華南理工大學

South China University of Technology

# The Experiment Report of Machine Learning

**SCHOOL:** SCHOOL OF SOFTWARE ENGINEERING

**SUBJECT:** SOFTWARE ENGINEERING

Author:
Xingyu Chen

Supervisor:
Mingkui Tan

Student ID：201530611289

Grade:
Undergraduate

December 9, 2017

# Logistic Regression, Linear Classification and Stochastic Gradient Descent

**Abstract—In this experiment we compare the difference between different ways of stochastic gradient descent.We also compare the differences and relationships between logistic regression and linear classification.From this experiment we can get a better understanding of the principles of SVM and the practice on larger data.**

## I. Introduction

Gradient decent(batch gradient decent) and stochastic gradient decent is two major ways to apply on algorithms that use gradient decent.The main difference of this two method is that they use different size of training examples to compute gradient .Batch gradient decent usually use all the training examples while stochastic gradient decent only use one or several examples(called minibatch stochastic gradient decent) to compute gradient and update parameters. The motivation of stochastic gradient decent is that sometimes the data set is too large to compute gradient or it will cost a lot of resources to compute ,in such case we can't simply compute the gradient over the whole training set ,so we randomly chose some examples ,which are also obey the origin distribution .By using this technique we can deal with a great amount of data and just use a small amount of resource to make the gradient decent algorithm work .

Despite the way of selecting examples to compute gradient ,there are also many ways to update parameters.In this experiment ,we compare four different ways of stochastic gradient decent ,namely NAG，RMSProp，AdaDelta and Adam.These four ways are all variation of stochastic gradient decent .

Linear Classification based on SVM and logic regression are the two models we use to apply gradient decent .Logic regression is actually a classification model and the main idea of it is to find a best line the separate the different class of data such that the loss can be minimized. The SVM model is similar to logic regression but SVM can have a better performance by changing the linear non-separable problems in low dimension to the linear separable problem in high dimension .

In this experiment we implements the two models mentioned above and use different gradient decent technique to compare the performance of them .

## II. Methods and Theory

### *A .Linear Regression*

In the linear regression , we implement a binary classifier ,which map the positive example as 1 and negative example as 0. Based on this ,we need a function to map real number into [-1,1] ,so we chose sigmoid function

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

And then we define our hypothesis as

$$\mathrm{h}_\theta(x)=g(\theta^T x+\mathrm{b})=\theta_0 x_0+\theta_1 x_1+\ldots+\theta_{n-1}x_{n-1}+b$$

The $\theta$ denotes the parameter to learn in the model and the $x$ denotes training examples .b is the bias term .With the hypothesis function ,we then define the loss function as

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)}))-(1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]+\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

The term at the tail of formula is the regularization term and $\lambda$ is the regularization parameter .m denotes the number of training examples and j denotes the number of features .For intuition ,the loss function will calculate the loss of examples that are classified to the wrong class .The next step is to compute the gradient of the loss function ,the result is

$$\frac{\partial J(\theta)}{\partial\theta_j}=\left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}\right)+\frac{\lambda}{m}\theta_j$$

Note that in the actual implement we merge b to $\theta$ by adding a dimension to $\theta$ and add a column to training examples ,so the derivative result above will not include b .With all these formula we can use gradient decent to implement linear regression .

### *B . Linear Classification*

In this section ,we illustrate the mathematics in linear classification .We use a different way to represent positive examples and negative examples ,that is ,use 1 to denote positive examples and -1 to denote negative examples .

First we define our hypothesis as

$$\mathrm{f}(x)=w^T x+\mathrm{b}=w_0 x_0+w_1 x_1+\ldots+w_{n-1}x_{n-1}+b$$

Given a training example $(x_i, y_i)$ ,we predict the label as 1 when $f(x_i)\geq 0$ and as 0 when $f(x_i)<0$ .Then we define the objective function (loss function ) as

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

In order to apply gradient decent ,we compute the gradient of the objective function .We denote

$$\xi_i=\max(0,1-y_i(\mathbf{w}^\top\mathbf{x}_i+b))$$

$$g_{\mathbf{w}}(\mathbf{x}_i)=\frac{\partial\xi_i}{\partial\mathbf{w}}$$

If $1-y_i(w^T x_i+b)\geq 0$

$$\begin{aligned} g_{\mathbf{w}}(\mathbf{x}_i) &= \frac{\partial(-y_i(\mathbf{w}^\top \mathbf{x}_i + b))}{\partial \mathbf{w}} \\ &= -\frac{\partial(y_i \mathbf{w}^\top \mathbf{x}_i)}{\partial \mathbf{w}} \\ &= -y_i \mathbf{x}_i \end{aligned}$$

If $1 - y_i(w^T x_i + b) < 0$

$$g_w(x_i) = 0$$

So we have

$$g_{\mathbf{w}}(\mathbf{x}_i) = \begin{cases} -y_i \mathbf{x}_i & 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) >= 0 \\ 0 & 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 0 \end{cases}$$

Then we denote $g_b(x_i) = \frac{\partial \xi_i}{\partial b}$ ,we have the derivative

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

To summarize ,we have

$$\nabla_w L(w,b) = w + \frac{C}{n} \sum_{i=1}^{n} g_w(w_i)$$

$$\nabla_b L(w,b) = \frac{C}{n} \sum_{i=1}^{b} g_b(x_i)$$

Now we can apply gradient decent on linear classification .

