South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCUT

SUBJECT: SOFTWARE ENGINEERING

Author: Yinqi Zhang	Supervisor: Mingkui Tan
Student ID：201630666394	Grade: Grade 2016

October 28, 2018

Linear Regression and Stochastic Gradient Descent

Abstract：

This article focuses on binary classification, and provide two method to solve this kind problem——Logistic Regression and SVM.

Introduction:

When we have a binary classification dataset, we want to find a function to classify other data to the right classification. We can use Logistic Regression or SVM to solve this kind of problem.

Method and theory:

1、Logistic Regression:

Binary classification problem often can't solve using linear model.So we introduce the Sigmoid function, which can change the range to [0,1].

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

If we using the loss method like linear regression, it will appear a lot of local optima.So we use another way to calculate the loss——Maximum Likelihood Estimate.Because we have used Sigmoid function, we can see $\phi(z)$ as the posterior probability of class 1.

$$P(y|x;w) = \phi(z)^y(1-\phi(z))^{1-y}$$

Using Maximum Likelihood Estimate, we can get:

$$L(w) = \prod_{i=1}^{n} P(y|x;w) \quad (10)$$

$$= \prod_{i=1}^{n} \phi(z)^y(1-\phi(z))^{1-y} \quad (11)$$

After &L(w)& formula changes a little, we can define the cost function:

$$J(w) = -\ln L(w) \quad (12)$$

$$= -\sum_{i=1}^{n}(y^{(i)}\ln\phi(z^{(i)}) + (1-y^{(i)})\ln(1-\phi(z^{(i)}))) \quad (13)$$

Then we can use gradient to optimize the parameter.

$$w := w + \Delta w \qquad \Delta w = -\eta \Delta J(w)$$

$$\frac{\partial J(w)}{\partial w_j} = -\sum_{i=1}^{n}(y^{(i)}\frac{1}{\phi(z^{(i)})} - (1-y^{(i)})\frac{1}{1-\phi(z^{(i)})})\frac{\partial \phi(z^{(i)})}{\partial w_j}$$
(1)

$$= -\sum_{i=1}^{n}(y^{(i)}\frac{1}{\phi(z^{(i)})} - (1-y^{(i)})\frac{1}{1-\phi(z^{(i)})})\frac{\partial \phi(z^{(i)})}{\partial z^{(i)}}\frac{\partial z^{(i)}}{\partial w_j}$$
(2)

$$= -\sum_{i=1}^{n}(y^{(i)}\frac{1}{\phi(z^{(i)})} - (1-y^{(i)})\frac{1}{1-\phi(z^{(i)})})\phi(z^{(i)})(1-\phi(z^{(i)}))\frac{\partial z^{(i)}}{\partial w_j}$$
(3)

$$= -\sum_{i=1}^{n}(y^{(i)}(1-\phi(z^{(i)})) - (1-y^{(i)})\phi(z^{(i)}))x_j^{(i)}$$
(4)

$$= -\sum_{i=1}^{n}(y^{(i)} - \phi(z^{(i)}))x_j^{(i)}$$
(5)

Then we can use gradient descent method to update parameter:

$$w_j = w_j + \eta\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \phi(z^{(i)}))x_j^{(i)}$$

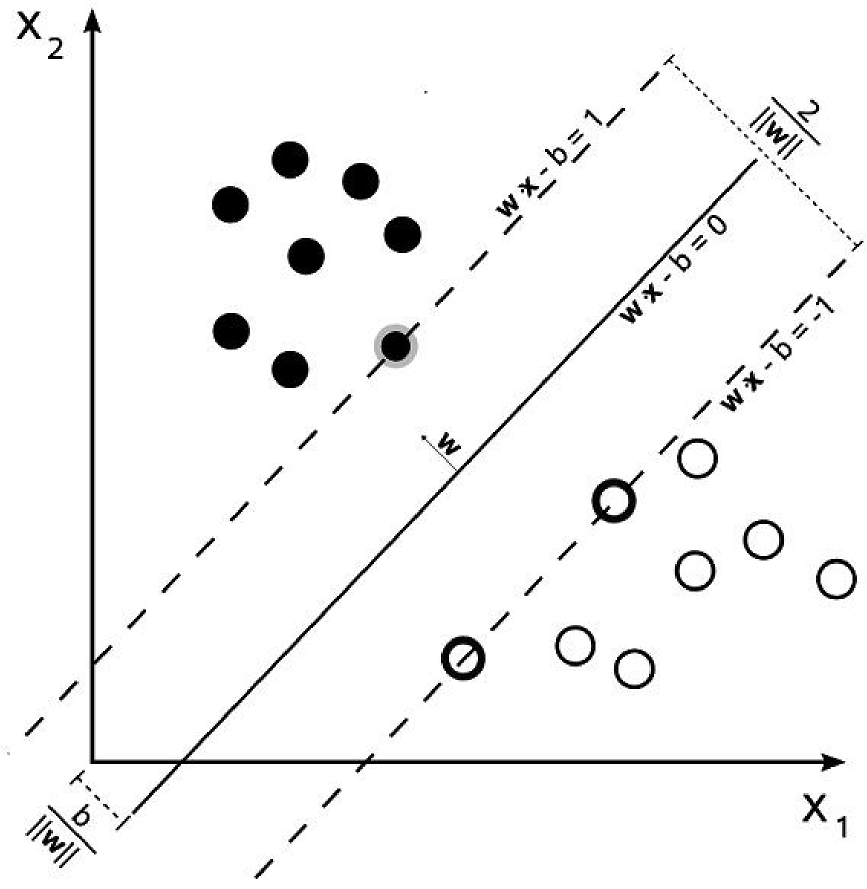
That’s all the derivation of Logistic Regression.

