Variations of Logistic Regression with Stochastic Gradient Descent

Panqu Wang(pawang@ucsd.edu)
Phuc Xuan Nguyen(pxn002@ucsd.edu)

January 26, 2012

Abstract

In this paper, we extend the traditional logistic regression model(LR) to the bounded logistic regression model(BLR) and compare them. We also derive the update rules of both model using stochastic gradient desent(SGD). The effects of choosing different learning rate schedule, stopping conditions, parameters initialization and learning algorithm settings are also discussed. We get the accuracy rate of 83.80% for LR model and 84.75% for BLR model on the test set.

1 Introduction

The logistic regression (LR) model is widely used in prediction the probability of occurrence of an event by fitting the training data to a logistic regression function. We learn a logistic regression classifier by maximizing the log joint conditional likelihood of training examples. The LR model can be extended to the bounded logistic regression (BLR) model by setting both upper and lower bound to the logistic regression function. In section 2, we discuss the explanation and comparison of the two models and derive the update rules for both models using stochastic gradient descent (SGD). In section 3, we will discuss the effects of different learning rate schedule, stopping conditions and parameters initialization. We will discuss the performance of the models and complexity of our implementation in section 3.4.

2 Logistic Regression Model

First, we will give a brief review of the regularized logistic regression model, and then we will introduce the bounded logistic regression model and derive the update rule.

2.1 Regularized Logistic Regression Model

Suppose y is a Bernoulli outcome and x is a real-valued vector, we can set up the logistic regression model for conditional likelihood $p(y \mid x; \alpha, \beta)$:

$$\log \frac{p}{1-p} = \alpha + \sum_{j=1}^{d} \beta_j x_j \tag{1}$$

We use j to index over d dimensional feature values $x_1$to x_d of a single example, and we will use i to index over training data x_i to x_n. In this model, our goal is to find β_j that maximize the log likelihood function(LCL), so we will get the β_j by setting the partial derivative of LCL

$$\frac{\partial}{\partial \beta_j} LCL = \frac{\partial}{\partial \beta_j} \log L(\beta; x, y) = \sum_i (y_i - p_i) x_{ij} \tag{2}$$

to zero. By using the SGD method to train β_j, the update rule is

$$\beta_j := \beta_j + \lambda (y - p) x_j \tag{3}$$

where λ is the learning rate.

We use regularization method to impose a penalty on the magnitude of the parameter values. We must minimize the penalty while maximizing the likelihood. So the optimization problem to be solved is

$$\hat{\beta} = \text{argmax}_\beta LCL - \mu \|\beta\|_2 \tag{4}$$

where $\|\beta\|_2$ is the L_2-norm of the parameter vector, μ is the strength of regularization which quantifies the trade-off between maximizing likelihood and minimizing parameter values. By simply including the penalty term when calculating the derivative of the gradient for each example, the stochastic gradient will be extended to the regularization as

$$\frac{\partial}{\partial \beta_j} [\log p(y \mid x; \beta) - \mu \sum_{j=0}^{d} \beta_j^2] = [\frac{\partial}{\partial \beta_j} \log L(\beta; x, y)] - \mu 2 \beta_j \tag{5}$$

By putting (2) into (5), we can get the update rule with regularization

$$\beta_j := \beta_j + \lambda \mid (y - p) x_j - 2\mu \beta_j \mid \tag{6}$$

where λ is the learning rate.

2.2 Bounded Logistic Regression Model

The expression for bounded logistic regression model is

$$P(y = 1 \mid x; \theta) = \sigma(f) + (\sigma(c) - \sigma(f)) \sigma(w^T x + b) \tag{7}$$

where$\sigma(z) = \frac{1}{1+e^{-z}}$. $\sigma(z)$ is monotonically increasing from 0 to 1 when z increases from $-\infty$ to $+\infty$.

