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:
Wei Wen

Supervisor:
Mingkui Tan

Student ID:
201530613009

Grade:
Undergraduate

December 15, 2017

Logistic Regression, Linear Classification and Stochastic Gradient Descent

***Abstract*—In this paper,I do the Logistic Regression experiment and the Support Vector Machine experiment on a9a dataset using stochastic gradient descent(SGD) and mini-batch SGD. I use different optimized methods(NAG, RMSProp, AdaDelta and Adam) to update my model parameters. I compare the differences and relationship of Logistic Regression and Support Vector Machine. I compare the difference of stochastic gradient descent(SGD) and mini-batch SGD. And I compare the the difference of different optimized methods of SGD.**

I. Introduction

In recent years, machine learning has become more and more popular . Logistic Regression and the Support Vector Machine(SVM) are two basic and useful models of machine learning. They are widely used in classification area.

Stochastic gradient descent(SGD) and mini-batch SGD are two good optimization algorithms to get the parameters of model. And some variant of SGD perform better, such as NAG, RMSProp, AdaDelta and Adam.They over some shortcomings of traditional SGD.

However, as a student, I havent realized Logistic regression and SVM algorithms. I havent the know the difference stochastic gradient descent(SGD) and mini-batch SGD. And I havent know the difference of NAG, RMSProp, AdaDelta and Adam algorithm.

In order to compare and understand the difference between stochastic gradient descent(SGD) and mini-batch SGD,comparing and understanding the differences and relationships between logistic regression and SVM, and further understand the performance of NAG, RMSProp, AdaDelta and Adam algorithm, we conduct the following experiment.

II. Methods and Theory

A. Logistic Regression

Logistic Regression is a linear regression model.It estimate the probability of a sample belong to which class.We use a linear transformation and a logistic function to predict probability the of a sample x:

$$s = \mathbf{w}^\top \mathbf{x}$$

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

To get a proper parameter w for the model and get the formulation can be easily computed, we define the cross entropy measure for the loss function:

$$\mathbf{E}_{in}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_n\cdot\mathbf{w}^\top\mathbf{x}})$$

To avoid overfitting, we add a regularization factor to the loss function, and we get the final loss function:

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_i\cdot\mathbf{w}^\top\mathbf{x}_i})+\frac{\lambda}{2}||\mathbf{w}||_2^2$$

To apply gradient descent algorithm, we calculate the gradient of the loss function and propose a way to update parameters with rate :

$$\mathbf{w}' \to \mathbf{w} - \eta\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = (1-\eta\lambda)\mathbf{w} + \eta\frac{1}{n}\sum_{i=1}^{n}\frac{y_i\mathbf{x}_i}{1+e^{y_i\cdot\mathbf{w}^\top\mathbf{x}_i}}$$

B. Support Vector Machine

SVM is a linear classification model.It is a linear model so we can use the following formulation:

$$f(x) = \mathbf{w}^\top\mathbf{x} + b$$

The motivation of SVM is to find the most stable participation under the perturbations of inputs. The SVM model tries to maximize the margin. The learning of SVM can be formulated as an optimization problem:

$$\min_{\mathbf{w},b}\frac{\|\mathbf{w}\|^2}{2}$$
$$s.t. \quad y_i(\mathbf{w}^\top\mathbf{x}_i+b) \geqslant 1, \qquad i=1,2,\ldots,n.$$

However, the training data may not be linearly separable. To improve the accuracy of SVM model, we introduce a variable to represent how much the example is on wrong side of margin boundary. After importing the Hinge Loss, the optimization problem becomes:

$$\min_{\mathbf{w},b}\quad \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{N}\max(0, 1-y_i(\mathbf{w}^\top\mathbf{x}_i+b))$$

