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:
Weizhao Li

Supervisor:
Mingkui Tan

Student ID:
201530611890

Grade:
Undergraduate

December 14, 2017

Comparison of Various Stochastic Gradient Descent Methods for Solving Classification Problems

***Abstract*—Abstractin this experiment, we compare the difference between NAG, RMSProp, AdaDelta and Adam, these four different optimization methods, under the implementations of logistic regression and linear SVM. The experiment result shows that these methods have different rates to decent, but they will reach similar point to convergence in the end.**

I. Introduction

GRADIENT method is an optimization algorithm, usually called the steepest descent method. The steepest descent method is one of the simplest and oldest methods for solving unconstrained optimization problems. Although it is not practical now, many effective algorithms are based on it and are improved and corrected. The steepest descent method is to use the negative gradient direction for the search direction, steepest descent method closer to the target value, the smaller the step, the slower the progress. We aim to compare NAG, RMSProp, AdaDelta and Adam. We use these four methods to update the models of logistic regression and SVM linear classification.

II. Methods and Theory

A. Logistic regression

In statistics, logistic regression is a regression model where the dependent variable (DV) is categorical. This article covers the case of a binary dependent variable- that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Cases where the dependent variable has more than two outcome categories may be analyzed in multinomial logistic regression, or, if the multiple categories are ordered, in ordinal logistic regression. In the terminology of economics, logistic regression is an example of a qualitative response/discrete choice model. In this experiment, the labels are binary. which means $y_i \in [-1, +1]$. And for all sample, the log-likelihood loss function is

$$L_D(w) = -\frac{1}{n}[\sum_{i=1}^{n} y_i log h_w(x_i) + (1-y_i) log(1-h_w(x_i)) + \frac{\lambda}{2}||w||^2]$$

where

$$h_w(x_i) = \frac{1}{1+e^{-w^T x}}$$

and the gradient of loss function of w is

$$\frac{\partial L(w)}{\partial w} = \frac{1}{n}[(h_w(x) - y)x + \lambda w]$$

B. linear SVM

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. In this experiment, the labels are binary. which means $y_i \in [-1, +1]$.And for all sample, the log-likelihood loss function is

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

and the gradient of loss function respecting to w is

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

the gradient of loss function respecting to b is

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

where

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

and

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

C. optimization methods

Let w_t denotes the parameters in i-th epoch,and ∂L_D denotes the gradient of loss function respecting to w_t, and the procedure of NAG,RMSProp,AdaDelta and Adam are showed below:

1) NAG:

$$g_t \leftarrow \nabla J(\theta_{t-1} - \gamma v_{t-1})$$

$$v_t \leftarrow \gamma v_{t-1} + \eta g_t$$

$$\theta_t \leftarrow \theta_{t-1} - v_t$$

2) RMSProp:

$$g_t \leftarrow \nabla J(\theta_{t-1})$$

$$G_t \leftarrow \gamma G_t + (1 - \gamma) g_t \odot g_t$$

$$\theta_t \leftarrow \theta_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

3) *AdaDelta:*

$$g_t \leftarrow \nabla J(\theta_{t-1})$$

$$G_t \leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t$$

$$g_t \leftarrow \nabla J(\theta_{t-1})$$

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$G_t \leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t$$

$$\alpha \leftarrow \eta \frac{\sqrt{1-\gamma^t}}{1-\beta^t}$$

$$\theta_t \leftarrow \theta_{t-1} - \alpha \frac{m_t}{\sqrt{G_t + \epsilon}}$$

III. Experiments

A. *Dataset*

Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features. Please download the training set and validation set.

B. *Implementation*

1) *Logistic Regression and Stochastic Gradient Descent:*

- Load the training set and validation set.
- Initalize logistic regression model parameters, you can consider initalizing zeros, random numbers or normal distribution.
- Select the loss function and calculate its derivation, find more detail in PPT.
- Calculate gradient G toward loss function from partial samples.
- Update model parameters using different optimized methods(NAGRMSPropAdaDelta and Adam).
- 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 different optimized method loss $L_NAGL_RMSPropL_AdaDelta$ and L_Adam .
- Repeate step 4 to 6 for several times, and drawing graph of $L_NAGL_RMSPropL_AdaDelta$ and L_Adam with the number of iterations.