The parameter c and f set the upper bound and lower bound of the probability distribution function. Since the value of probability p should between 0 and 1, c and f are passed through the sigmoid which restricts the upper bound to be 1 and lower bound to be 0. In this model we set $\sigma(c) > p > \sigma(f)$ by default. In particular, if $\sigma(c) = 1$ and $\sigma(f) = 0$, the bounded logistic regression model will be degraded to the basic logistic regression model. The bounded logistic regression model is easier to understand in the form

$$\log \frac{p - \sigma(f)}{\sigma(c) - p} = b + w^T x \tag{8}$$

which is similar to the form (2).

Same to the logistic regression model, we use j to index over d dimensional feature values x_1 to x_d of a single example and i to index over training data x_i to x_n. To simplify the following discussion, we assume from now on that $b = w_0$ and $x_0 = 1$ for every example x, so the parameter vector θ is $(f, c, w) \in \mathbb{R}^{d+3}$. The bounded log conditional likelihood for each training example is

$$\begin{aligned} BLCL &= \sum_{i=1}^{n} \log L(\theta; y_i \mid x_i) = \sum_{i=1}^{n} \log f(y_i \mid x_i; \theta) \\ &= \sum_{i;y_i=1} \log p_i + \sum_{i;y_i=0} \log(1 - p_i) \end{aligned} \tag{9}$$

where $p_i = p(y = 1 \mid x_i; \theta)$. To maximize the BLCL, we should take partial derivative with respect to f, c and w *separately* and find the point to let the value of each partial derivative be 0.

For an individual example $< x, y >$, if its label y=1 the partial derivative with respect to w_j is

$$\frac{\partial}{\partial w_j} \log p = \frac{1}{p} \frac{\partial}{\partial w_j} p \tag{10}$$

while if $y = 0$ it is

$$\frac{\partial}{\partial w_j} \log(1 - p) = \frac{1}{1 - p}(-\frac{\partial}{\partial w_j} p) \tag{11}$$

let $e = \exp[-\sum_{j=0}^{d} w_j x_j]$, so $p = \sigma(f) + (\sigma(c) - \sigma(f))/(1 + e)$. Rewrite this we can have $e = \frac{\sigma(c)-p}{p-\sigma(f)}$ and $1 + e = \frac{\sigma(c)-\sigma(f)}{p-\sigma(f)}$. With this notation we have

$$\begin{aligned} \frac{\partial}{\partial w_j} p &= (\sigma(c) - \sigma(f)) \frac{-1}{(1+e)^2} (e) \frac{\partial}{\partial w_j} e \\ &= \frac{(p - \sigma(f))(\sigma(c) - p)}{\sigma(c) - \sigma(f)} x_j \end{aligned} \tag{12}$$

So

$$\frac{\partial}{\partial w_j}\log p = \frac{1}{p}\frac{(p-\sigma(f))(\sigma(c)-p)}{\sigma(c)-\sigma(f)})x_j \tag{13}$$

$$\frac{\partial}{\partial w_j}\log(1-p) = \frac{-1}{1-p}\frac{(p-\sigma(f))(\sigma(c)-p)}{\sigma(c)-\sigma(f)}x_j \tag{14}$$

For the entire training set the partial derivative with respect to w_j is

$$\begin{aligned}\frac{\partial}{\partial w_j}BLCL &= \sum_{i;y_i=1}\frac{\partial}{\partial w_j}\log p_i + \sum_{i;y_i=0}\frac{\partial}{\partial w_j}\log(1-p_i) \\ &= \sum_i \frac{y_i-p_i}{p_i(1-p_i)}\frac{(p_i-\sigma(f))(\sigma(c)-p_i)}{\sigma(c)-\sigma(f)}x_{ij} \end{aligned}\tag{15}$$

Now we will take the partial derivative of BLCL with respect to f and c. Note $p = \sigma(f)+(\sigma(c)-\sigma(f))\sigma(w^Tx)$. By utilizing the property $\sigma'(f) = \sigma(f)\cdot\sigma(-f)$, we can easily write the partial derivative as

$$\frac{\partial}{\partial f}\log p = \frac{1}{p}\sigma(f)\sigma(-f)(1-\sigma(w^Tx)) \tag{16}$$

$$\frac{\partial}{\partial f}\log(1-p) = \frac{-1}{1-p}\sigma(f)\sigma(-f)(1-\sigma(w^Tx)) \tag{17}$$

