Simplified Cost Function and Gradient Descent

Note: [6:53 - the gradient descent equation should have a 1/m factor]

We can compress our cost function's two conditional cases into one case:

$\mathrm{Cost}(h_\theta(x), y) = -y \; \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$

Notice that when y is equal to 1, then the second term $(1-y)\log(1-h_\theta(x))$ will be zero and will not affect the result. If y is equal to 0, then the first term $-y\log(h_\theta(x))$ will be zero and will not affect the result.

We can fully write out our entire cost function as follows:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

Logistic Regression Cost Function

A vectorized implementation is:

$$\begin{aligned} h &= g(X\theta) \\ J(\theta) &= \frac{1}{m}\cdot\left(-y^T\log(h) - (1-y)^T\log(1-h)\right) \end{aligned}$$

Gradient Descent

Remember that the general form of gradient descent is:

$$\begin{aligned} & Repeat\ \{ \\ & \quad \theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta) \\ & \} \end{aligned}$$

We can work out the derivative part using calculus to get:

$$\begin{aligned} & Repeat\ \{ \\ & \quad \theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \\ & \} \end{aligned}$$

Notice that this algorithm is identical to the one we used in linear regression. We still have to simultaneously update all values in theta.

A vectorized implementation is:

$\theta := \theta - \frac{\alpha}{m}X^T(g(X\theta) - \vec{y})$