Notes on Logistic Regression and Logit Boost

dave debarr

Given a data set $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ with $\boldsymbol{x}_i \in R^p$ and $y_i \in \{+1, -1\}$, the goal of logistic regression is to learn a function that estimates $probability(y_i = +1|\boldsymbol{x}_i)$. A logistic regression model is a Generalized Linear Model (GLM) having the following form: $ln\left(\frac{probability(y_i = +1|\boldsymbol{x}_i)}{1-probability(y_i = +1|\boldsymbol{x}_i)}\right) = \boldsymbol{w}^t\boldsymbol{x_i}$. The function $ln\left(\frac{probability(y_i = +1|\boldsymbol{x}_i)}{1-probability(y_i = +1|\boldsymbol{x}_i)}\right)$ is known as the logit function. While simple linear regression could be used to estimate the $probability(y_i = +1|\boldsymbol{x}_i)$ directly, the range of the probability function is zero to one. By taking the logarithm of the odds ratio, we are converting the range from $[0,1]$ to $(-\infty, \infty)$. Solving $ln\left(\frac{probability(y_i = +1|\boldsymbol{x}_i)}{1-probability(y_i = +1|\boldsymbol{x}_i)}\right) = \boldsymbol{w}^t\boldsymbol{x_i}$ for $probability(y_i = +1|\boldsymbol{x}_i)$, we get $probability(y_i = +1|\boldsymbol{x}_i) = \frac{1}{1+exp(-\boldsymbol{w}^t\boldsymbol{x_i})}$:

$$
\begin{aligned}
ln\left(\frac{probability(y_i|\boldsymbol{x}_i)}{1 - probability(y_i|\boldsymbol{x}_i)}\right) &= \boldsymbol{w}^t\boldsymbol{x_i} \\
\frac{probability(y_i|\boldsymbol{x}_i)}{1 - probability(y_i|\boldsymbol{x}_i)} &= exp(\boldsymbol{w}^t\boldsymbol{x_i}) \\
probability(y_i|\boldsymbol{x}_i) &= exp(\boldsymbol{w}^t\boldsymbol{x_i})\big(1 - probability(y_i|\boldsymbol{x}_i)\big) \\
probability(y_i|\boldsymbol{x}_i) &= exp(\boldsymbol{w}^t\boldsymbol{x_i}) - exp(\boldsymbol{w}^t\boldsymbol{x_i})\, probability(y_i|\boldsymbol{x}_i) \\
probability(y_i|\boldsymbol{x}_i) + exp(\boldsymbol{w}^t\boldsymbol{x_i})\, probability(y_i|\boldsymbol{x}_i) &= exp(\boldsymbol{w}^t\boldsymbol{x_i}) \\
probability(y_i|\boldsymbol{x}_i)\big(1 + exp(\boldsymbol{w}^t\boldsymbol{x_i})\big) &= exp(\boldsymbol{w}^t\boldsymbol{x_i}) \\
probability(y_i|\boldsymbol{x}_i) &= \frac{exp(\boldsymbol{w}^t\boldsymbol{x_i})}{1 + exp(\boldsymbol{w}^t\boldsymbol{x_i})} \\
probability(y_i|\boldsymbol{x}_i) &= \frac{\frac{exp(\boldsymbol{w}^t\boldsymbol{x_i})}{exp(\boldsymbol{w}^t\boldsymbol{x_i})}}{\frac{1}{exp(\boldsymbol{w}^t\boldsymbol{x_i})} + \frac{exp(\boldsymbol{w}^t\boldsymbol{x_i})}{exp(\boldsymbol{w}^t\boldsymbol{x_i})}} \\
probability(y_i|\boldsymbol{x}_i) &= \frac{1}{1 + exp(-\boldsymbol{w}^t\boldsymbol{x_i})}
\end{aligned}
$$

