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:
Muyi Li

Supervisor:
Mingkui Tan

Student ID:
201530612019

Grade:
Undergraduate

December 14, 2017

Linear Regression, Linear Classification and Stochastic Gradient Descent

Abstract—In order to compare the performance between AdaDelta, Adam, NAG, and RMSProp, we established the logistic regression model and the linear classification model. Then we used those four optimization methods to update the parameters of the two methods. And the result showed that AdaDelta declines fast at first but fluctuate later. Adam and RMSProp are more stable. While, NAG's speed differs in different models.

I. Introduction

Though gradient descent algorithm is effective and widely used, it is difficult to choose a reasonable learning rate, and may meet with the saddle points. Then, there are various optimization methods appear, and each has its advantages and deficiencies. People usually use these methods without thinking about which one is the most appropriate. Therefore, this experiment aims to make a clear distinction between AdaDelta, Adam, NAG, and RMSProp when used to update parameters in logistic regression model and linear SVM.

II. Methods and Theory

A. *Logistic regression*

Logistic regression is a kind of generalized linear model, and it is used to predict the influencing factors or probabilities of occurrence, from datasets. We get weight of independent variables to predict through logistic regression analysis. And the dependent variables can be values like true or false. While, independent variables can contain a lot, and it can not only be continuous, but also be discrete.

In this experiment, we assume that the labels are binary:

$$y_i \in \{-1,+1\}$$

According to sigmoid function the expression for prediction function is:

$$h^{\omega}(x)=g(\omega^{\mathrm{T}}x)=\frac{1}{1+e^{-\omega^{\mathrm{T}}x}}$$

And we use maximum likelihood method to estimate w and b, so the log-likelihood loss function is:

$$L_D(\omega)=\frac{1}{m}\sum_{i=1}^{m}[-y_i\omega^{\mathrm{T}}x_i+\ln(1+e^{\omega^{\mathrm{T}}x_i})]$$

$$L_D(\omega)=-\frac{1}{m}\sum_{i=1}^{m}\ln(g(y_i)\cdot\omega^{\mathrm{T}}x_i)$$

And the gradient of loss function respecting to ω is:

$$\frac{\partial L_D(\omega)}{\partial\omega}=-\frac{1}{m}\sum_{i=1}^{m}\frac{1}{g(z)}\cdot\frac{\partial g(z)}{\partial\omega}$$

$$z=y_i\cdot\omega^{\mathrm{T}}x_i$$

from g(z) we get the gradient:

$$grad=\frac{\partial L_D(\omega)}{\partial\omega}=-\frac{1}{m}\sum_{i=1}^{m}(1-g(y_i\cdot\omega^{\mathrm{T}}x_i))\ (y_i\cdot x_i)$$

B. *Linear SVM*

Linear support vector machine(SVM) is a fast machine learning algorithm for solving multiclass classification problems from data sets. People use SVM to solve the optimal classification surface. And the minimum distance from the two point sets to this plane is the largest. In other words, the distance from the edge of the two point sets to this plane is the largest.

In this experiment, we assume that the labels are binary: And for all the samples, the loss function is:

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

Besides, the gradient of loss function to ω is:

$$\frac{\partial L_D(\omega,b)}{\partial\omega}=\begin{cases}\omega^{\mathrm{T}}-CX^{\mathrm{T}}y, & if\ \ 1-Y_i(\omega^{\mathrm{T}}x_i+b)\geq 0\\ \omega^{\mathrm{T}}, & if\ \ 1-Y_i(\omega^{\mathrm{T}}x_i+b)<0\end{cases}$$

C. *Optimization methods*

SGD uses one or more data at random and make a gradient descent, however, there are tons of problems to be solve. For example, learning rate extremely impacts the speed of convergence. And it is hard to regulate. Besides, it is easy to get into a less good minimum or saddle point. Therefore, kinds of premiums:

NAG: The core idea is to use Momentum to predict the next step, instead of using the current ω:

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

RMSProp: This method solves the problem that the speed of learning tends to 0 in AdaGrad:

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

AdaDelta: This method can also solve the problem of AdaGrad, but this one do not need to set the learning rate:

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

Adam: Adam leverages the advantages of AdaGrad and RMSProp on sparse data. And the correction of the initialized bias also makes Adam perform better:

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

III. EXPERIMENT

A. Dataset

Data sets	From	a9a of LIBSVM	
	Samples	a9a	32561
		a9a(testing)	16281
	Features (/ sample)	123 with label as -1 or 1	

B. Steps

a) Regression Experiment

1. Build a model:

(1) Load the training set and validation set.

(2) Initialize logistic regression model parameters.

2. Select the loss function.

3. Optimization (SGD):

(1) Calculate gradient toward loss function from one sample.

(2) Update model parameters using different optimized methods.

(3) Select an appropriate threshold, and mark the sample whose predict scores > the threshold as positive, and the others as negative.

(4) Repeat step 3 for several times.

4. Draw graph of loss functions' values with the number of iterations.

b) Classification Experiment

1. Build a model

(1) Load the training set and validation set

(2) Initialize logistic regression model parameters

2. Select the loss function

3. Optimization (SGD)

(1) Calculate gradient toward loss function from one sample

(2) Update model parameters using different optimized methods

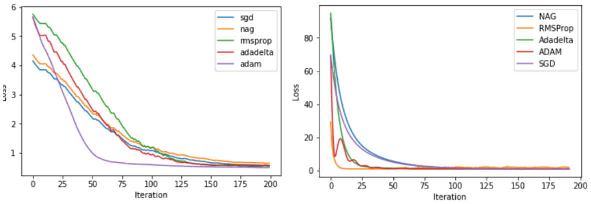
(3) Select an appropriate threshold, and mark the sample whose predict scores > the threshold as positive, and the others as negative.

(4) Repeat step 3 for several times

4. Draw graph of loss functions' values with the number of iterations.

In this experiment, we use a9a.txt for training, and a9atest.t for validation. And we initialize parameters as zero.

C. Results



IV. CONCLUSION

In this experiment, we built logistic regression model and linear SVM model, and optimize them with four methods: Adam, AdaDelta, RMSProp, and NAG. Logistic regression is a kind of generalized linear model used to predict the influencing factors or probabilities of occurrence. And linear SVM is a little difficult to realize and speed up. As for the performance of the four optimization methods. Parts of codes, which are used to realize these methods, are based on some other books and blogs because their parts are better organized than mine. As a result, Adam and RMSProp are stable and fast, Adadelta is unstable, while NAG is in depends. I'm sorry about that here is something wrong, after I have changed my codes. There is not enough time to run again.