To apply gradient descent algorithm on SVM, we calculated the gradient of the optimization target:

$$\frac{\partial f(\mathbf{w},b)}{\mathbf{w}} = \begin{cases}\mathbf{w}^\top - C\mathbf{y}^\top\mathbf{X} & 1-y_i(\mathbf{w}^\top\mathbf{x}_i+b) >= 0\\ \mathbf{w}^\top & 1-y_i(\mathbf{w}^\top\mathbf{x}_i+b) < 0\end{cases}$$

$$\frac{\partial f(\mathbf{w},b)}{b} = \begin{cases}-C\sum_{i=1}^{N} y_i & 1-y_i(\mathbf{w}^\top\mathbf{x}_i+b) >= 0\\ 0 & 1-y_i(\mathbf{w}^\top\mathbf{x}_i+b) < 0\end{cases}$$

C. SGD and the varients of SGD

The SGD algorithm fetch a sample, and calculate the gradient on the sample, and update the parameters with following formulation:

$$\mathbf{g}_t \leftarrow \nabla J_i(\boldsymbol{\theta}_{t-1})$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta\mathbf{g}_t$$

The core idea of NAG(Nesterov accelerated gradient) is to use Momentum to predict the next step, instead of using the current theta. It has following formulation:

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

RMSProp tried to calculate the learning rate based on the gradient previously obtained. And we add an exponential decay to prevent the gradient of long age impacting the current gradient. It has following formulation:

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

AdaDelta, proposed an optimization procedure more automated, the programmer dont need to set up the initial learning rate any more:

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

Adam exploited the advantages of RMSProp on sparse data. The correction of the initialization bias also made Adam perform better.It enables us to do the initialization bias correction during the optimization phase:

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

III. Experiments

A. Dataset

We use the a9a dataset from LIBSVM Data, which include 32561 samples and each sample has 123 features in training set and 16281 samples and each sample has 123 features in testing set. The class of sample is marked as 1 and -1, corresponding to positive and negative class.

B. Implementation

We realized Logistic Regression and Linear Classification model and conduct the regression and classification experiment in a9a dataset. To optimize the parameter, we used SGD and mini-batch SGD. And we realized NAG, RMSProp, AdaDelta and Adam as the optimization algorithm and compare the performance of them.

In Logistic Regression experiment, we compare the SGD with mini-batch SGD, using the same training sample in the same order during the training phase. And I realized SGD and mini-batch SGD with NAG, RMSProp, AdaDelta and Adam.

For SGD,We set the iteration time to 3000, and the learn rate to 0.001.The experimental result is in figure 1.

For the mini-batch SGD,We set the iteration to 500 ,the batch size to 100,and learn rate to 0.001.The experimental result is in figure 2.

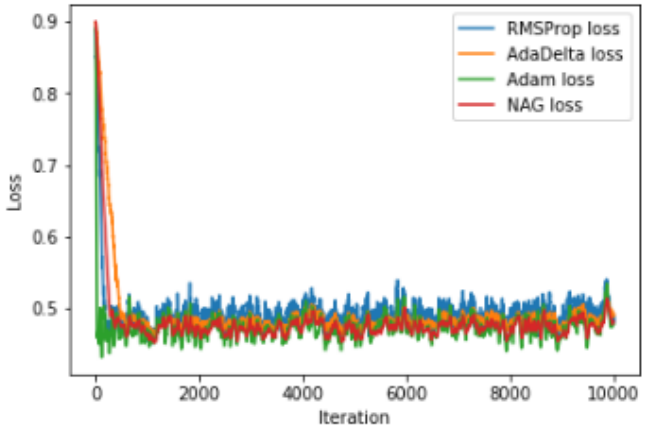

Fig. 1. loss curve for logistic regression experiment(SGD).

