South China University of Technology

The Experiment Report of Machine Learning

School: School of Software Engineering

Subject: Software Engineering

Author:
Yuehao Xiao

Supervisor:
Mingkui Tan

Student ID:
201630719984

Grade:
Undergraduate

October 25, 2018

Logistic Regression and Support Vector Machine

Abstract—Two different methods for classification, logistic regression and linear classification

I. Introduction

In this report, we will try two methods for classification, logistic regression and linear classification, and use the SGD optimization method.

II. Methods and Theory

In logistic regression, we will randomly take some samples, and calculate gradient G by calculate the loss function and calculate its derivation, and use the sigmoid function to adjust the predict scores to (0,1). And then get the loss.

In linear classification, we will also randomly take some samples, and calculate gradient G by calculate the loss function and calculate its derivation, but the loss function is totally different with the former, for using SVM model. And we will use a judgment formula to mark the sample whose predict scores are positive, and whose and negetive.

III. Experiments

A. Dataset

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

B. Logistic Regression

With samples:

$$x_1, x_2 \dots x_n$$

Desired outputs:

$$y_i, y_2 \dots y_n$$

Hypothesis:

$$h_\omega(x) = g(\omega^T X)$$

Logistic function:

$$g(z) = \frac{1}{1+e^{-z}}$$

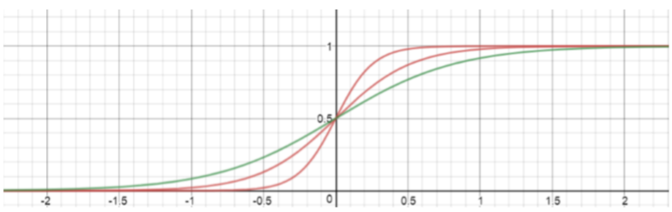

Fig. 1. Logistic function

If $z \to +\infty$, then $g(z) \to 1$; If $z \to -\infty$, then $g(z) \to 0$.

So, if $h_\omega(X) = g(\omega^T X)$ closely captures $\mathbb{P}[+1|X]$:

$$P(y|X) = \begin{cases} g(\omega^T X) & y = 1 \\ 1 - g(\omega^T X) = g(-\omega^T X) & y = -1 \end{cases} \tag{1}$$

Or, more compactly:

$$P(y|X) = g(y \cdot \omega^T X)$$

Loss function and Regularization:

$$J(\omega) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i \cdot \omega^T X_i}) + \frac{\lambda}{2}\|\omega\|_2^2$$

Update parameters with rate η:

$$\omega' \to \omega - \eta\frac{\partial J(\omega)}{\partial \omega} = (1-\eta\lambda)\omega + \eta\frac{1}{n}\sum_{i=1}^{n}\frac{y_i X_i}{1+e^{y_i \cdot \omega^T X_i}}$$

Log-likehood loss function:

$$J(\omega) = -\frac{1}{n}[\sum_{i=1}^{n} y_i \log h_\omega(X_i) + (1-y_i)\log(1-h_\omega(X_i))]$$

The Gradient of The Loss Function:

$$\frac{\partial J(\omega)}{\partial \omega} = (h_\omega(X) - y)X$$

$$\omega := \omega - \frac{1}{n}\sum_{i=1}^{n}\eta(h_\omega(x_i) - y_i)x_i$$

Result:

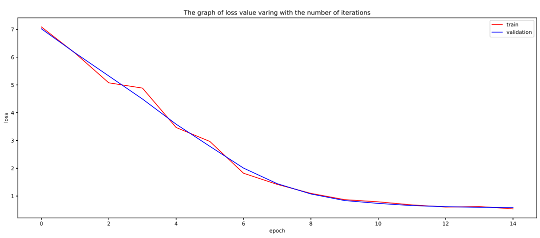

Fig. 2. Result of training and validation

C. Linear Classification

With samples:

$$x_1, x_2 \dots x_n$$

Desired outputs:

$$y_i, y_2 \dots y_n$$

Loss function:

$$\mathcal{L}(\omega) = \frac{\|\omega\|_2^2}{2} + C\sum_{i=1}^{N}\max(0, 1 - y_i(\omega^T x_i + b))$$

Gradient computation:
Let
$$g_\omega(x_i) = \frac{\partial \max(0, 1 - y_i(\omega^T x_i + b))}{\partial \omega}$$
If $1 - y_i(\omega^T x_i + b) \geq 0$:
$$g_\omega(x_i) = -y_i x_i$$
If $1 - y_i(\omega^T x_i + b) < 0$:
$$g_\omega(x_i) = 0$$
At last we have:
$$\frac{\partial f(\omega, b)}{\omega} = \omega + C \sum_{i=1}^{N} g_\omega(x_i)$$
And:
$$\frac{\partial f(\omega, b)}{b} = C \sum_{i=1}^{N} g_b(x_i)$$
Result:

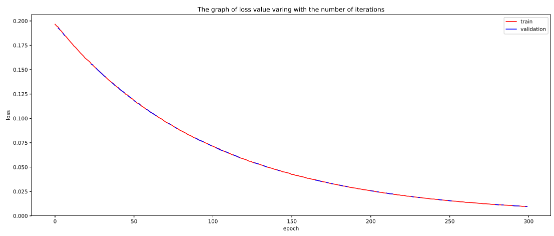

Fig. 3. Result of training and validation

IV. Conclusion

In this report, we conduct some experiments under small LIBSVM datasets, compared and understood the difference between gradient descent and batch random stochastic gradient descent, logistic regression and linear classification. Finally, we have further understand of the principles of SVM and practice on larger data.