



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:  
Juncong Huang

Supervisor:  
Mingkui Tan

Student ID: 201530611746

Grade:  
Undergraduate

December 14, 2017

# Linear Regression, Linear Classification and Stochastic Gradient Descent

## I. INTRODUCTION

The motivations of the second experiment are as follows. Firstly, Compare and understand the difference between gradient descent and stochastic gradient descent. Secondly, Compare and understand the differences and relationships between Logistic regression and linear classification. Thirdly, Further understand the principles of SVM and practice on larger data.

## II. METHODS AND THEORY

**SGD:** a gradient descent algorithm may be difficult to use when the training data set is large. The computational cost of each iteration linearly increases with the number of samples. Therefore, we take the SGD:

$$\nabla f_B(x) = \frac{1}{|B|} \sum_{i \in B} \nabla f_i(x)$$

The computational cost of each iteration is  $O(|B|)$ . So when the batch size is small, the cost will decrease.

**NAG:** normal gradient descent parameters will move more violently in the vertical direction than in the horizontal direction. Therefore, we take the NAG:

$$v := yv + n \nabla f_B(x)$$

$$x := x - v$$

**Adagrad:** the main idea of Adagrad is that if the partial loss of a model loss function with respect to a parameter element is always large, then its learning rate will drop a little faster; on the contrary, if the partial derivative of a model loss function with respect to a parameter element is always small, then its learning rate will drop a little slow down.

$$s := s + G G^T$$

$$g' := \frac{n}{\sqrt{s+e}} G^T$$

$$x := x - g'$$

**RMSProp:** Adagrad may find it harder to find a useful solution later in the iteration when the learning rate drops faster early in the iteration and the current solution is still not ideal. So RMSProp improves it.

$$s := ys + (1-y) G G^T$$

$$g' := \frac{n}{\sqrt{s+e}} G^T$$

$$x := x - g'$$

It ensures the learning rate will decrease and increase possibly during the iteration

**Adadelta:** it don't need the learning rate, though it is the same idea of RMSProp.

$$s := ps + (1-p) G G^T$$

$$g' = \frac{\sqrt{\Delta X + e}}{\sqrt{s+e}} G^T$$

$$\Delta X := p \Delta X + (1-p) g' g'^T$$

$$x := x - g'$$

**Adam:** we take a momentum variable  $v$  and a exponentially wighted moving average variable  $s$  in RMSprop. In each iteration, we compute the momentum variable  $v$  and wighted moving average variable  $s$ . Additionally, we use the deviation correction to avoid the effect of initializing the variables to 0.

$$t := t + 1$$

$$v := \beta_1 v + (1 - \beta_1) g$$

$$s := \beta_2 s + (1 - \beta_2) G G^T$$

$$v' = \frac{v}{1 - \beta_1^t}$$

$$s' = \frac{s}{1 - \beta_2^t}$$

$$g' := \frac{nv'}{\sqrt{s'+e}}$$

$$x := x - g'$$

## III. EXPERIMENT

### A. Dataset

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

### B. Implementation

#### Regression:

Fig. 1. NAG initializing parameter

Weight parameter	w=zeros
Momentum parameter	u=0.8
Momentum variable	v=0
Learning rate	rate=0.01

Fig. 2. RMSEprop initializing parameter

Weight parameter	w=zeros
Exponentially weighted	r=0

moving average variable	
Weighted parameter	$u=0.8$
Stable constant	$\theta=0.000001$
Learning rate	$rate=0.001$

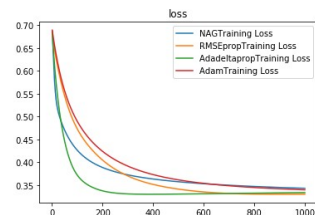
Fig. 3. Adadelta initializing parameter

Weight parameter	$w=zeros$
Exponentially weighted moving average variable	$s=0$
Momentum parameter	$u=0.8$
Stable constant	$\theta=0.000001$
Learning rate	$rate=0.001$
Another exponentially weighted moving average variable	$\delta=0$

Fig. 4. Adam initializing parameter

Weight parameter	$w=zeros$
Momentum variable	$v=0$
Parameter one	$p1=0.9$
Parameter two	$p2=0.999$
Learning rate	$rate=0.001$
Stable constant	$\theta=0.000001$
Exponentially weighted moving average variable	$s=0$
Times	$t=0$

begin  
NAG准确率  
0.838838538665  
RMSEprop准确率  
0.847675284226  
Adadelta准确率  
0.847429519871  
Adam准确率  
0.841287398209



## Classification

Fig. 5. NAG initializing parameter

Weight parameter	$w=zeros$
Momentum parameter	$u=0.8$
Momentum variable	$v=0$
Learning rate	$rate=0.0001$

Punished parameter	$reg=1$
--------------------	---------

Fig. 6. RMSEprop initializing parameter

Weight parameter	$w=zeros$
Exponentially weighted moving average variable	$r=0$
Weighted parameter	$u=0.8$
Stable constant	$\theta=0.000001$
Learning rate	$rate=0.001$
Punished parameter	$reg=1$

Fig. 7. Adadelta initializing parameter

Weight parameter	$w=zeros$
Exponentially weighted moving average variable	$s=0$
Momentum parameter	$u=0.8$
Stable constant	$\theta=0.000001$
Learning rate	$rate=0.0001$
Another exponentially weighted moving average variable	$\delta=0$
Weight parameter	$w=zeros$
Exponentially weighted moving average variable	$s=0$
Punished parameter	$reg=1$

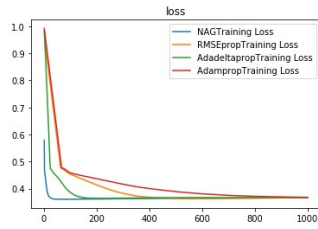
Fig. 8. Adam initializing parameter

Weight parameter	$w=zeros$
Momentum variable	$v=0$
Parameter one	$p1=0.9$
Parameter two	$p2=0.999$
Learning rate	$rate=0.001$
Stable constant	$\theta=0.000001$
Exponentially weighted moving average variable	$s=0$
Times	$t=0$
Punished parameter	$reg=1$

```

begin
NAG准确率
0.844297633352
RMSEprop准确率
0.844112769486
Adadelat准确率
0.844419875929
Adam准确率
0.840550334746

```



#### IV. CONCLUSION

From this experiment, I feel the efficiency of SGD and learn four different methods to find the optimized resolution. I think I could compared the four different methods on various datasets and summarize their strengths and weakness.