*C . Batch Gradient decent and Stochastic Gradient Decent*

In the section we describe how to implement Batch Gradient Decent and Stochastic Gradient Decent .

To illustrate the process of stochastic gradient decent ,we use linear classification model as example .The whole progress are list as figure 1.

```
Initialize parameter $\mathbf{w}$ and learning rate $\eta$
while stopping condition is not achieved do
    Randomly select an example i in the training set
    $\mathbf{w} = \mathbf{w} - \eta \nabla_{\mathbf{w}} L(\mathbf{w}, b)$
    $b = b - \eta \nabla_b L(\mathbf{w}, b)$
end
```

*Figure 1*

The most important thing is that we chose one example to calculate gradient in each iteration until the loss function is converged (namely the stopping condition ) .To contrast ,batch gradient decent uses all the examples to compute gradient and minibatch stochastic gradient decent uses some examples whose number is defined as $S_k$ .

*D .NAG，RMSProp，AdaDelta and Adam*

The four variation of stochastic all have their own motivation and mathematics details ,and in this section we are not going to describe how to derive thesis optimized version of stochastic gradient decent but just list the update rules of each technique .

Figure 2 to figure 5 illustrate the update rules of each technique .

$$\begin{aligned} &\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1} - \gamma \mathbf{v}_{t-1}) \\ &\mathbf{v}_t \leftarrow \gamma \mathbf{v}_{t-1} + \eta \mathbf{g}_t \\ &\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \mathbf{v}_t \end{aligned}$$

*Figure 2 :NAG update rules*

$$\begin{aligned} \mathbf{g}_t &\leftarrow \nabla J(\boldsymbol{\theta}_{t-1}) \\ G_t &\leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t \\ \Delta\boldsymbol{\theta}_t &\leftarrow -\frac{\sqrt{\Delta_{t-1} + \epsilon}}{\sqrt{G_t + \epsilon}} \odot \mathbf{g}_t \\ \boldsymbol{\theta}_t &\leftarrow \boldsymbol{\theta}_{t-1} + \Delta\boldsymbol{\theta}_t \\ \Delta_t &\leftarrow \gamma \Delta_{t-1} + (1-\gamma)\Delta\boldsymbol{\theta}_t \odot \Delta\boldsymbol{\theta}_t \end{aligned}$$

*Figure 3 :AdaDelta update rules*

$$\begin{aligned} \mathbf{g}_t &\leftarrow \nabla J(\boldsymbol{\theta}_{t-1}) \\ \mathbf{m}_t &\leftarrow \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t \\ G_t &\leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t \\ \alpha &\leftarrow \eta \frac{\sqrt{1-\gamma^t}}{1-\beta^t} \\ \boldsymbol{\theta}_t &\leftarrow \boldsymbol{\theta}_{t-1} - \alpha \frac{\mathbf{m}_t}{\sqrt{G_t + \epsilon}} \end{aligned}$$

*Figure 4 :Adam update rules*

$$\begin{aligned} \mathbf{g}_t &\leftarrow \nabla J(\boldsymbol{\theta}_{t-1}) \\ G_t &\leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t \\ \boldsymbol{\theta}_t &\leftarrow \boldsymbol{\theta}_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot \mathbf{g}_t \end{aligned}$$

*Figure 5 :RMSProp update rules*

Note that the $\eta$ in the formula denotes learning rate and $\gamma$ , $\varepsilon$ , $\beta$ are all parameters set by user .

## III. EXPERIMENT

In this part we will describe how we do our experiment in details .We have two model and we apply four ways of stochastic gradient on each model .In order to compare the different methods ,we plot the loss with the number of iterations ,to see how the loss function converged .

To be more specific ,we first load the data and initialize the parameters , then in each iteration we randomly select a batch of examples ,compute gradient on it and use the result to update the parameter in four ways .We record the value of loss function in every iteration and plot it in the end of algorithm .

In the actual important we not only merge the parameter b to parameter $\theta$ in logic regression but also merge the parameter b to parameter w in linear classification .This method can deal with gradient in one time rather than calculate gradient for different parameter .And the result shows that this technique makes no difference in the accuracy of model but accelerate the speed of the model .

The data set we use is a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features.