2) *Linear Classification and Stochastic Gradient Descent:*

- Load the training set and validation set.
- Initalize SVM model parameters, you can consider initalizing zeros, random numbers or normal distribution.
- Select the loss function and calculate its derivation, find more detail in PPT.
- Calculate gradient G toward loss function from partial samples.
- Update model parameters using different optimized methods(NAGRMSPropAdaDelta and Adam).
- 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 different optimized method loss $L_NAGL_RMSPropL_AdaDelta$ and L_Adam.
- Repeat step 4 to 6 for several times, and drawing graph of $L_NAGL_RMSPropL_AdaDelta$ and L_Adam with the number of iterations.

In both logistic regression and linear SVM classification ,we use hold-out method to validate the model .That is we use 'a9a.txt' to train and use 'a9a.t' to predict.Furthermore, we initialize the model parameter into zero.

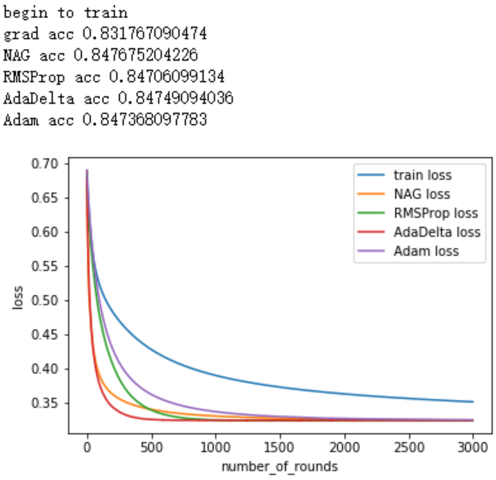
Figures and tables should be labeled and numbered

TABLE I
Logistic Regression Parameters

NAG	$\gamma = 0.9$
RMSProp	$\gamma = 0.9 \quad \epsilon = 1e-9 \quad \eta = 0.001$
AdaDelta	$\gamma = 0.95 \quad \epsilon = 1e-6$
Adam	$\beta_1 = 0.85 \quad \gamma = 0.999 \quad \eta = 1e-3 \quad \epsilon = 1e-8$

TABLE II
Linear SVM Classification Parameters

NAG	$\gamma = 0.9$
RMSProp	$\gamma = 0.9 \quad \epsilon = 1e-9 \quad \eta = 0.001$
AdaDelta	$\gamma = 0.95 \quad \epsilon = 1e-6$
Adam	$\beta_1 = 0.85 \quad \gamma = 0.999 \quad \eta = 1e-3 \quad \epsilon = 1e-8$

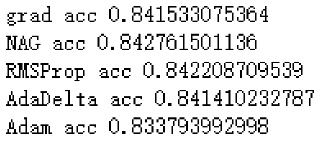


```
begin to train
grad acc 0.831767090474
NAG acc 0.847675204226
RMSProp acc 0.84706099134
AdaDelta acc 0.84749094036
Adam acc 0.847368097783
```

Fig. 1. Logistic Regression Result Figure

IV. Conclusion

In this experiment, through implementing these four different optimization algorithms on logistic regression and linear SVM classification, we can summarize that different optimization algorithms have different performance on rate .RMSProp and Adam decent with a slower rate than the other two. NAG and AdaDelta have similar decent rate. About the accuracy, the



```
grad acc 0.841533075364
NAG acc 0.842761501136
RMSProp acc 0.842208709539
AdaDelta acc 0.841410232787
Adam acc 0.833793992998
```

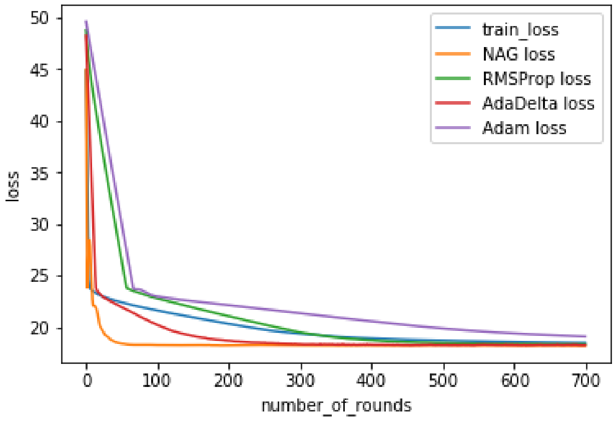


Fig. 2. Linear SVM Classification Result Figure

four algorithms performs similarly, the deviation is less than 1% ,which results in these algorithms converge to the same point. Whats more, we find that the parameters values in the algorithms are very important. In the beginning, we set the epsilon in AdaDelta as 1e-9, the loss is a straight line. Then we change the epsilon to 1e-5. The loss becomes converge.