j \, \text{Sin}\left[x_j^2\right] \tag{3}$$

To formalize this notion we can rewrite equation 3 as:

$$f_w(x) = \sum_{j=1}^{m} w_j \phi_j[x] \tag{4}$$

Returning to the example in equation 3 we can use the notation of equation 4 by defining:

$$\phi_j[x] = \begin{cases} x_j & \text{if } 1 \le j \le d \\ \text{Sin}\left[x_j^2\right] & \text{if } d+1 \le j \le 2\,d \\ 0 & \text{otherwise} \end{cases}$$

This technique allows us to lift our simple linear function $f_w$ into a more complex space permitting a richer class of

functions in our original space $\mathbb{R}^d$. With this transformation we can define a matrix $\Phi$ which is like $X$ but consists of the transformed features. If we do not want to transform our features then we simply define:

$$\phi_j[x] = \begin{cases} x_j & \text{if } 1 \le j \le d \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The matrix $\Phi$ is constructed by:

$$\Phi = \begin{pmatrix} \phi_1[X_{11}, \ldots, X_{1d}] & \ldots & \phi_m[X_{11}, \ldots, X_{1d}] \\ \ldots & \ldots & \ldots \\ \phi_1[X_{n1}, \ldots, X_{nd}] & \ldots & \phi_m[X_{n1}, \ldots, X_{nd}] \end{pmatrix} \tag{6}$$

If we use the trivial transform in equation 5 equation 6 becomes:

$$\Phi = \begin{pmatrix} X_{11} & \ldots & X_{1d} \\ \ldots & \ldots & \ldots \\ X_{n1} & \ldots & X_{nd} \end{pmatrix} = X \tag{7}$$

For the rest of these notes I will use the trivial feature space $X$. However feel free to substitute $\Phi$ where ever $X$ is used if a nonlinear feature space is desired.

### Noise

Sadly we live in the real world where there is random noise $\epsilon$ that gets mixed into our observations. So a more natural model would be of the form:

$$y = f_w(x) + \epsilon \tag{8}$$

We have to pick what type of noise we expect to observe. A common choice is 0 mean independent gaussian noise of the form:

$$\epsilon \sim N(0, \sigma)$$

### Which $f_w$

Having selected the feature transformation $\phi$ and having decided to use a linear model we have reduced our hypothesis space (the space of functions we are willing to consider for $f$) from all the functions (and then some) to linear functions in the feature space determined by $\phi$. The functions in this space are indexed by $w$ (the weight vector). How do we pick $f$ from this reduced hypothesis space? We simply choose the "best" $w$. For the remainder of these notes we will be describing how to choose the $w$ that maximizes the likelihood of our data $X$ and y.

# Matrix Notation

Lets begin with some linear algebra. We can apply our model to the data in the following ways:

$$y = \begin{pmatrix} y_1 \\ \ldots \\ y_n \end{pmatrix} = \begin{pmatrix} f_w(< X_{11}, \ldots, X_{1d} >) + \epsilon_1 \\ \ldots \\ f_w(< X_{n1}, \ldots, X_{nd} >) + \epsilon_n \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{d} w_j X_{1j} + \epsilon_1 \\ \ldots \\ \sum_{j=1}^{d} w_j X_{nj} + \epsilon_n \end{pmatrix} = X\, w + \epsilon \tag{9}$$

where $w$ is a $d \times 1$ column vector of weights and $\epsilon$ is a $d \times 1$ column vector of iid $\epsilon_i \sim N(0, \sigma)$ gaussian noise. Notice how we can compactly compute all the $y$ at once by simply multiplying $X\, w$. If we solve for the noise in equation 9 we obtain:

$$y - X\, w = \epsilon \sim N(0, \sigma I);$$

$$(y - X\,w) \sim N(0,\ \sigma\,I); \tag{10}$$

We see that the residual of our regression model follows a multivariate gaussian with covariance $\sigma\,I$ were $I$ is the identity matrix. The density of the multivariate Gaussian takes the form:

$$p(V) = \frac{1}{(2\,\pi)^{N/2}\,|\,\Sigma\,|^{1/2}}\,\text{Exp}\Big[-\frac{1}{2}\,(V-\mu)^{\mathrm{T}}\,\Sigma^{-1}(V-\mu)\Big] \tag{11}$$

where $V \sim N(\mu,\ \Sigma)$ and $V \in \mathbb{R}^{N\times 1}$ is a column vector of size $N$.

# Likelihood

Using equation 10 and 11 we can express the likelihood of our data given our weights $w$ as:

$$P(X,\ y\,|\,w)\ \propto L(w) \propto \text{Exp}\Big[-\frac{1}{2}\,(y - X\,w)^{\mathrm{T}}\,\frac{1}{\sigma}\,I\,(y - X\,w)\Big]$$

We now want to maximize the likelihood of our data given the weights. First we take the Log to make thinks easier

$$l(w)\ \propto (y - X\,w)^{\mathrm{T}}\,I\,(y - X\,w)$$

Notice that we can remove any additional multiplicative constants. We now have

$$l(w) \propto \underbrace{(y - X\,w)^{\mathrm{T}}}_{\text{row vector}}\,\underbrace{\_\_I\_\_}_{\text{identity Matrix}}\,\underbrace{(y - X\,w)}_{\text{col vector}}$$

You should be able to convince yourself that this is equivalent to:

$$l(w) \propto (y - X\,w)^{\mathrm{T}}\,(y - X\,w)$$

Now lets take the gradient (row vector) derivative with respect to $w$:

$$\frac{\partial}{\partial w}\,l(w) \propto \frac{\partial}{\partial w}[(y - X\,w)^{\mathrm{T}}\,(y - X\,w)]$$

To compute this we will use the gradient of a quadratic matrix equation.
For more details see http://en.wikipedia.org/wiki/Matrix_calculus
http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/calculus.html#deriv_quad

$$\frac{\partial}{\partial w}\,l(w) \propto -(y - X\,w)^{\mathrm{T}}\ X - (y - X\,w)^{\mathrm{T}}\,X$$

Simplifying a little

$$\frac{\partial}{\partial w}\,l(w) \propto -2\,(y - X\,w)^{\mathrm{T}}\,X$$

Removing extraneous constants

$$\frac{\partial}{\partial w}\,l(w) \propto -(y - X\,w)^{\mathrm{T}}\,X$$

Apply the tranpose

$$\frac{\partial}{\partial w} l(w) \propto -(y^{\mathrm{T}} - w^{\mathrm{T}} X^{\mathrm{T}}) X$$

Multiplying through by $X$:

$$\frac{\partial}{\partial w} l(w) \propto -y^{\mathrm{T}} X + w^{\mathrm{T}} X^{\mathrm{T}} X$$

Finally we set the derivative equal to zero and solve for $w$ to obtain:

$$\frac{\partial}{\partial w} l(w) \propto -y^{\mathrm{T}} X + w^{\mathrm{T}} X^{\mathrm{T}} X = 0$$

$$w^{\mathrm{T}} X^{\mathrm{T}} X = y^{\mathrm{T}} X$$

$$w^{\mathrm{T}} = y^{\mathrm{T}} X (X^{\mathrm{T}} X)^{-1}$$

Finally remvoing the transpose we have:

$$w = (X^{\mathrm{T}} X)^{-1} X^{\mathrm{T}} y$$

Thus you have the matrix form of the MLE.