The objective for logistic regression is to minimize the negative log likelihood loss function. By likelihood, we mean the likelihood of the model parameters. Given the mapping $y_i^* = \frac{y_i - 1}{2}$, which converts y_i to a binary zero/one indicator, the loss function can be expressed as

$$
\begin{aligned}
&- log\big(probability(y_i|\,\boldsymbol{x_i}; \boldsymbol{w})\big) \\
&= -log\left(\left(\frac{1}{1 + exp(-\boldsymbol{w}^t\boldsymbol{x_i})}\right)^{(y_i^*)}\left(1 - \frac{1}{1 + exp(-\boldsymbol{w}^t\boldsymbol{x_i})}\right)^{(1-y_i^*)}\right) \\
&= -log\left(\frac{1}{1+exp(-y_i\,\boldsymbol{w}^t\,\boldsymbol{x_i})}\right) = log\big(1 + exp(-y_i\,\boldsymbol{w}^t\,\boldsymbol{x_i})\big).
\end{aligned}
$$

Stochastic gradient descent is a commonly used method for learning the logistic regression model. The gradient of a function identifies the direction of change with the greatest increase for the value of a function, so gradient descent for logistic regression involves subtracting the gradient of the negative log likelihood loss function from the weight vector. The negative gradient of the loss function with respect to the weight vector is computed as follows:

$$
\begin{aligned}
-\frac{\partial}{\partial w} ln\big(1 + exp(-ywx)\big) &= -\frac{1}{1 + exp(-ywx)} \frac{\partial}{\partial w}\big(1 + exp(-ywx)\big) \\
&= -\frac{1}{1 + exp(-ywx)}\left(\frac{\partial}{\partial w}(1) + \frac{\partial}{\partial w}\big(exp(-ywx)\big)\right) \\
&= -\frac{1}{1 + exp(-ywx)}\left(0 + \frac{\partial}{\partial w}\big(exp(-ywx)\big)\right) = -\frac{1}{1 + exp(-ywx)} \frac{\partial}{\partial w}\big(exp(-ywx)\big) \\
&= -\frac{1}{1 + exp(-ywx)} exp(-ywx) \frac{\partial}{\partial w}(-ywx) = -\frac{exp(-ywx)}{1 + exp(-ywx)} \frac{\partial}{\partial w}(-ywx) \\
&= yx\left(\frac{exp(-ywx)}{1 + exp(-ywx)}\right) = yx\left(\frac{1}{1 + exp(ywx)}\right) = yx\left(1 - \frac{1}{1 + exp(-ywx)}\right) \\
&= \begin{cases} \left(1 - \dfrac{1}{1 + exp(-wx)}\right) x, & y = +1 \\ \left(0 - \dfrac{1}{1 + exp(-wx)}\right) x, & y = -1 \end{cases}
\end{aligned}
$$

Stochastic gradient descent involves updating the weight vector using a randomly selected training set observation: $\boldsymbol{w} = \boldsymbol{w} + \lambda\left(y_i{}^* - \frac{1}{1+exp(-\boldsymbol{w}^t \boldsymbol{x}_i)}\right)\boldsymbol{x}_i$ where λ is the size of the step in the direction of the negative gradient. The parameter λ is known as the learning rate parameter in the machine learning literature and the shrinkage parameter in the statistical learning literature.

Here's a simple synthetic logistic regression example. We assume the following model for generating the synthetic data: $probability(y_i = +1|x_i) = \frac{1}{1+exp(-(2x_i-1))}$. To generate random data, we can compare a uniform random number in the interval [0, 1] to the assumed probability for positive class membership, assigning the positive class label if the random number is less than the assumed probability. The following contingency table shows the proportion of positive class members for $x_i = 1$ and $x_i = 0$ for 2,000 training set observations.

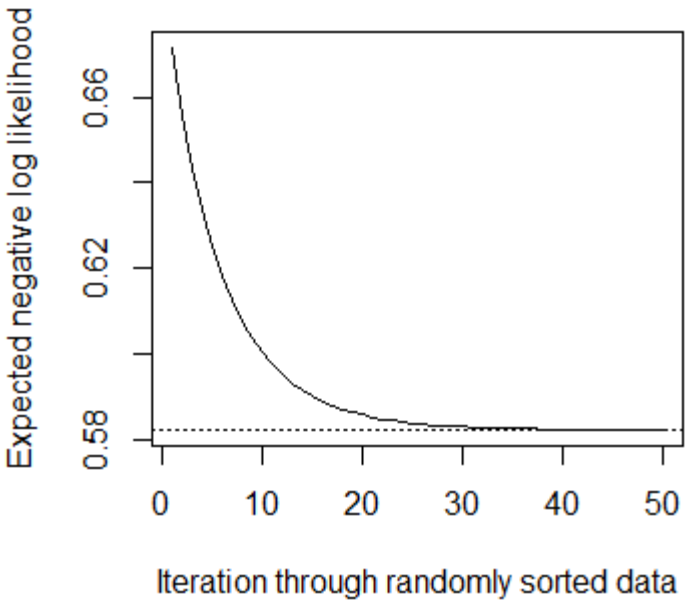
	$y_i = +1$	$y_i = -1$
$x_i = 1$	731	269
$x_i = 0$	269	731

