华南理工大学

South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Haiying zhou

Supervisor:
Mingkui Tan

Student ID:
201630666547

Grade:
Undergraduate

October 26, 2018

Logistic Regression and Support Vector Machine

***Abstract*—The experiment includes two parts: LogisticRegression and Batch Stochastic Gradient Descent and Linear Classification and Batch Stochastic Gradient Descent.**

I. Introduction

This experiment intends to use LinearRegression and Stochastic Gradient Descent respectively for conducting some experiments under small scale data sets and realizing the process of optimization and adjusting parameters.

II. Methods and Theory

A. Logistic Regression and Batch Stochastic Gradient Descent

- Hypothesis Function

$$h_w(x) = g(\sum_{i=1}^{m} w_i x_i) = g(W^\top x)$$

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

- Loss function

$$J(w) = -\frac{1}{n}\left[\sum_{i=1}^{n} y_i \log h_w(x_i) + (1-y_i)\log(1-h_w(x_i))\right]$$

- The Gradient of The Loss Function
  For a sample:

$$\frac{\partial J(w)}{\partial w} = (h_w(x) - y)x$$

  For all samples:

$$\frac{\partial J(w)}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}(h_w(X_i) - y)x_i$$

- Update w

$$w := w - \frac{1}{n}\sum_{i=1}^{n}\alpha(h_w(x_i) - y_i)x_i$$

B. Linear Classification and Batch Stochastic Gradient Descent

- Linear Classifier Form

$$f(x) = w^\top x + b$$

- Hinge loss

$$\xi_i = \max(0, 1 - y_i(x^\top x_i + b))$$

- Target

$$\min_{w,b} f : \frac{\|w\|^2}{2} + C\sum_{i=1}^{n}\max(0, 1 - y_i(w^\top x_i + b))$$

- SGD
  Randomly select an example i in the training set:

$$w := w - \eta\nabla L_i(w)$$

- Gradient
  - g_w

$$g_w(x_i) = \begin{cases} -y_i x_i & 1 - y_i(w^\top x_i + b) >= 0 \\ 0 & 1 - y_i(w^\top x_i + b) < 0 \end{cases}$$

  - g_b

$$g_b(x_i) = \begin{cases} -y_i & 1 - y_i(w^\top x_i + b) >= 0 \\ 0 & 1 - y_i(w^\top x_i + b) < 0 \end{cases}$$

  - We have...

$$\frac{\partial f(w,b)}{\partial w} = w + C\sum_{i=1}^{N} g_w(x_i)$$

$$\frac{\partial f(w,b)}{\partial b} = C\sum_{i=1}^{N} g_b(x_i)$$

III. Experiments

A. Dataset

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

B. Implementation

1) Logistic Regression and Batch Stochastic Gradient Descent:

i) Load the training set and validation set.

```
import requests
import numpy
r=requests.get('''https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/a9a''')
t=requests.get('''https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/a9a.t''')
```

ii) Initialize logistic regression model parameter.

```
batch_size=64
max_epoch=2000
learning_rate=0.01
losses_train=[]
losses_val=[]
n_features=123
```

iii) Select the Loss function and calculate its derivation.

```
def sigmoid(x):
return 1 / (1 + numpy.exp(-x))
def loss(a,b):
return -1/b.shape[0] * (b*numpy.log(sigmoid(a))+(1-b)*numpy.log(1-sigmoid(a))).sum()
```

iv) Determine the size of the batch_size and randomly take some samples,calculate gradient G toward loss function from partial samples.

```
batch_sample=numpy.random.choice(x_train.
    shape[0],batch_size)
x=x_train[batch_sample]
y=y_train[batch_sample]
G=1/batch_size*numpy.dot((sigmoid(x@w)-y).
    T,x).T
G=-G
w+=learning_rate*G
```

v) Use the SGD optimization method to update the parametric model and encourage additional attempts to optimize the Adam method.

vi) 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 $Lvalidation$.

```
y_predict=x_val.dot(w)
a=sigmoid(y_predict)
losses_val.append(loss(y_predict,y_val)
```

2) Linear Classification and Batch Stochastic Gradient Descent:

i) Load the training set and validation set.

ii) Initialize SVM model parameters.

```
batch_size=64
max_epoch=2000
learning_rate=0.01
C=0.5
```

iii) Select a Loss function and calculate its derivation.

```
def loss(a,b):
e=np.maximum(0,(1-a*b))
return ((C*e.sum())+0.5*np.dot(w.transpose
    (),w).sum()) / b.shape[0]
```

iv) Determine the size of the batch_size and randomly take some samples,calculate gradient G toward loss function from partial samples.

```
batch_sample=np.random.choice(x_train.
    shape[0],batch_size)
x=x_train[batch_sample]
y=y_train[batch_sample]
y[1 - y * (x .dot(w)) < 0] = 0
G=(-1)*(x.T.dot(y))/y.shape[0]+w
G=-G
w+=learning_rate*G
```

v) Use the SGD optimization method to update the parametric model and encourage additional attempts to optimize the Adam method.

vi) 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 $Lvalidation$.

```
val_predict=x_val.dot(w)
losses_val.append(loss(val_predict,y_val))
```

C. Result

1) Linear Regression and Batch Stochastic Gradient Descent: we can see the result of Linear Regression and Batch Stochastic Gradient Descent according to figure 1

2) Linear Classification and Batch Stochastic Gradient Descent: we can see the result of Linear Regression and Batch Stochastic Gradient Descent according to figure 2

D. Analysis

From experience's result, we can find that the gradient descent is very smooth, which means that the descent process meets the requirements of this lab.

IV. Conclusion

This experience is a little bit more difficult to calculate the derivation of loss functions of logistic regression and support vector machine than the last one.But by consulting various documents and communicating with partners, not only do I compare and understand the difference between gradient descent and batch random stochastic gradient descent, but also I understand the differences and relationships between Logistic regression and linear classification.What' more, I further understand the principles of SVM and practice on larger data.

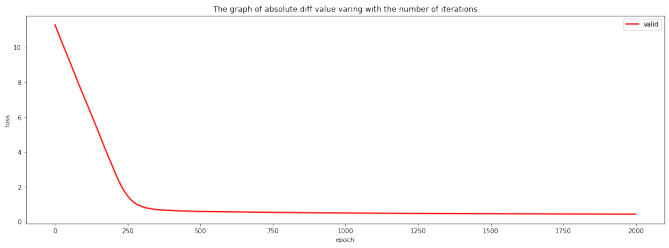

Fig. 1. The loss validation with the number of iterations.

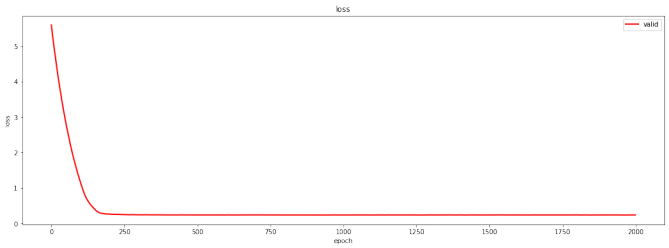

Fig. 2. The loss validation with the number of iterations.