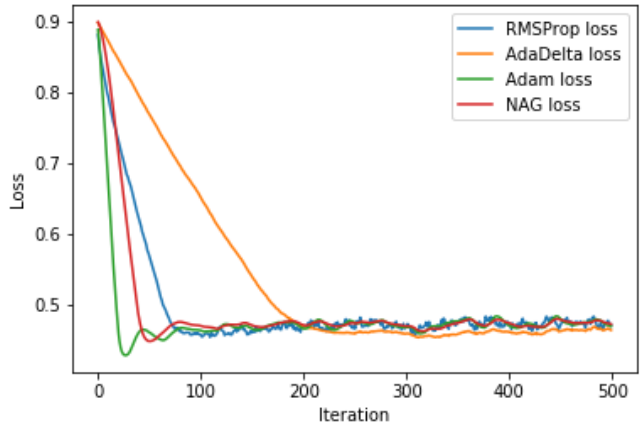

Fig. 2. loss curve for logistic regression experiment(mini-batch SGD)

In SVM experiment, we also compare the SGD with mini-batch SGD, using the same training sample in the same order during the training phase. And I realized SGD and mini-batch SGD with NAG, RMSProp, AdaDelta and Adam.

For SGD,We set the iteration time to 3000, and the learn rate to 0.001.The experimental result is in figure 3.

For the mini-batch SGD,We set the iteration to 500 ,the batch size to 100,and learn rate to 0.001.The experimental result is in figure 4.

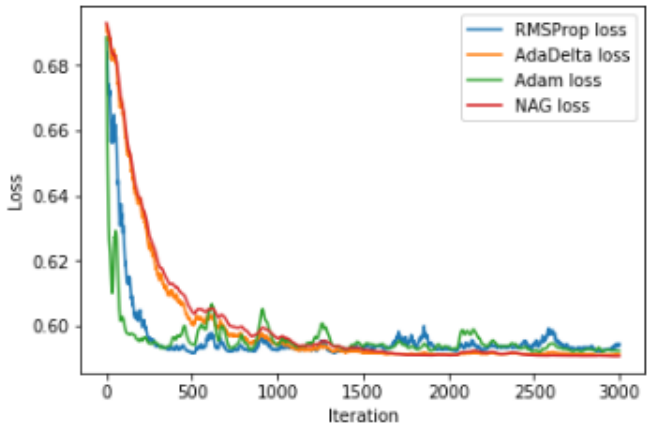

Fig. 3. loss curve for SVM experiment(SGD).

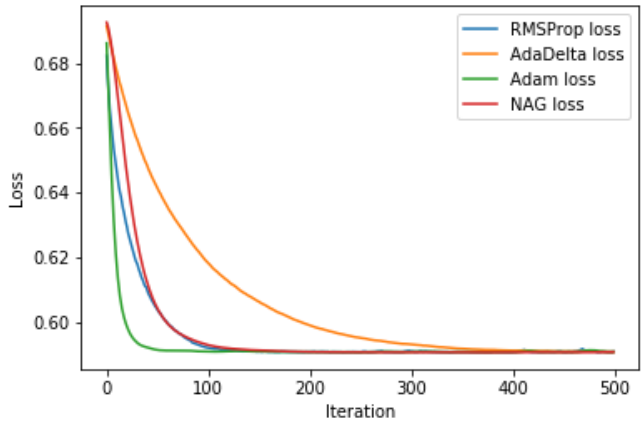

Fig. 4. loss curve for SVM experiment(mini-batch SGD).

IV. Conclusion

This section summarizes the paper. In our experiments, you can also write your gains and inspirations in here. Through this experiment,I found that logistic regression and SVM can both use in classification.But SVM classify intuitive.And logistic regression can predict the probability of a sample.

And I found that Adam descent so fast compare to NAG, RMSProp, AdaDelta.It is a good optimized method. AdaDelta descent slow,but it can get smooth loss curve.

And I found that mini-batch SGD perform well than SGD. It can get a smooth curve and descent fast.But the loss curve of SGD has fluctuation.

Through this experiment, I get a deeper comprehension of Logistic regression and SVM.And I also get a deep comprehension of different optimized methods, such as NAG, RMSProp, AdaDelta and Adam.And I get a intuitive understanding of SGD and mini-batch SGD. This experiment increased my interest in machine learning. I got a big harvest. And I will continue to learn more knowledge about machine learning.