South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:

Zhenyi Tang

Student ID：201630503026

Supervisor:

Qingyao Wu

Grade: Undergraduate

October 25, 2018

Logistic Regression and Support Vector Machine

Abstract— Logistic Regression and Support Vector Machine Are very common classification methods. This experiment implements both algorithms and evaluates their performance through MBGD-Mini-batch Gradient Descent.

I. INTRODUCTION

This experiment will compare and understand the difference between gradient descent and Mini-batch Gradient Descent And to understand the difference and connection between logistic regression and linear classification. Finally, we will further understand the principles of SVM and practice it on larger data.

II. METHODS AND THEORY

In lab2.1 first need to load the training set and validation set. Initialize logistic regression model parameter (you can consider initializing zeros, random numbers or normal distribution). Select the loss function and calculate its derivation, and the formula is showing below.

$$g\left(z\right)=\frac{1}{1+e^{-z}} \quad (1.1)$$

Determine the size of the batch size and randomly take some

$$h_{\mathbf{w}}(\mathbf{x})=g(\mathbf{w}^{\top}\mathbf{x})=\frac{1}{1+e^{-\mathbf{w}^{\top}\mathbf{x}}} \quad (1.2)$$

samples, then we use another formula to replace z.

$$\frac{\partial J\left(\mathbf{w}\right)}{\partial \mathbf{w}}=\frac{1}{n}\sum_{i=1}^{n}\left(h_{\mathbf{w}}\left(\mathbf{x}_i\right)-y\right)\mathbf{x}_i \quad (1.3)$$

Calculate gradient G toward loss function from partial samples. Use the SGD optimization method to update the

$$\mathbf{w}:=\mathbf{w}-\frac{1}{n}\sum_{i=1}^{n}\alpha\left(h_{\mathbf{w}}\left(\mathbf{x}_i\right)-y_i\right)\mathbf{x}_i \quad (1.4)$$

parametric model finally predict under validation set and get the loss with the number of iterations.

$$J\left(\mathbf{w}\right)=-\frac{1}{n}\left[\sum_{i=1}^{n}y_i\log\,h_{\mathbf{w}}(\mathbf{x}_i)+(1-y_i)\log\,\left(1-h_{\mathbf{w}}(\mathbf{x}_i)\right)\right] \quad (1.5)$$

In lab2.2 fist load the training set and validation set. Initialize SVM model parameters then initializing zeroes. Select the loss function and calculate its derivation as follow

$$\mathcal{L}(\mathbf{w},b,\boldsymbol{\alpha})=\frac{1}{2}\left\|\mathbf{w}\right\|^2+\sum_{i=1}^{n}\alpha_i(1-y_i(\mathbf{w}^T\mathbf{x}_i+b)) \quad (1.6)$$

Determine the size of the batch_size and randomly take some samples, calculate gradient G toward loss function from partial samples. Use the SGD optimization method to update the

$$g_{\mathbf{w}}(\mathbf{x}_i)=\begin{cases}-y_i\mathbf{x}_i & 1-y_i(\mathbf{w}^{\top}\mathbf{x}_i+b)>=0\\ 0 & 1-y_i(\mathbf{w}^{\top}\mathbf{x}_i+b)<0\end{cases} \quad (1.7)$$

$$g_b(\mathbf{x}_i)=\begin{cases}-y_i & 1-y_i(\mathbf{w}^{\top}\mathbf{x}_i+b)>=0\\ 0 & 1-y_i(\mathbf{w}^{\top}\mathbf{x}_i+b)<0\end{cases} \quad (1.8)$$

parametric model Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Predict under validation set and get the loss.

III. EXPERIMENT

A. Dataset

Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features. Please download the training set and validation set.

B. Implementation

For the lab2.1 first load the training set and validation set. Initialize logistic regression model parameter consider initializing zeroes, select the loss function and calculate its derivation. Determine the size of the batch size and randomly take some samples, calculate gradient G toward loss function from partial samples. Use the SGD optimization method to update the parametric model, drawing graph of with the number of iterations. The result is showing in the figure 1.

For the lab2.2 first load the training set and validation set. Initialize SVM model parameters. Select the loss function and calculate its derivation. Determine the size of the batch size and randomly take some samples, calculate gradient G toward loss function from partial samples. Use the SGD optimization method to update the parametric model and get the loss. Draw graph of with the number of iterations.

table I. parameter initial

	value
learning rate	0.01
batch size	30
epoch	8000

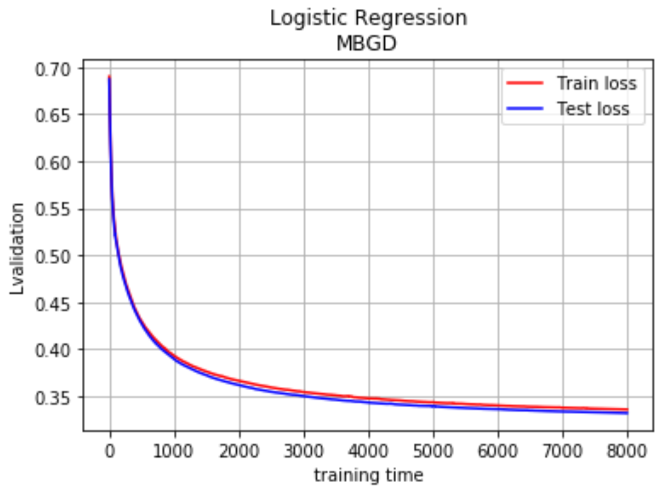

figure 1. the result of LR loss by using MBGD

table II. parameter initial

	value
learning rate	0.01
batch size	50
epoch	1800
C	0.15

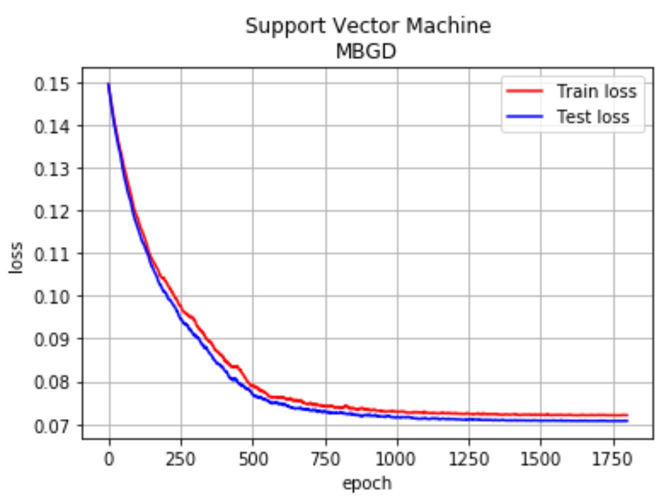

figure 2. the result of SVM loss by using MBGD

IV. CONCLUSION

Through this experiment, I got a better understanding of LR and SVM. At the same time, I tried to use SGD, MBGD and BGD to realize the above two models. Through experiment and comparison, I found that BGD can get good W in less epoch, but because of the need to calculate each sample, the calculation time consumed is more than twice slower than SGD. However, SGD calculates the gradient drop each time it is just randomly looking for a sample to cause the loss curve to be unsmooth and the number of iterations is higher. MBGD is a good solution to the two defects, got a nice W in a shorter calculation time.