2、*SVM:*

For a lot of data, there are two kind of dataset——linearly separable and not linearly separable.A linear classifier has the form:

$$f(X) = W^T X + b$$

A good classification will be best one when the margin is max.



And the margin is:

$$\frac{w}{\|w\|}(x_+ - x_-) = \frac{w^T(x_+ - x_-)}{\|w\|} = \frac{2}{\|w\|}$$

We often have some trade off in this kind of problem. We often use soft margin formulation.The SVM can be formulated as an optimization:

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

The cost function is

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

Using batch gradient descent method to optimism the parameter:

$$w := w + \Delta w \qquad \Delta w = -\eta\, grand\, w = -\eta(w - CX^T y)$$

That’s all the derivation of SVM.

Experiment:

Dataset:

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

Implementation:

1、Logistic Regression:

The implementation of the algorithm is:

First we get the dataset from Internet and change the form of the data .

Then we initialize the variable and initialize the parameter using different method:

```
learning_rate = 0.01
max_epoch = 300
batch_size=1000
losses_train = []
losses_val = []
#w = numpy.zeros((n_features + 1, 1))    # initialize with zeros
w = numpy.random.random((n_features + 1, 1))    # initialize with random numbers
# w = numpy.random.normal(1, 1, size=(n_features + 1, 1))    # initialize with zero normal
distribution
```

Then update the parameter using SGD, and calculate the loss of the validation set:

```
for epoch in range(max_epoch):
        X_batch, y_batch = sample(X_train, y_train, batch_size)
        G=numpy.dot(X_batch.transpose(),(sigmoid(numpy.dot(X_batch,w))-y_batch))/batch_si
ze
        w=w-learning_rate*G

        z_predict = numpy.dot(X_val, w)
        loss_val=-(numpy.dot(y_val.transpose(),numpy.log(sigmoid(z_predict)))+numpy.dot((1-
y_val.transpose()),numpy.log(1-sigmoid(z_predict))))/y_val.shape[0]
        losses_val.append(loss_val[0])
```

Lastly, we calculate the loss of validation set.

2、SVM

First we get the dataset from Internet and change the form of the data .

Then we initialize the variable and initialize the parameter using different method:

```
learning_rate = 0.003
max_epoch = 800
batch_size=500
C=0.5
losses_train = []
losses_val = []
# w = numpy.zeros((n_features + 1, 1))    # initialize with zeros
w = numpy.random.random((n_features + 1, 1))    # initialize with random numbers
# w = numpy.random.normal(1, 1, size=(n_features + 1, 1))    # initialize with zero normal
distribution
```

Then update the parameter:

```
for epoch in range(max_epoch):
        X_batch, y_batch = sample(X_train, y_train, batch_size)# 取 batch
        temp=1-y_batch*numpy.dot(X_batch,w)
        st=numpy.where(temp>0,y_batch,0)
        w-=learning_rate*(w-C*numpy.dot(X_batch.transpose(),st))

        temp_val = 1-y_val*numpy.dot(X_val,w)
        st_val=numpy.where(temp_val>0,temp_val,0)
        loss_val=(numpy.sum(w*w)/2+C*numpy.sum(st_val))/n_val_samples
        losses_val.append(loss_val)
```

Lastly, print the graph of absolute diff value varing with the number of iterations.

Result:

1、Logistic Regression:

(1) The way to initialize parameter:

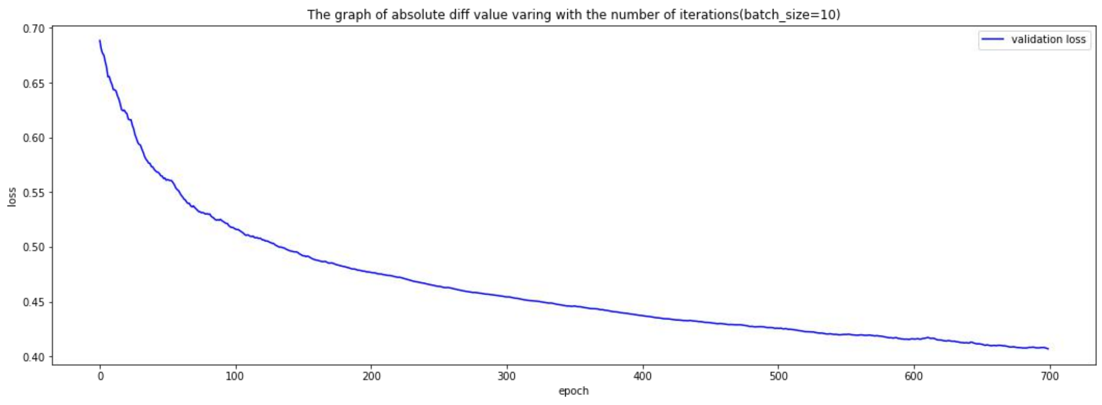
Like experiment_1, the choice of initialize the paramenter w doesn't have too much impact of the result.(set all parameter into zero, initialize it randomly or with normal distribution)

