Logistic Regression

Jason Rennie
jrennie@ai.mit.edu

April 23, 2003

Abstract

This document gives the derivation of logistic regression with and without regularization.

1 Introduction

We consider binary classification where each example is labeled $+1$ or -1. We assume that an example has l features, each of which can take the value zero or one. We denote an example by $\vec{x}$ and the value of the k^{th} feature as x_k. We define an additional feature, $x_0 \equiv 1$, and call it the "bias" feature. We say that the probability of an example being drawn from the positive class is

$$p(y = +1|\vec{x}) = g\left(\sum_{k=0}^{l} w_k x_k\right), \tag{1}$$

where $g(z) = \frac{1}{1+e^{-z}}$. We use w_k, $k \in \{0, \ldots, l\}$, to denote the weight for the k^{th} feature. We call w_0 the bias weight.

2 Without Regularization

Logistic regression (LR) learns weights so as to maximize the likelihood of the data. Let $(\vec{x}_1, \ldots, \vec{x}_n)$ be a set of training data; let $(y_1, \ldots, y_n)$ be their corresponding labels. Let x_{ik} be the value of the k^{th} feature of example i. LR maximizes the (log-) likelihood of the data,

$$L(\vec{w}) = \sum_{i=1}^{n} \log g(y_i z_i), \tag{2}$$

where $z_i = \sum_k w_k x_{ik}$. Note that $1 - g(z) = g(-z)$.

2.1 Gradient Descent

First, we show how to learn the weights via gradient descent. The gradient of the log-likelihood with respect to the k^{th} weight is

$$\frac{\partial L}{\partial \vec{w}} \quad \text{where} \quad \frac{\partial L}{\partial w_k} = \sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i). \tag{3}$$

Note that $\frac{\partial g(z)}{\partial z} = g(z)g(-z)dz$. Recall that $z_i = \sum_k w_k x_{ik}$, $k \in \{0, \ldots, l\}$, and $x_{i0} \equiv 1$. Increasing our weight vector in the direction of the gradient increases L; each round we calculate new weights by adding a fraction of the gradient,

$$w_k^{(t+1)} = w_k^{(t)} + \epsilon \sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i). \tag{4}$$

ϵ is the learning rate.

Iteratively updating the weights in this fashion increases likelihood each round. The likelihood is convex, so we eventually reach the maximum. We are near the maximum when changes in the weights are small. We choose to stop when the sum of the absolute values of the weight differences is less than some small number, e.g. 10^{-6}.

2.2 Newton's Method

We can also learn the weights without having to select a learning rate. We do this by solving for the weight vector that gives a zero gradient. Newton's method iteratively updates weights as follows:

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \left[\frac{\partial^2 L}{\partial \vec{w} \partial \vec{w}}\right]^{-1} \frac{\partial L}{\partial \vec{w}}. \tag{5}$$

We have already given the gradient. Here we give the Hessian, or the matrix of second derivatives:

$$\frac{\partial^2 L}{\partial \vec{w} \partial \vec{w}} \quad \text{where} \quad \frac{\partial^2 L}{\partial w_j \partial w_k} = -\sum_{i=1}^{n} x_{ij} x_{ik} g(y_i z_i) g(-y_i z_i). \tag{6}$$

Again, the likelihood is a convex function of the weights, so we are guaranteed to find the maximum eventually. In practice, we stop when the sum of the absolute values of the weight differences is less than some small number.

3 With Regularization

The derivation and optimization of regularized LR is very similar to regular LR. The benefit of adding the regularization term is that we enforce a tradeoff between matching the training data and generalizing to future data.

For our regularized objective, we change the sign and add the squared L2 norm.

$$L = -\sum_{i=1}^{n} \log g(y_i z_i) + \frac{C}{2} \sum_{k=1}^{l} w_k^2. \tag{7}$$

C balanaces the tradeoff between the two terms. Note that we do not regularize the bias weight. The derivatives are nearly the same as before, the only differences being a change in sign and the addition of regularization terms. We only show cases that are different than the case of no regularization.

$$\frac{\partial L}{\partial w_k} = -\sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i) + C w_k, \quad k \neq 0, \tag{8}$$

$$\frac{\partial^2 L}{\partial w_k \partial w_k} = \sum_{i=1}^{n} x_{ik}^2 g(-y_i z_i) + C, \quad k \neq 0. \tag{9}$$

As before, we can perform gradient descent using the gradient. We change the sign since we are looking for a minimum,

$$w_k^{(t+1)} = w_k^{(t)} + \epsilon \sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i) - \epsilon C w_k^{(t)}, \quad k \neq 0. \tag{10}$$

The update for the bias weight, w_0, is identical to the non-regularized version. As can be seen, the regularization term encourages smaller weights. Alternately, we can use Newton's method as before, using the updated derivatives given above.