Introduction to Linear Regression

Damien Benveniste

October 10, 2017

Introduction

The idea is to find a functional relationship between a response y to a set of variables $\{X_1, X_2, ..., X_m\}$ in a linear fashion:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n = \sum_{j=0}^{m} \beta_j X_j \text{ with } X_0 = 1 \tag{1}$$

However $\sum_{j=0}^{m} \beta_j X_j$ is simply an estimate of y and we use the hat notation:

$$\sum_{j=0}^{m} \beta_j X_j = \hat{y} \tag{2}$$

A typical metric to understand how far $\hat{y}$ is from y is the Mean Square Error (MSE) or square loss function:

$$\epsilon = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \sum_{j=0}^{m} \beta_j X_{ij})^2 \tag{3}$$

where n is the number of samples in the data.

1 The Normal Equations

How can we choose the β_i such that ϵ is minimized? Let's redefine the variables:

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1m} \\ x_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ X_{n1} & \dots & \dots & X_{nm} \end{bmatrix}, \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{4}$$

We have

$$\epsilon(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2 \tag{5}$$

We find β^* that minimizes ϵ by solving

$$\begin{aligned}
\nabla_\beta \epsilon(\beta) &= 0 \\
&= \nabla_\beta \|\mathbf{y} - \mathbf{X}\beta\|^2 \\
&= 2\mathbf{X}^T(\mathbf{X}\beta - \mathbf{y}) \\
&= 2\mathbf{X}^T\mathbf{X}\beta - 2\mathbf{X}^T\mathbf{y} \qquad (6)
\end{aligned}$$

$$\Rightarrow \boxed{\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}} \qquad (7)$$

2 The Gradient Descent algorithm

For large data sets it might be computationally difficult to inverse the $(\mathbf{X}^T\mathbf{X})$ matrix. The gradient descent algorithm allows to learn the weights β by iteratively performing updates from an initial guess:

$$\beta_j \leftarrow \beta_j - \alpha \frac{\partial \epsilon(\beta)}{\partial \beta_j} \qquad (8)$$

we have

$$\frac{\partial \epsilon(\beta)}{\partial \beta_j} = 2 \sum_{i=1}^{n} (y_i - \sum_{k=0}^{m} \beta_k X_{ik}) X_{ij} \qquad (9)$$

The gradient descent algorithm becomes:

Data: $\mathbf{X}$, $\mathbf{y}$
Initialize at random: $\{\beta_0, \beta_1, \ldots, \beta_m\}$
while *not convergence* **do**
  for j *in* $\{0, 1, \ldots, m\}$ **do**
    $\beta_j \leftarrow \beta_j - \sum_{i=1}^{n}(y_i - \sum_{k=0}^{m} \beta_k X_{ik})X_{ij}$
  end
end