We initialize the parameter in random values in [0,1] ,the parameters we use are list in table 1 to table 3， the result of our experiment is illustrate in table 4 and figure 6 to figure 7 .

| | learning rate | Number rounds | regularization parameter | C | Batch size |
|---|---|---|---|---|---|
| Linear Regression | 0.1 | 500 | 1 | | 100 |
| Linear Classification | 0.1 | 300 | | 5 | 100 |

*Table 1：model parameter selected in experiment*

| | $\gamma$ | $\varepsilon$ | $\beta$ |
|---|---|---|---|
| NAG | 0.9 | | |
| RMSProp | 0.99 | 1e-8 | |
| AdaDelta | 0.95 | 1e-4 | |
| Adam | 0.999 | 1e-8 | 0.9 |

*Table 2：SGD parameter selected in linear regression*

| | $\gamma$ | $\varepsilon$ | $\beta$ |
|---|---|---|---|
| NAG | 0.9 | | |
| RMSProp | 0.9 | 1e-8 | |
| AdaDelta | 0.95 | 1e-4 | |
| Adam | 0.99 | 1e-6 | 0.9 |

*Table 3：SGD parameter selected in linear classification*

| | *BGD* | *NAG* | *RMS Prop* | *Ada Delta* | *Adam* |
|---|---|---|---|---|---|
| *LR* | *0.7845* | *0.7820* | *0.7613* | *0.7521* | *0.7194* |
| *LC* | *0.8274* | *0.8075* | *0.7818* | *0.7638* | *0.8133* |

*table 4：the classification accuracy of linear regression (LR ) and linear classification (LC) using different gradient decent .*

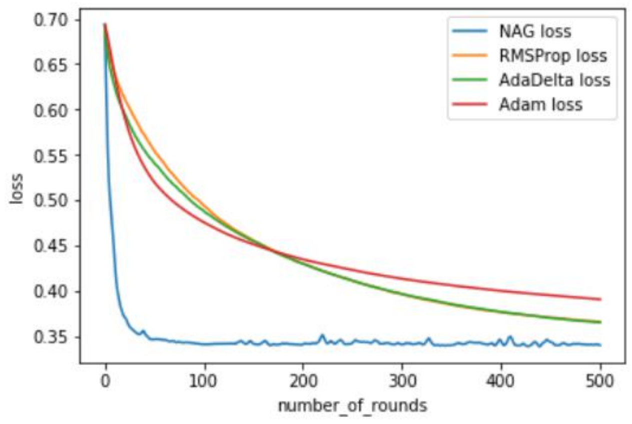

*Figure 6：Linear Regression Loss figure*

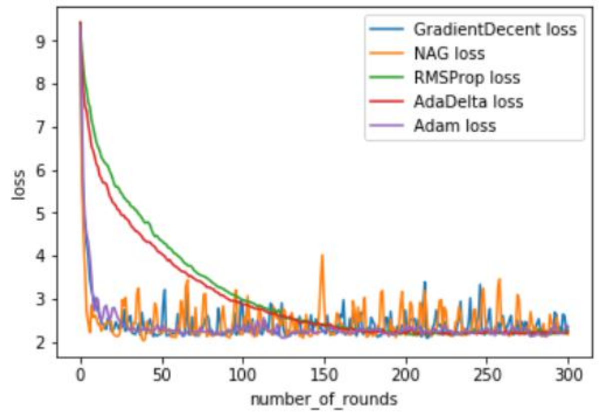

*Figure 7：Linear Classification Loss figure*

From the expense result , we can see that batch gradient decent (BGD) reached the highest accuracy in both two model ,which isn't hard to understand because stochastic gradient decent only use a small number of example to compute gradient so it can't always fit all training examples ,or in other word ,find the best gradient to minimize the objective function .But as we can see ,even BGD outperformed SGD ,SGD methods are already good enough to use not to the saving of computing resource .

As we look closer into the data ,we can find that NAG is sightly better than RMSProp and AdaDelta ,but Adam performed differently in two model .One reasonable explanation is that in the Linear Regression model Adam is stuck in local minimum and failed to leave .As we continue to run the experiment ,these algorithm performed differently due to the random initialization ,but the trends of them are stable .

From the figure we can find another feature of SGD is that ,it converges slower and the loss is up and down as the number of interaction increase .One way to explain this phenomenon is that ,the process of SGD is not smooth ,it can't always find the “best gradient” but chose a direction that close to the “best gradient ”. So SGD will wander around the minimum point but not straightly reach it .Even though the loss curve is not smooth but the trend of SGD shows that it actually converges and the accuracy of model also proves that the algorithm works well .

On last thing to point out is that Linear Classification based on SVM is indeed outperforms Linear Regression ,though two algorithms have similar progress ,the difference in mathematics do makes a difference .Linear Regression is a simple way to implement and is easy to understand but Linear Classification is a more powerful way to apply to classification task with a more complicated base in mathematics .

## IV. CONCLUSION

In this experiment we implement two model and four different SGD algorithm ,compared their performance by plotting loss curve and checking the accuracy .We now can have a conclusion that SGD is a good way compared to BGD and after the step of tuning parameters ,SGD can outperforms BGD with less memory occupation and computing resource .The detail of different SGD algorithm is worth further investigating and we will study on the efficiency of SGD in the coming research .