For the entire training set the partial derivative with respect to f is

$$\begin{aligned}\frac{\partial}{\partial f}BLCL &= \sum_{i;y_i=1}\frac{\partial}{\partial f}\log p_i + \sum_{i;y_i=0}\frac{\partial}{\partial f}\log(1-p_i) \\ &= \sum_i \frac{y_i-p_i}{p_i(1-p_i)}\sigma(f)\sigma(-f)(1-\sigma(w^Tx)) \end{aligned}\tag{18}$$

Similarly we can get the the partial derivative of BLCL with respect to c:

$$\frac{\partial}{\partial c}\log p = \frac{1}{p}\sigma(c)\sigma(-c)\sigma(w^Tx) \tag{19}$$

$$\frac{\partial}{\partial c}BLCL = \sum_i \frac{y_i-p_i}{p_i(1-p_i)}\sigma(c)\sigma(-c)\sigma(w^Tx) \tag{20}$$

After we get all the partial derivatives, we can update the parameters and increase the log likelihood incrementally by doing stochastic gradient training. The update rules of bounded logistic regression model are

$$w_j = w_j + \lambda \frac{y-p}{p(1-p)} \frac{(p-\sigma(f))(\sigma(c)-p)}{\sigma(c)-\sigma(f)} x_j \quad (21)$$

$$f = f + \lambda \frac{y-p}{p(1-p)} \sigma(f)\sigma(-f)(1-\sigma(w^T x)) \quad (22)$$

$$c = c + \lambda \frac{y-p}{p(1-p)} \sigma(c)\sigma(-c)\sigma(w^T x) \quad (23)$$

where λ is the learning rate. Similar to the logistic regression model, if we want to use regularization method to solve the overfitting problem, we can add a penalty of $\mu\|\theta\|_2$ after each BLCL function. After taking derivatives like (5), the update rules with regularization for all parameters in bounded logistic regression model are

$$w_j : = w_j + \lambda \mid \frac{y-p}{p(1-p)} \frac{(p-\sigma(f))(\sigma(c)-p)}{\sigma(c)-\sigma(f)} x_j - 2\mu w_j \mid \quad (24)$$

$$f = f + \lambda \mid \frac{y-p}{p(1-p)} \sigma(f)\sigma(-f)(1-\sigma(w^T x)) - 2\mu f \mid \quad (25)$$

$$c = c + \lambda \mid \frac{y-p}{p(1-p)} \sigma(c)\sigma(-c)\sigma(w^T x) - 2\mu c \mid \quad (26)$$

2.3 Comparison of the Models

The logistic regression(LR) model and the bounded logistic regression(BLR) model are similar. From equation (1) and (8), we can see that if we let $\sigma(c)$ be 1 and $\sigma(f)$ be 0, the bounded logistic model will reduce to basic logistic regression model. The sigmoid function restricts the bounds to be within range 0 to 1.

The BLR model can capture probability distributions that the LR model cannot. For example, if the distribution of all x_js lie between $[0.1, 0.9]$, the BLR model can capture it by learning the parameter c and f and let $\sigma(c)$ be 0.9 and $\sigma(f)$ be 0.1 as a result. The LR model will output probability outside of this range if $\| x_j \|$goes to infinity. So it cannot have a cap of 0.9 on the outputs. In that sense it cannot mimic the behavior of BLR. In general, if the probability distribution are guaranteed within a range less than $[0, 1]$, the BLR model will capture the distribution while the LR model cannot.

On the other hand, the LR model can also capture probability distributions which the BLR model cannot. If the probability distribution is $p(y = 1 \mid x) = \sigma(w_0 x)$ and we want to learn w_0, the LR model can find w_0. However, if the BLR model tries to learn this distribution, it will need to let c to be $+\infty$ and f to be $-\infty$. This theory, this is possible. However, in practice numerical issues may arise. In this sense the BLR model will not manage to recover w_0.

