

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **袁强**

**Student ID 201530613511**

**E-mail 1371647899@qq.com**

**Tutor**   **谭明奎**

**Date submitted** **2017. .**

**1. Topic:**  Logistic Regression, Linear Classification and Stochastic Gradient Descent

**2. Time:** 2017/12/2

**3. Reporter:** 袁强

**4. Purposes:**

a. Compare and understand the difference between gradient descent and stochastic gradient descent.

b. Compare and understand the differences and relationships between Logistic regression and linear classification.

c. Further understand the principles of SVM and practice on larger data.

**5. Data sets and data analysis:**

The training data set has 32561 samples and 123 features, and the testing data set has 16281 samples and 123 features

**6. Experimental steps:**

1. Load the training set and validation set.

2. Initialize logistic regression and linear classifier model parameters by setting them up to zeros.

3. Select loss function respectively.

4. Calculate gradient toward loss function from partial samples.

5. Update model parameters using different optimized methods(NAG，RMSProp，AdaDelta and Adam).

6. Select threshold 0.5, then mark the sample whose predict scores greater than it as positive, on the contrary as negative. Predict under validation set and get the different optimized method loss ，， and .

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

**7. Code:**

Logistic Regression:

def gradient(X, y, w,λ=0.1):

row,col= w.shape

grad = - y \* X /(1 + np.exp(y \* X.dot(w)))

return np.mean(grad, axis=0).reshape(row,col)+w\*λ

for i in range(T):

g = gradient(x, y, w - γ\* v)

v = γ \* v + η \* g

w = w - v

Linear Classifier:

def gradient(X, y, w, λ=0.01):

row, col = X.shape

grad=np.zeros((row,col))

b = y \* (X.dot(w).reshape(row,1))<1

b = b.ravel()

grad[b,:] = - X[b,:] \* y[b,:]

grad\_average = np.mean(grad, axis=0)

return grad\_average.reshape(col,1) + λ\*w

for i in range(T):

g = gradient(x, y, w - γ\* v, λ = λ)

v = γ \* v + η \* g

w = w - v

(Fill in the contents of 8-11 respectively for logistic regression and linear classification)

**8. The initialization method of model parameters:**

Set all parameters up to zeros

**9. The selected loss function and its derivatives:**

Logistic Regression:

Loss function:

Derivatives:

Linear Classifier:

Loss function:

Derivatives:

**10. Experimental results and curve:**(Fill in this content for various methods of gradient descent respectively)

## Hyper-parameter selection:

## Predicted Results (Best Results):

## Loss curve:

**11. Results analysis:**

**12. Similarities and differences between logistic regression and linear classification：**

**13. Summary:**