Multiple Linear Regression

1 Multiple Linear Regression

Multiple linear regression is a method used to model the relationship between multiple input features $x_1, x_2, \ldots, x_D$ and one output feature y. The model is represented by the equation,

$$y = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_D \cdot x_D + \epsilon$$

Here w_0 represents the bias or y-intercept which does not depend on the features, $w_1, w_2, \ldots w_D$ represent the feature weights and ϵ represents the error term. Succinctly we can represent the model using vector notation. Let the input features $x_1, x_2, \ldots, x_D$ be represented by the vector $\mathbf{x} \in \mathbb{R}^D$ and let the weights $w_0, w_1, w_2, \ldots w_D$ be represented by the vector $\mathbf{w} \in \mathbb{R}^{D+1}$.

You might notice that $\mathbf{x}$ and $\mathbf{w}$ do not have compatible shape, rather we extend our input features with a new feature $x_0 = 1$. Thus for each datapoint we have $x_0 = 1, x_1, x_2, \ldots, x_D = \mathbf{x} \in \mathbb{R}^{D+1}$. The model can then be represented as follows,

$$y = \mathbf{w} \cdot \mathbf{x} + \epsilon$$

1.1 Least squares estimate in the general case

Now let's consider the dataset $(\mathbf{x}_i, y_i)_{i=1}^N$, which consists of N data-points, i.e. N output features y_i and N input vectors $\mathbf{x}_i$. Now our output predictions $\hat{y}_i$ can be written as follows,

$$\hat{y}_i = \sum_{j=0}^{D} x_{ij} \cdot w_j$$

or similarly,

$$\hat{y}_i = \mathbf{w} \cdot \mathbf{x}_i$$

where x_{ij} denotes the j-th feature of the i-th datapoint and we have $x_{i0} = 1$ for all i. We now express everything in matrix notation, let $\mathbf{X}$ denote the $N \times (D+1)$ data matrix, whose i-th row corresponds to the transpose of the input i-th input features $\mathbf{x}_i$, $\hat{\mathbf{y}}$ is the column vector whose i-th value corresponds to the i-th prediction $\hat{y}_i$, and $\mathbf{w}$ are the $D+1$ weights to be learned.

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_D \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_D^T \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix} = \begin{bmatrix} x_{10} & \ldots & x_{1D} \\ x_{20} & \ldots & x_{2D} \\ \vdots & \ddots & \vdots \\ x_{N0} & \ldots & x_{ND} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}$$

More succinctly we write the following, $\hat{\mathbf{y}} = \mathbf{X} \times \mathbf{w}$, where × denotes matrix multiplication, we won't write × explicitly from now on. The loss function $\mathcal{L}(\mathbf{w})$ can also be expressed with matrix notation as follows,

$$\begin{aligned} \mathcal{L}(\mathbf{w}) &= \frac{1}{2N} \sum_{i=1}^{N} (\mathbf{x}_i^T \mathbf{w} - y_i)^2 = \frac{1}{2N} (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= \frac{1}{2} \left(\mathbf{w}^T \left(\mathbf{X}^T \mathbf{X} \right) \mathbf{w} - \mathbf{w}^T \mathbf{X}^T \mathbf{y} - \mathbf{y}^T \mathbf{X} \mathbf{w} + \mathbf{y}^T \mathbf{y} \right) \\ &= \frac{1}{2} \left(\mathbf{w}^T \left(\mathbf{X}^T \mathbf{X} \right) \mathbf{w} - 2 \cdot \mathbf{y}^T \mathbf{X} \mathbf{w} + \mathbf{y}^T \mathbf{y} \right) \end{aligned}$$

Now rather than calculating every partial derivative one its own, we can leverage multivariate calculus to compute the least squares estimate directly. We just need to be aware of the following two rules for multivariate calculus,

(i) Linear expressions: $\nabla_{\mathbf{w}} \left(\mathbf{c}^T \mathbf{w}\right) = \mathbf{c}$

This is similar to the univariate calculus rule of: $\frac{d(c \cdot w)}{dw} = c$

(ii) Quadratic expressions: $\nabla_{\mathbf{w}} \left(\mathbf{w}^T \mathbf{A} \mathbf{w}\right) = \mathbf{A}\mathbf{w} + \mathbf{A}^T \mathbf{w}$ or $(= 2\mathbf{A}\mathbf{w}$ when $\mathbf{A}$ is symmetric).

This is similar to the univariate calculus rule of: $\frac{d(a \cdot w^2)}{dw} = 2a \cdot w$

Now taking the gradient of $\mathcal{L}(\mathbf{w})$ with respect to $\mathbf{w}$ gives us the following expression:

$$\nabla_{\mathbf{w}} \mathcal{L} = \frac{1}{N} \left(\left(\mathbf{X}^T \mathbf{X}\right) \mathbf{w} - \mathbf{X}^T \mathbf{y}\right)$$

By setting $\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{0}$ and solving we get,

$$\begin{aligned} \left(\mathbf{X}^T \mathbf{X}\right) \mathbf{w} &= \mathbf{X}^T \mathbf{y} \\ \mathbf{w} &= \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y} \quad \text{(Assuming that } \left(\mathbf{X}^T \mathbf{X}\right) \text{ is invertible)} \end{aligned}$$

The predictions made by the model on the data matrix $\mathbf{X}$ are then given by,

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} = \mathbf{X} \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$$

The end!