South China University of Technology

# The Experiment Report of Machine Learning

**SCHOOL:** SCHOOL OF SOFTWARE ENGINEERING

**SUBJECT:** SOFTWARE ENGINEERING

Author:
Yang Zhang

Supervisor:
Mingkui Tan

Student ID：201630666370

Grade:
Undergraduate

October 14, 2018

# Logistic Regression and Support Vector Machine

***Abstract*— In this experiment, we are supposed to compare and understand the difference between gradient descent and batch random stochastic gradient descent. In addition, we will further understand the principles of SVM and practice on larger data.**

## I. INTRODUCTION

This experiment is the second experiment in machine learning class. To some extent, it is based on the first experiment. This experiment is nearly about logistic regression and support vector machine. Meanwhile, we will use Adam method. As a result, we will compare and understand the differences and relationships between Logistic regression and linear classification

## II. METHODS AND THEORY

### *A. Logistic Regression*

In statistics, logistic regression, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical. This experiment covers the case of a binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick.

The logistic function σ(t) is defined as follows:

$$\sigma(t) = \frac{e^t}{e^t+1} = \frac{1}{1+e^{-t}} \tag{1}$$

For binary labels: $y = \{0, 1\}$

$$h_w(x) = g(w^T x) = \frac{1}{1+e^{-w^T x}} \tag{2}$$

The loss function is:

$$J(w) = -\frac{1}{m}[\sum_{i=1}^{m} y_i log h_w(x_i) + (1-y_i) log(1-h_w(x_i))] \tag{3}$$

The derivation of loss function is:

$$\frac{\partial J(w)}{\partial w} = -y\frac{1}{h_w(x)}\frac{\partial h_w(x)}{\partial w} + (1-y)\frac{1}{1-h_w(x)}\frac{\partial h_w(x)}{\partial w} \tag{4}$$

### *B. Adam*

Adaptive estimates of lower-order moments. Has both the advantage of AdaGrad and RMSProp.

$$\begin{aligned} g_t &\leftarrow \nabla J(\theta_{t-1}) \\ m_t &\leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t \\ G_t &\leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t \\ \alpha &\leftarrow \eta \frac{\sqrt{1-\gamma^t}}{1-\beta^t} \\ \theta_t &\leftarrow \theta_{t-1} - \alpha \frac{m_t}{\sqrt{G_t+\epsilon}} \end{aligned} \tag{5}$$

### *C. Linear Classification*

In the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. The hinge loss is used for maximum-margin classification, most notably for support vector machines (SVMs). It's defined as follows:

$$\ell(y) = \max(0, 1 - t \cdot y) \tag{6}$$

The derivation of hinge loss is:

$$\frac{\partial \ell}{\partial w_i} = \begin{cases} -t \cdot x_i & \text{if } t \cdot y < 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The loss function of SVM is:

$$J(w) = \frac{1}{2}\|w\|^2 + C\sum_i \max(0, 1 - y_i(w_i^x + b)) \tag{8}$$

### *D. SGD*

Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is an iterative method for optimizing a differentiable objective function, a stochastic approximation of gradient descent optimization. Machine learning consider the problem of minimizing an objective function that has the form of a sum:

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w) \tag{9}$$

When used to minimize the above function, a standard (or "batch") gradient descent method would perform the following iterations :

$$w := w - \eta \nabla Q(w) = w - \eta \sum_{i=1}^{n} \nabla Q_i(w)/n \tag{10}$$

## III. EXPERIMENT

### *A. Dataset*

Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features.

### *B. Experiment*

*(1) Logistic Regression and Batch Stochastic Gradient Descent*

1. Load the training set and validation set.
2. Initialize logistic regression model parameter.
3. Select the loss function and calculate its derivation.
4. Determine the size of the batch_size and randomly take some samples, calculate gradient G toward loss function from partial samples.
5. Use the SGD optimization method to update the parametric model and encourage additional attempts to optimize the Adam method.
6. 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 *Lvalidation*.

7. Repeat step 4 to 6 for several times, and drawing graph of *Lvalidation* with the number of iterations.

*(2) Linear Classification and Batch Stochastic Gradient Descent*

1. Load the training set and validation set.
2. Initialize SVM model parameters.
3. Select the loss function and calculate its derivation.
4. Determine the size of the batch_size and randomly take some samples, calculate gradient G toward loss function from partial samples.
5. Use the SGD optimization method to update the parametric model and encourage additional attempts to optimize the Adam method.
6. 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 *Lvalidation*.
7. Repeat step 4 to 6 for several times, and draw graph of *Lvalidation* with the number of iterations.

*C. Result*

*(1) Logistic Regression and Batch Stochastic Gradient Descent*

In this experiment, the epoch is 600 and batch_size is 300s. The result has been shown below.

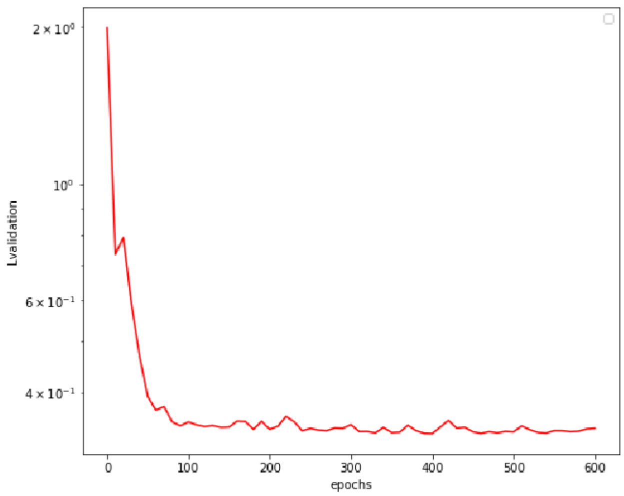


Fig. 1. Result of experiment I

*(2) Linear Classification and Batch Stochastic Gradient Descent*

In this experiment, the epochs is 600 and batch_size is 256. The result has been shown below.

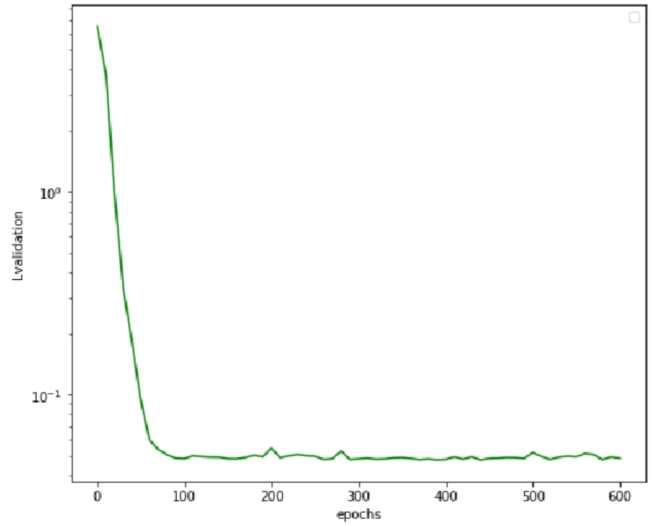


Fig. 2. The result of experiment II

## IV. CONCLUSION

The experiment is even more interesting than the first one. It's a bit difficult to calculate the derivation of loss functions of logistic regression and support vector machine. To be honest, this experiment costs me a host of time. You can not be too careful when you do this project, because a little error in your code can ruin your whole project. Therefore, in the following experiments, I will pay more attention to the each step, straining every nerve to accomplish the experiment well.