The expected negative log likelihood for an optimal weight vector (logistic regression model) is

$$
-\left(\frac{1000}{2000}\left(\frac{1}{1+exp(-(2-1))} ln\left(\frac{1}{1+exp(-(2-1))}\right) + \left(1 - \frac{1}{1+exp(-(2-1))}\right) ln\left(1 - \frac{1}{1+exp(-(2-1))}\right)\right) + \right.
$$

$$
\left.\frac{1000}{2000}\left(\frac{1}{1+exp(-(0-1))} ln\left(\frac{1}{1+exp(-(0-1))}\right) + \left(1 - \frac{1}{1+exp(-(0-1))}\right) ln\left(1 - \frac{1}{1+exp(-(0-1))}\right)\right)\right) = 0.582.
$$

The following graph shows the progress of the stochastic gradient descent algorithm with w initialized to $[0,0]$ and $\lambda = 0.001$. The expected negative log likelihood for the optimal model is marked by the dotted horizontal line. After 50 iterations, $w = [1.96, -0.98]$. The first element of the weight vector is the coefficient for x_i, while the second element of the weight vector is the intercept (also known as a bias term).



A discriminant function is a function that assigns an observation to a class. For example, using logistic regression for discrimination, we may choose to assign observations to the positive class based on $sign(\boldsymbol{w}^t \boldsymbol{x}_i)$, assigning observations to the positive class if $\boldsymbol{w}^t \boldsymbol{x}_i > 0$ or to the negative class otherwise.

The term boost is a verb that means "to improve." In machine learning, boosting is the use of an ensemble of "weak" (slightly better than random) machine learning models (often stumps or shallow trees), where each model added focuses on reducing residual error for previously constructed models. The Logit Boost algorithm was defined by Friedman, Hastie, and Tibshirani in their "Additive Logistic Regression: a Statistical View of Boosting" paper: http://www.stanford.edu/~hastie/Papers/AdditiveLogisticRegression/alr.pdf.

As shown in algorithm 3 of their paper, an adaptive Newton method is used for learning the Logit Boost (additive logistic regression) model. The Logit Boost algorithm has 3 major steps:

1. For all observations, initialize observation weight $w_i = \frac{1}{n}$, log odds $F_0(\boldsymbol{x}_i) = 0$, and probability $p(\boldsymbol{x}_i) = \frac{1}{1+exp(-F_0(\boldsymbol{x}_i))}$
2. Repeat for model $m = 1, 2, \ldots, M$:
    a. Compute the working responses (residual error) r_i and weights w_i for the current iteration
    $$r_i = \frac{{y_i}^* - p(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)\big(1 - p(\boldsymbol{x}_i)\big)}$$
    $$w_i = p(\boldsymbol{x}_i)\big(1 - p(\boldsymbol{x}_i)\big)$$
    b. Fit the function $\hat{f}_m(\boldsymbol{x}_i)$ by a weighted least-squares regression (using $\boldsymbol{x}_i$ to predict the residual r_i)
    c. Update $F_m(\boldsymbol{x}_i) = F_{m-1}(\boldsymbol{x}_i) + \hat{f}_m(\boldsymbol{x}_i)$ and $p(\boldsymbol{x}_i) = \frac{1}{1+exp(-F_m(\boldsymbol{x}_i))}$
3. Output the additive logistic regression function as
$$probability(y_i = +1|\boldsymbol{x}_i) = \frac{1}{1 + exp\big(-\sum_{m=1}^{M} \hat{f}_m(\boldsymbol{x}_i)\big)}$$

The working response r_i is simply the ratio of the negative first derivative of the negative log likelihood loss function to the second derivative of the negative log likelihood loss function. The negative first derivative of the negative log likelihood function is $-\frac{\partial}{\partial F(\boldsymbol{x})} ln\left(1 + exp\big(-y\, F(\boldsymbol{x})\big)\right) = y^* - \frac{1}{1+\exp(-F(\boldsymbol{x}))}$.

