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:
Zijie Hong

Supervisor:
Qingyao Wu

Student ID：
201530611593

Grade:
Undergraduate

December 14, 2017

Linear Regression, Linear Classification and Gradient Descent

Abstract— this report is in order to compare the difference Linear Regression and Linear Classification for the problem of classification . Moreover , the difference among several gradient descent methods.

I. INTRODUCTION

As we all know , both Logistic regression and SVM ,which is without a kernel, can work well in the supervised learning problem of linear classification. but they have different implementation method . An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. Logistic regression is a regression model where the dependent variable is categorical. The output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick.

What's more , they can be both optimized by the method of gradient descent. So we want to have some experiment to show difference among several gradient descent methods.

II. METHODS AND THEORY

Using the gradient decent methods , including NAG , adagrad , RMSProp , Adam in order to optimize the loss function of SVM and Logistic regression

The dataset is downloaded from the libsvm. And then I split it into the training set and validation set , and the trainging take 2/3 of the original dataset and validation set take 1/3.

III. EXPERIMENT

A.Dataset:

a9a from LIBSVM Data

B.methods:

Using the NAG , adagrad , RMSProp , Adam to optimize the SVM and Logistic regression.

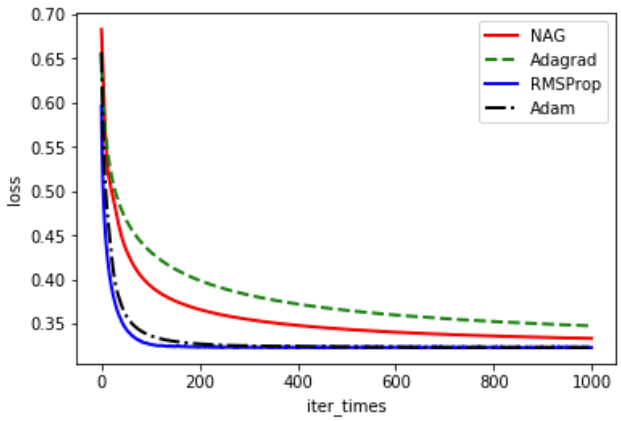
C.Implementain

Logistic Regression and Stochastic Gradient Descent

1. Load the training set and validation set.
2. Initalize logistic regression model parameters, you can consider initalizing zeros, random numbers or normal distribution.
3. Select the loss function and calculate its derivation, find more detail in PPT.
4. Calculate gradient toward loss function from **partial samples**.
5. **Update model parameters using different optimized methods(NAG，RMSProp，AdaDelta and Adam).**
6. 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 .
7. Repeate step 4 to 6 for several times, and **drawing graph of losses with the number of iterations**.

Simulation parameters

The number of iterations	1000
Learning rate	0.01
The length of a batch	5000
Gamma(NAG, RMS)	0.9
Gamma(Adam)	0.999
Beta(Adam)	0.9

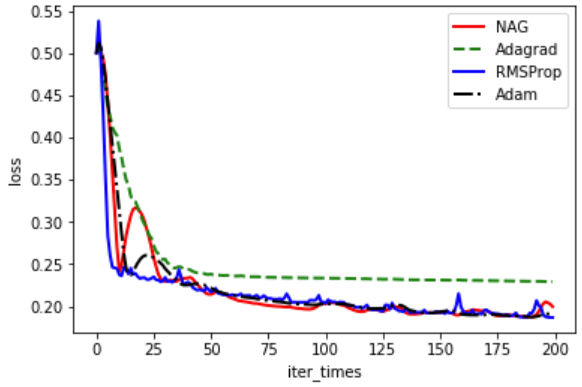


Linear Classification and Stochastic Gradient Descent

1. Load the training set and validation set.
2. Initalize SVM model parameters, you can consider initalizing zeros, random numbers or normal distribution.
3. Select the loss function and calculate its derivation, find more detail in PPT.
4. Calculate gradient toward loss function from **partial samples**.
5. **Update model parameters using different optimized methods(NAG，RMSProp，AdaDelta and Adam).**
6. 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.
7. Repeate step 4 to 6 for several times, and **drawing graph of losses with the number of iterations**.

Simulation parameters

The number of iterations	200
Learning rate	0.01
The length of a batch	50
Gamma(NAG, RMS)	0.9
Gamma(Adam)	0.999
Beta(Adam)	0.9



IV. CONCLUSION

From the experiment, I make the following conculsion:
1. When using the SGD , select the bigger batch number and smaller learining rate can be more smoothly descent.
2.Among four methods , adam has better Synthetically performance than the other , for it can update the learing rate by itself and have adaptive estimates of lower-order moments , and also it can update this parameters by itself. Moreover , the RMSProp shows fater descent trend than the other , which I think
3.If the method cannot update the learning rate itself , the function to calculate the gradient must divde by the batch number **not the training set numbe**r , which I make a big mistake here and makes my NAG curve an approximately straight line.