In addition, the LR model results in a convex optimization problem, while the BLR model does not. Since the convex optimization problem requires to find the only global minimum of the log conditional likelihood(LCL) function, it requires the second order derivative of the negative LCL function to be greater or equal to zero at all points of x_j. The second order derivative of LR model is

$$\frac{\partial^2}{\partial \beta_j^2} LCL = \frac{\partial}{\partial \beta_j}(\frac{\partial}{\partial \beta_j} LCL) = \frac{\partial}{\partial \beta_j}(\sum_i (y_i - p_i) x_{ij}) = \sum_i x_{ij}(-\frac{\partial}{\partial \beta_j} p_i) \quad (27)$$

Since $x_{ij} \geq 0$ and $-\frac{\partial}{\partial \beta_j} p_i = -p_i(1-p_i)x_{ij} \leq 0$, so $\frac{\partial^2}{\partial \beta_j^2} LCL \leq 0$ and the global optimum exists. So the linear regression model is a convex optimization problem.

Now let us examine the BLR model. For convenience we will take only one training example from x_i. We will examine the second order partial derivative with respect to w_j first.

Recall

$$\frac{\partial}{\partial w_j} BLCL = \frac{y-p}{p(1-p)} \cdot \frac{(p-\sigma(f))(\sigma(c)-p)}{\sigma(c)-\sigma(f)} x_j$$

So

$$\frac{\partial^2}{\partial w_j^2} BLCL = \frac{\partial}{\partial w_j}(\frac{y-p}{p(1-p)} \cdot \frac{(p-\sigma(f))(\sigma(c)-p)}{\sigma(c)-\sigma(f)} x_j)$$

if $y = 1$,

$$\begin{aligned} \frac{\partial^2}{\partial w_j^2} BLCL &= \frac{\partial}{\partial w_j}(\frac{1}{p} \cdot \frac{(p-\sigma(f))(\sigma(c)-p)}{\sigma(c)-\sigma(f)} x_j) \\ &= \frac{1}{p^2} \frac{\sigma(c)\sigma(f) - p^2}{\sigma(c)-\sigma(f)} x_j \frac{\partial}{\partial w_j} p \end{aligned} \quad (28)$$

Note $\sigma(c) > p > \sigma(f)$, so if we require $\frac{\partial^2}{\partial w_j^2} BLCL \leq 0$, $\sigma(c)\sigma(f) - p^2$ should be less than 0. However, this condition will not always hold. Suppose $\sigma(c) = \sigma(f) = 0.5$, $p = 0.25$, $\sigma(c)\sigma(f) - p^2 = \frac{3}{16} > 0$. So the second derivative of BLCL with respect to w_j will not always be negative, so the gradient being 0 does not guarantee being a global optimum. As a result, the gradient descent in BLR will only find a locally optimal solution hence it is not a convex optimization problem with respect to w_j. Please note if we take the second order derivative with respect to parameter f and c, the results are:

$$\frac{\partial^2}{\partial f^2} BLCL = \frac{1}{p^2}(1-\sigma(w^T x))\sigma(f)\sigma(-f)[p(\sigma(f)+\sigma(-f)) - 1] \leq 0$$

$$\frac{\partial^2}{\partial c^2} BLCL = \frac{1}{p^2}\sigma(w^T x)\sigma(c)\sigma(-c)[p(\sigma(c)+\sigma(-c)) - 1] \leq 0$$

So if we fix w_j and train the parameter f and c, we will find the global optimum of f and c. However, the BLR model is in general not a convex optimization problem because we can only get local optimum of BLR model with respect to w_j.

3 Experimentation

3.1 Dataset and feature description

We want to learn these two models using UCI Adult dataset(Blake & Merz, 1998), composed of 32561 training examples and 16281 testing examples. Each example consists of 14 features and a label of whether that person's income exceeds 50k.

We take preprocessing steps to transform data and encode into features. We ignore the fnlwgt feature because we wish to use weights from our own model. We also use encode the categories features(such as sex, education) using k-1 binary features to provide uniqueness to our model. We also notice that the majority data has native country of USA and Mexico. We transform the native country category set into a smaller set {"USA", "Mexico", "Rest"}. Besides reducing overfitting, this further speeds up the algorithm.