The second derivative of the negative log likelihood function is:

$$\frac{\partial^2}{\partial^2 F(\boldsymbol{x})} ln\left(1+exp\big(-y\,F(\boldsymbol{x})\big)\right) = \frac{\partial}{\partial F(\boldsymbol{x})}\left(-y\left(\frac{exp\big(-y\,F(\boldsymbol{x})\big)}{1+exp\big(-y\,F(\boldsymbol{x})\big)}\right)\right)$$

$$= -y\frac{\partial}{\partial F(\boldsymbol{x})}\left(\frac{exp\big(-y\,F(\boldsymbol{x})\big)}{1+exp\big(-y\,F(\boldsymbol{x})\big)}\right)$$

$$= -y\left(\frac{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)\frac{\partial}{\partial F(\boldsymbol{x})}\Big(exp\big(-y\,F(\boldsymbol{x})\big)\Big) - exp\big(-y\,F(\boldsymbol{x})\big)\frac{\partial}{\partial F(\boldsymbol{x})}\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)}{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)^2}\right)$$

$$= -y\left(\frac{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)exp\big(-y\,F(\boldsymbol{x})\big)\frac{\partial}{\partial F(\boldsymbol{x})}\big(-y\,F(\boldsymbol{x})\big) - exp\big(-y\,F(\boldsymbol{x})\big)exp\big(-y\,F(\boldsymbol{x})\big)\frac{\partial}{\partial F(\boldsymbol{x})}\big(-y\,F(\boldsymbol{x})\big)}{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)^2}\right)$$

$$= -y\left(\frac{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)exp\big(-y\,F(\boldsymbol{x})\big)(-y) - exp\big(-y\,F(\boldsymbol{x})\big)exp\big(-y\,F(\boldsymbol{x})\big)(-y)}{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)^2}\right)$$

$$= y^2\left(\frac{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)exp\big(-y\,F(\boldsymbol{x})\big)}{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)^2} - \frac{exp\big(-2y\,F(\boldsymbol{x})\big)}{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)^2}\right)$$

$$= \frac{exp\big(-y\,F(\boldsymbol{x})\big)}{1+exp\big(-y\,F(\boldsymbol{x})\big)} - \frac{exp\big(-2y\,F(\boldsymbol{x})\big)}{\Big(1+exp\big(-y\,F(\boldsymbol{x})\big)\Big)^2} = \frac{exp\big(-y\,F(\boldsymbol{x})\big)}{1+exp\big(-y\,F(\boldsymbol{x})\big)}\left(1-\frac{exp\big(-y\,F(\boldsymbol{x})\big)}{1+exp\big(-y\,F(\boldsymbol{x})\big)}\right)$$

$$= \frac{1}{1+exp\big(y\,F(\boldsymbol{x})\big)}\left(1-\frac{1}{1+exp\big(y\,F(\boldsymbol{x})\big)}\right) = \left(1-\frac{1}{1+exp\big(-y\,F(\boldsymbol{x})\big)}\right)\frac{1}{1+exp\big(-y\,F(\boldsymbol{x})\big)}$$

The residual $r_i = \frac{-\frac{\partial}{\partial F(x_i)} ln\left(1+exp\left(-y\,F(x_i)\right)\right)}{\frac{\partial^2}{\partial^2 F(x_i)} ln\left(1+exp\left(-y\,F(x_i)\right)\right)} = \frac{{y_i}^* - \frac{1}{1+\exp\left(-F(x_i)\right)}}{\frac{1}{1+\exp\left(-F(x_i)\right)}\left(1-\frac{1}{1+\exp\left(-F(x_i)\right)}\right)}$ is simply an application of the Newton-Raphson learning method to additive logistic regression, based on Taylor Series expansion. The Taylor Series can be used to approximate the value of some function value g$(x+\delta)$:

$$g(x+\delta) \approx g(x) + g'(x)\big((x+\delta)-x\big)$$

We want to add δ so the gradient of the negative log likelihood function will be zero, which gives us:

$$\begin{aligned} g(x) + g'(x)\big((x+\delta)-x\big) &\approx 0 \\ g'(x)\big((x+\delta)-x\big) &\approx -g(x) \\ \big((x+\delta)-x\big) &\approx \frac{-g(x)}{g'(x)} \\ \delta &\approx \frac{-g(x)}{g'(x)} \end{aligned}$$

The r_i residual is our δ and the $\frac{\partial}{\partial F(\boldsymbol{x}_i)} ln\left(1+exp\big(-y\,F(\boldsymbol{x}_i)\big)\right)$ function is our $g(x)$.

