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:

SongSheng Wang

Student ID：201530612880

Supervisor:

Mingkui Tan or Qingyao Wu

Grade:

Undergraduate

December14, 2017

Logistic Regression, Linear Classification and Gradient Descent

Abstract—In this paper, we realize the Logistic Regression and Support Vector Machine and do the regression and classification experiment on a9a dataset, using SGD and variant of SGD: NAG, RMSProp, AdaDelta and Adam to do the optimization of the parameter. We compare the SGD with these variant algorithms in the aspect of convergence rate, accuracy and degree of automation. We found all of these algorithms can get an approximation of optimal solution. However, the performance of them are different in detail. We also found the Logistic Regression is an efficient regression learning module, compared with Linear Regression in experiment 1.

I. Introduction

In recent years, machine learning has become a popular subject. Some of machine learning models, such as Support Vector Machine, Logistic Regression model, has been widely used in multiple areas. In order to get the optimal parameter, lots of optimization algorithms have been proposed to overcome the shortcoming of traditional gradient descent and stochastic gradient descent.

However, as a student, I don't have a clear view of the difference of effection with several optimization modules, and I haven't realized the Logistic Regression model.

In order to compare and understand the difference between gradient descent, stochastic gradient descent, getting a deeper understanding of logistic regression and linear classification, and understanding the principles of SVM and practice on larger data, we conduct the following experiment.

II. Methods and Theory

A. Logistic Regression

Logistic Regression is a linear regression model, it predict the output of a sample x by linear transformation using the formulation and a logistic function:

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

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

To get the proper parameter w for the model and get the formulation can be easily calculated, 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 on the loss function to be the final loss function in use:

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

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

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

B. Support Vector Machine

SVM is a linear classification model that being widely used in data mining and natural language processing. It is basically a linear classification model using 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". Learning the 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 compatibility of SVM model, we introduce a variable ξ 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 its varients

In SGD, the algorithm fetch a group of samples, calculate the gradient on these samples, and update the parameters with following formulation:

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

NaG algorithm predict the gradient in advance using the Momentum, according to the formulation:

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

RMSProp tried to calculate the learning rate ή

based on the gradient previously obtained. To avoid the learning rate becoming 0 too early, we add a exponential decay to the historical data:

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

AdaDelta, proposed an optimization procedure more automated, the programmer don't need to set up the initial learning rate any more:

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

Adam enables us to do the initialization bias correction during the optimization phase:

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

III. Experiment

A. Dataset

We use the a9a dataset from LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features. The class of sample is marked as 1 and -1, corresponding to positive and negative class.

B. Inplementation

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 its varients: NAG, RMSProp,AdaDelta and Adam as the optimization algorithm and compare the performance of them.
In Logistic Regression experiment, we compare the SGD with its varients, using the same training sample in the same order during the training phase. We set the iteration time to 4000, and the learning rate to 0.002. We choose 30 samples randomly and calculate its gradient during every time of iteration. We evaluate the quality of the parameter by calculating the loss in validation set. The experimental result is in figure 1.

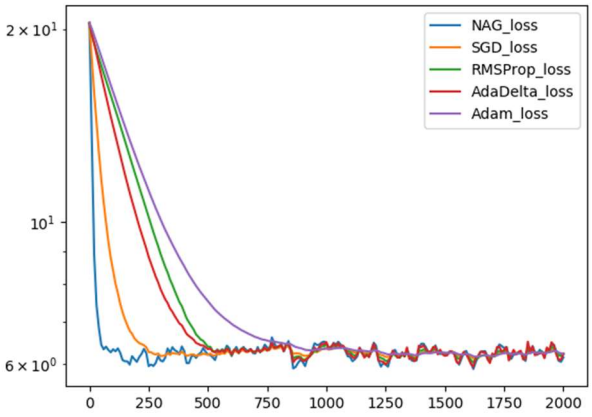

Figure 1 loss curve for logistic regression experiment

From the picture we can find that the Adam and RMSProp convergences relatively slow. However, they could learn the learning rate automatically. The NAG convergences much faster than SGD.
In SVM experiment, we did the similar operation as the Logistic regression experiment with SVM learning model, and different gradient computation. The experimental result is in figure 2.

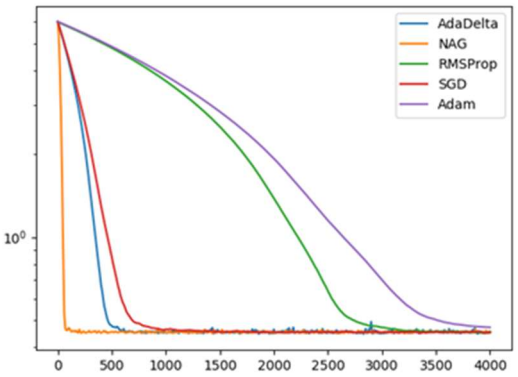

Figure 2 loss curve for SVM experiment

IV. CONCLUSION

In this experiment, We realized the Logistic regression and SVM model, get a deeper comprehension of Logistic regression and SVM.

We compared the difference between SGD and its variants, found that the NAG could accelerate the convergence procedure significantly. Although the RMSProp and Adam converges relatively slow, they can get the learning rate automatically, saving human effort.