We use the stochastic gradient descent technique to learn the parameters of each models. Since SGD updates requires features to have similar ranges, we normalize the continuous features, such as age, into the range $[0, 1]$ by subtracting the minimum and dividing by the difference between the maximum and the minimum.

3.2 Design

3.2.1 Learning rate schedule

We wish to decide on which learning rate schedule fits the training data the best. Two schedules that we are interested are:

1. exponential dampening, $\lambda_e = \lambda_0/c^e$ for some constant c > 1
2. inverse dampening, $\lambda_e = \lambda_0/(1 + \lambda_0 ce)$ for some constant c

We perform k=50 epochs of the stochastic gradient descent with logistic regression model on the training data. For each epoch, we record the log conditional likelihood of all the data. We pick k=50 because the algorithm generally converges at the epoch.

3.2.2 Settings variables

We wish to learn the regularization strength and initial learning rate as settings for the stochastic gradient descent algorithm. We use Nelder-Mead algorithm(Nelder, 1965) combined with a 5-fold cross-validation on the training data to find the optimal settings. We use inaccuracy(false positive + false negative) as the objective function for the Nelder-Mead algorithm.

For the last run, we use the optimal settings to train β_j with the whole training set. We limit the number iterations of the Nelder-Mead algorithm to 50 as Figure 1 shows that the objective function value doesn't change significantly after 50 iterations.

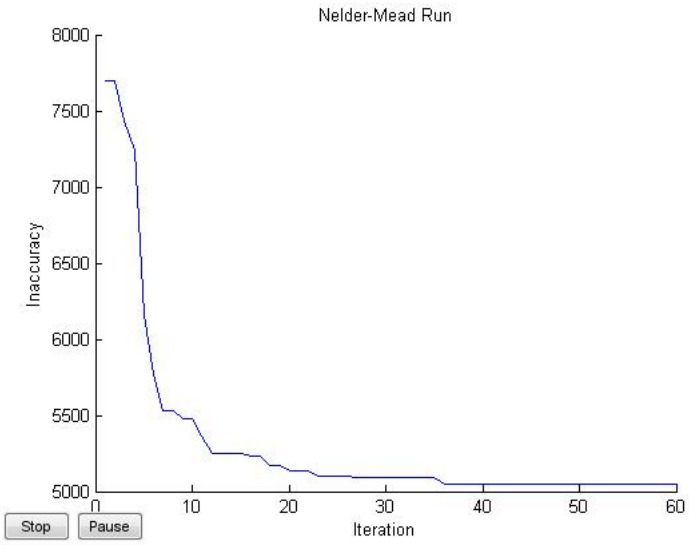

Figure 1: Nelder-Mead run. The objective function value does not change significantly after 50 iterations.

3.2.3 Stopping Condition

We use the mean change in the parameter value as a stopping condition instead of the convergence of the log conditional likelihood due to the fact that calculating the log conditional likelihood requires summing over the whole training set. After each epoch, we check whether the L1 error distance between β_e and β_{e+1},

$$error = \frac{1}{d}\sum_{j=1}^{d} |\beta_{e+1}^j - \beta_e^j|$$

is lower than the preset threshold, ϵ. We set the threshold, $\epsilon = .001$, as we wish to see the average change in the parameter values to be within 3-digit precision.

3.2.4 Parameters Initialization

Since the regularized logistic regression problem is convex, we just have to initialize the parameters such that the weighted sum is near the linear part of the sigmoid function. If the weight sum is big, the sigmoid function saturates and gradient gets closer to 0. For the regularized logistic regression model, it is sufficient to initialize $\beta_j = 0$ for $j = \{1...d\}$.

For the bounded model, more cares must be taken. In section 2.3, we have proved that the bounded logistic regression problem is not convex. So, we must be careful in initializing our parameters to avoid getting stuck in a local

a.

tp=11603	fn=1805
fp=832	tn=2041

b.

tp=11352	fn=1083
fp=1400	tn=2446

Table 1: Confusion matrix for logistic regression. Table a describes the confusion matrix for the regularized logistic regression model. Table b describes the confusion matrix for the regularized bounded logistic regression.

optimum. For parameter c and f , as is shown in (29) and (30), they are convex optimization problem if we fix β_j, so we just randomly assign the value of them and they will reach the global optimum after running the bounded logistic regression model. The initialization of β_j is harder because it isn't a convex optimization problem.