Regression stumps are commonly used for the $\hat{f}_m(\boldsymbol{x}_i)$ functions. Each possible split is evaluated to construct each regression stump. For numeric features, the partition conditions are "less than or equal to split value" and "greater than split value." For nominal features, the partition conditions are "equal to the split value" and "not equal to the split value." The split that minimizes the weighted variance of the predicted responses is chosen as the partitioning criteria:

$$\min_{PartitionSet}\left(\sum_{partition}\sum_{P\in PartitionSet}\left(\frac{\sum_{i\in P}w_i}{\sum_{i=1}^{n}w_i}\left(\frac{\sum_{i\in P}\left(w_i({r_i}^2)\right)}{\sum_{i\in P}w_i}-\left(\frac{\sum_{i\in P}(w_i r_i)}{\sum_{i\in P}w_i}\right)^2\right)\right)\right)$$

For each partition, the stump predicts the expected response for observations in that partition:

$$\frac{\sum_{i\in P}(w_i r_i)}{\sum_{i\in P}w_i}$$

For the synthetic example, the first regression stump would be:

$\hat{f}_1(x_i) \coloneqq$

$if\ x_i == 1\ then$

$$return\ \frac{731*\left(\frac{1}{2000}\right)*(+2)+\ 269*\left(\frac{1}{2000}\right)*(-2)}{1000*\left(\frac{1}{2000}\right)} = +0.972$$

else

$$return\ \frac{731*\left(\frac{1}{2000}\right)*(-2)+\ 269*\left(\frac{1}{2000}\right)*(+2)}{1000*\left(\frac{1}{2000}\right)} = -0.972$$

Implementation notes:

The range of r_i is often restricted to [-3, 3].

A learning rate parameter λ is often applied to the $\hat{f}_m(\boldsymbol{x}_i)$ estimates, to encourage smaller steps.

For multi-class problems with J classes, we simply construct J weight functions as shown in algorithm 6 of the additive logistic regression paper cited earlier (repeated below). The probability that observation $\boldsymbol{x}_i$ belongs to class j is estimated as

$$p_j(\boldsymbol{x}_i) = \frac{exp\left(F_j(\boldsymbol{x}_i)\right)}{\sum_{k=1}^{J} exp\left(F_k(\boldsymbol{x}_i)\right)}$$

where $\sum_{k=1}^{J} F_k(\boldsymbol{x}_i) = 0$.

LogitBoost (*J classes*)

1. Start with weights $w_{ij} = 1/N,\ i = 1, \ldots, N,\ j = 1, \ldots, J,\ F_j(x) = 0$ and $p_j(x) = 1/J\ \forall\ j$.

2. Repeat for $m = 1, 2, \ldots, M$:

   (a) Repeat for $j = 1, \ldots, J$:

   (i) Compute working responses and weights in the jth class,

$$z_{ij} = \frac{y_{ij}^* - p_j(x_i)}{p_j(x_i)(1 - p_j(x_i))},$$

$$w_{ij} = p_j(x_i)(1 - p_j(x_i)).$$

   (ii) Fit the function $f_{mj}(x)$ by a weighted least-squares regression of z_{ij} to x_i with weights w_{ij}.

   (b) Set $f_{mj}(x) \leftarrow \frac{J-1}{J}\left(f_{mj}(x) - \frac{1}{J}\sum_{k=1}^{J} f_{mk}(x)\right)$, and $F_j(x) \leftarrow F_j(x) + f_{mj}(x)$.

   (c) Update $p_j(x)$ via (40).

3. Output the classifier $\arg\max_j F_j(x)$.

ALGORITHM 6. *An adaptive Newton algorithm for fitting an additive multiple logistic regression model.*