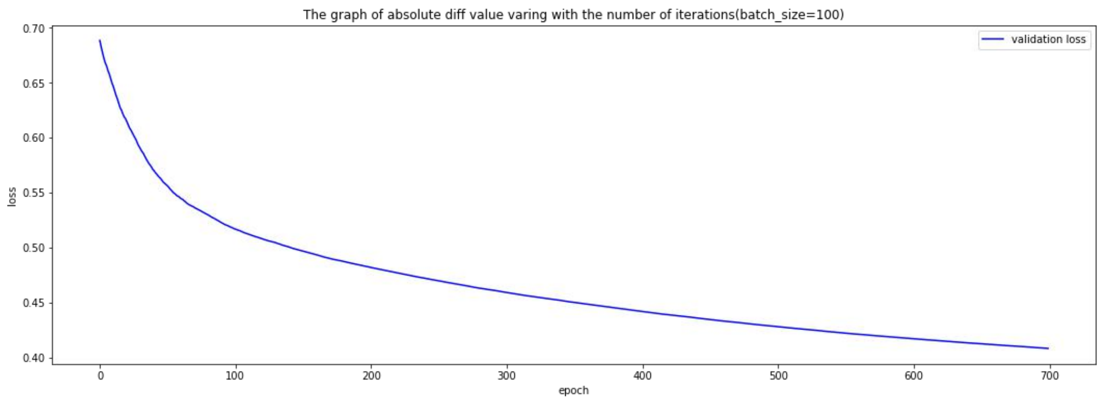
(2) Learning Rate:

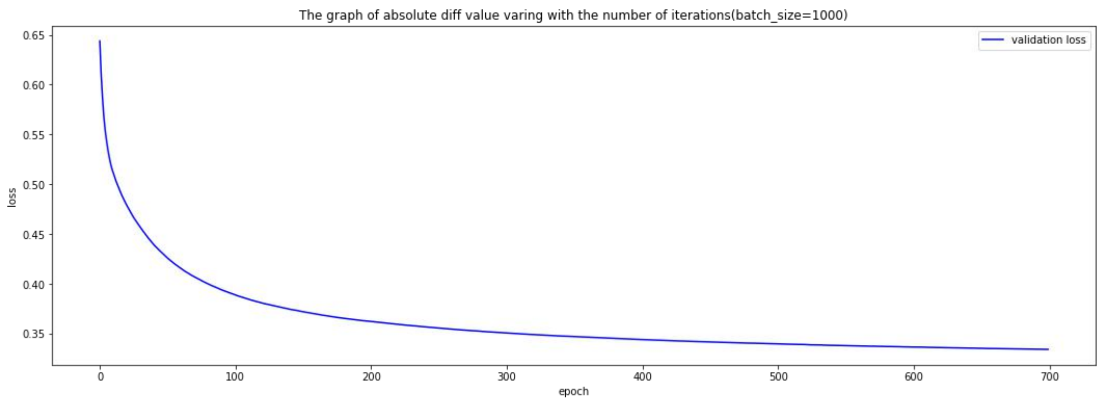
Like experiment_1, bigger the learning rate is, greater the curve fluctuate, faster the image converges.

(3) Batch size:

Bigger batch size is, faster the image converges, greater the curve fluctuate.







(4) Threshold:

I chose 0.5 as the threshold.(I have already change the y vector to [0,1])

2、SVM

(1)The way to initialize parameter:

Like experiment_1, the choice of initialize the paramenter w doesn't have too much impact of the result.(set all parameter into zero, initialize it randomly or with normal distribution)

(2)Learning Rate:

Like experiment_1, bigger the learning rate is, greater the curve fluctuate, faster the image converges.
(3)Batch size:
Like Bigger batch size is, faster the image converges, greater the curve fluctuate.
(5) Threshold:I chose 0 as threshold.

Conclusion:

For binary classification problem, there are two methods to solve it——Logistic Regression and SVM. Logistic Regression have assume that the dataset obey the Logistic distribution, but SVM have no assumption.So if we have a dataset, and we have no information about it. It's a good choice to use Logistic Regression. If we have already knew that dataset obey the Logistic distribution, it's a better choice SVM.

This experiment talked about binary classification providing two method——Logistic Regression and SVM.I learned a lot during the implementation of those two algorithms.I have a better understanding about classification problem and machine learning.