Logistic Regression Explained

James Taylor

Suppose we have training data (x_α, y_α) for $\alpha = 1, \ldots, n$ where $x_\alpha = (1, x_{\alpha 1}, \ldots, x_{\alpha p})$ is a vector of observed values for our predictors, and $y_\alpha \in \{0, 1\}$ is its corresponding qualitative response.

We begin by assuming that $p(X) := p(Y = 1 \mid X)$ can be well approximated by the logistic function

$$f_\beta(X) = \frac{1}{1 + e^{-\beta^T X}} \qquad f_\beta : \mathrm{dom}(X) \to [0, 1]$$

for some value of β. We consider the optimal value of β to be that which maximizes the likelihood function

$$L(\beta) = \prod_{\alpha=1}^{n} f_\beta(x_\alpha)^{y_\alpha} (1 - f_\beta(x_\alpha))^{1 - y_\alpha}$$

or equivalently, which maximizes the log-likelihood function

$$\begin{aligned} \ell(\beta) &= \log \left(\prod_{\alpha=1}^{n} f_\beta(x_\alpha)^{y_\alpha} (1 - f_\beta(x_\alpha))^{1 - y_\alpha} \right) \\ &= \sum_{\alpha=1}^{n} \left(y_\alpha \log f_\beta(x_\alpha) + (1 - y_\alpha) \log(1 - f_\beta(x_\alpha)) \right) \end{aligned}$$

There does not exist a closed-form expression for the β which maximizes this likelihood, so we're going to use gradient ascent to estimate β. As such, we're going to need to compute the partial derivatives of ℓ.

Our first step will be to compute the jth partial of f_β. Observe that $g(z) = 1/(1 + e^{-z})$ has $g'(z) = e^{-z}/(1 + e^{-z})^2 = g(z)(1 - g(z))$. Thus

$$\begin{aligned} \frac{\partial}{\partial \beta_j} f_\beta(x_\alpha) &= \frac{\partial}{\partial \beta_j} g(\beta^T x_\alpha) \\ &= g(\beta^T x_\alpha)(1 - g(\beta^T x_\alpha)) \frac{\partial}{\partial \beta_j} (\beta^T x_\alpha) \\ &= g(\beta^T x_\alpha)(1 - g(\beta^T x_\alpha)) x_{\alpha j} \\ &= f_\beta(x_\alpha)(1 - f_\beta(x_\alpha)) x_{\alpha j} \end{aligned}$$

It follows that the jth partial derivative of $\ell(\beta)$ is

$$\begin{aligned} \frac{\partial}{\partial \beta_j} \ell(\beta) &= \sum_{\alpha=1}^{n} \left(\frac{y_\alpha}{f_\beta(x_\alpha)} - \frac{1 - y_\alpha}{1 - f_\beta(x_\alpha)} \right) \frac{\partial}{\partial \beta_j} f_\beta(x_\alpha) \\ &= \sum_{\alpha=1}^{n} \left(\frac{y_\alpha}{f_\beta(x_\alpha)} - \frac{1 - y_\alpha}{1 - f_\beta(x_\alpha)} \right) f_\beta(x_\alpha)(1 - f_\beta(x_\alpha)) x_{\alpha j} \\ &= \sum_{\alpha=1}^{n} \left(y_\alpha (1 - f_\beta(x_\alpha)) - (1 - y_\alpha) f_\beta(x_\alpha) \right) x_{\alpha j} \\ &= \sum_{\alpha=1}^{n} \left(y_\alpha - f_\beta(x_\alpha) \right) x_{\alpha j} \end{aligned}$$

Therefore, if X is the $n \times (p+1)$ matrix with αth row $x_\alpha = (1, x_{\alpha 1}, \ldots, x_{\alpha p})$, and Y is the $n \times 1$ matrix of observed responses, we have

$$\frac{\partial}{\partial \beta_j} \ell(\beta) = (Y - f_\beta(X)) \cdot X_{*j}$$

where X_{*j} denotes the jth column of X. Thus,

$$\nabla \ell(\beta) = (Y - f_\beta(X))^T X$$

The Algorithm (Gradient Ascent).
The basic (and oversimplified) idea is that we first choose an initial value for β and a learning rate $\delta > 0$. We then continually update β via the algorithm:

1) $\beta_{new} = \beta_{old} + \delta \nabla(\ell(\beta_{old}))$
2) $\beta_{old} = \beta_{new}$
3) Repeat until some stopping criterion is met.

As I said, this is really oversimplified. In reality, we only want to update β if our last step actually increased the function we want to maximize (that is, only if $\ell(\beta_{new}) > \ell(\beta_{old})$). If it didn't then apparently we took too big of a step in the direction of the gradient, meaning we need to decrease our learning rate δ until we find a value that works. On the other hand, if our last update to β did "work", then we update β and increase δ to hopefully speed up convergence.

Logistic Regression with L2 Regularization.
To limit overfitting, we can penalize large values of β_i. One of the more common ways of doing this is via L2 regularization, in which we seek the value of β that maximizes

$$\ell(\beta) - \lambda \sum_{i>0} \beta_i^2$$

where $\lambda \geq 0$ is some user-supplied constant. Note that the partial derivative $\partial \ell / \partial \beta_0$ is the same as before, but for $j \neq 0$ we now have

$$\frac{\partial}{\partial \beta_j} \left(\ell(\beta) - \lambda \sum_{i>0} \beta_i^2 \right) = -2\lambda \beta_j + \sum_{\alpha=1}^{n} \left(y_\alpha - f_\beta(x_\alpha) \right) x_{\alpha j}$$

Thus,

$$\nabla \left(\ell(\beta) - \lambda \sum_{i>0} \beta_i^2 \right) = (Y - f_\beta(X))^T X - 2\lambda(0, \beta_1, \ldots, \beta_p)$$

Note that if $\lambda = 0$ then this reduces to regular logistic regression.