Since the bounded model is similar to the unbounded model, a good initial set of parameters is the one we get from training the unbounded case. For the first run, we use these parameters to establish a baseline estimate for the log conditional likelihood. To address the issue of local minima, we run SGD 100 times with different randomized initial parameters, and return the set with the highest conditional likelihood. We sample β_j uniformly from [-10,10]. We believe that this range is sufficient as all the parameters from the unbounded model is between [-1,1].

3.3 Results

With the appropriate choice of constant c, two learning rate schedules converge at the same log conditional likelihood, however, at different rates. The exponential dampening converges much faster(epoch=6) than inverse dampening(epoch=36). For this reason, we decide to use exponential dampening for the rest of the experiment. Figure 2 shows the two learning rate schedule converges at the same log conditional likelihood, but with different rates.

Table 1 shows the confusion matrix for the test set using regularized logistic regression and the bounded model.

3.4 Discussion

The bounded logistic regression model gives us 84.75% accuracy over the test data, while the logistic regression model gives us 83.70%. We found that upper bound $\sigma(c)$=0.9947 and lower bound $\sigma(f)$=0.1798. So we can see the BLR model can find the bound of the probabilistic distribution more accurate than the LR model. With the bounds in place, the learning algorithm can focus more on the most informative part of the distribution, thus making the search of parameter more efficiently and accurately. As a result, the parameter BLR model learns will fit the training set better.

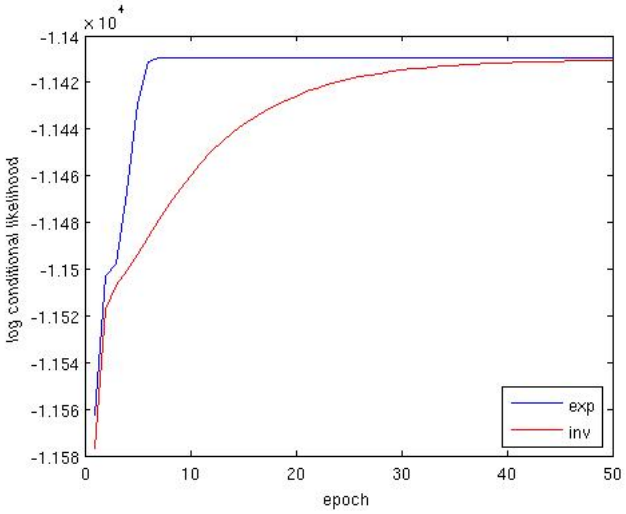

Figure 2: Learning rate schedule comparison
The figure shows the two learning schedule tuned with the best constant c will converges to the same log conditional likelihood. However, the exponential dampening(exp) converges at a much faster rate than the inverse dampening(inv).

3.4.1 Complexity

The time complexity for finding the optimal settings is $O(Nknf)$, where N is the number of iteration allowed for the Nelder-Mead algorithm, k is the number of folds, n is the number of training example, and f is the average number of nonzero x_j per example. The space complexity for optimal settings is $O(d+k)$, where d is the dimension of a training example, as at any point we need to store the β_j's and we have to calculate the average function value for cross-validation.

The parameters training time complexity is $O(nf)$ and the space complexity is $O(d)$ as we only need to store the weight vector.

4 Conclusion

In this paper, we have presented the logistic regression and the bounded logistic regression model and the update rule to learn the model using stochastic gradient descent(SGD). We also described the further cares for using SGD for optimization, such as, feature normalization, learning rate schedule, stopping condition, and parameter initialization. We also show the results of using both models on the UCI Adult dataset. Finally, we show the time and space complexity of our algorithm.

References

[1] Nelder, John A.; R. Mead (1965). "A simplex method for function minimization". Computer Journal 7: 308–313. doi:10.1093/comjnl/7.4.308.

[2] Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. Department of Information and Computer Sciences, University of California, Irvine. http://www.ics.uci.edu/mlearn/MLRepository.html