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:
Hu Lin

Supervisor:
Mingkui Tan or Qingyao Wu

Student ID：
201721045497

Grade:
Graduate

December 14, 2017

Linear Regression, Linear Classification and Gradient Descent

***Abstract*—This is the experiment report two of the Machine Learning. The main motivation of experiment is to get further understanding of the principles of SVM and practice on larger data. Then compare and understand the difference between gradient descent and stochastic gradient descent , as well as differences and relationships between Logistic regression and linear classification.**

I. Introduction

The main motivation of experiment is to get further understanding of the principles of SVM and practice on larger data. Then compare and understand the difference between gradient descent and stochastic gradient descent , as well as differences and relationships between Logistic regression and linear classification.

II. Methods and Theory

In statistics, logistic regression, or logit regression, or logit model is a regression model where the dependent variable 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 analysed 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.

Logistic regression was developed by statistician David Cox in 1958. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor.

Logistic regression just solves this problem using logistic loss and linear hypothesis function:

$$\text{minimize}_\theta \sum_{i=1}^{m} \log(1+\exp(-y^{(i)} \cdot \theta^T x^{(i)}))$$

Gradient descent updates:

$$\theta := \theta - \alpha \sum_{i=1}^{m} -y^{(i)} x^{(i)} \frac{1}{1+\exp(y^{(i)} \cdot \theta^T x^{(i)})}$$

In the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics. An object's characteristics are also known as feature values and are typically presented to the machine in a vector called a feature vector. Such classifiers work well for practical problems such as document classification, and more generally for problems with many variables (features), reaching accuracy levels comparable to non-linear classifiers while taking less time to train and use.

A (linear) support vector machine (SVM) just solves the canonical machine learning optimization problem using hinge loss and linear hypothesis, plus an additional regularization term,

$$\text{minimize}_\theta \sum_{i=1}^{m} \max\{1 - y^{(i)} \cdot \theta^T x^{(i)}, 0\} + \frac{\lambda}{2}\|\theta\|_2^2$$

Updates using gradient descent:

$$\theta := \theta - \alpha \sum_{i=1}^{m} -y^{(i)} x^{(i)} 1\{ y^{(i)} \cdot \theta^T x^{(i)} \leq 1\} - \alpha\lambda\theta$$

In this experiment we compare four optimize function:

1. NAG

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

2. RMSprop

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

3. AdaDelta

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

4. Adam

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

III. Experiment

A. Dataset

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set.

Linear classification uses australian in LIBSVM Data, including 690 samples and each sample has 14 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set.

B. *Environment for Experiment*

python3, at least including following python package: sklearn, numpy, jupyter, matplotlib. It is recommended to install anaconda3 directly, which has built-in python package above.

C. *Experiment Step*

The experimental code and drawing are completed on jupyter.

Linear Regression and 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 loss function and calculate its derivation. Find more detail in PPT.
4. Calculate gradient G 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 L_{NAG} , $L_{RMSProp}$, $L_{AdaDelta}$ and L_{Adam} with the number of iterations.

Linear Classification and 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 loss function and calculate its derivation. Find more detail in PPT.
4. Calculate gradient G 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 L_{NAG} , $L_{RMSProp}$, $L_{AdaDelta}$ and L_{Adam} with the number of iterations.

D. *Experiment Result*

We only use part of train set to train the model, but use all of validation set for test. Each iteration we randomly choose a batch of samples. The batch size is 3000.

Logistic Regression and Stochastic Gradient Descent

1. NAG

The parameter theta is initial as zero, and the learning rate is 0.01, epoch is 50, gamma is set as 0.9 as well as 0.1 for lambda. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 1.

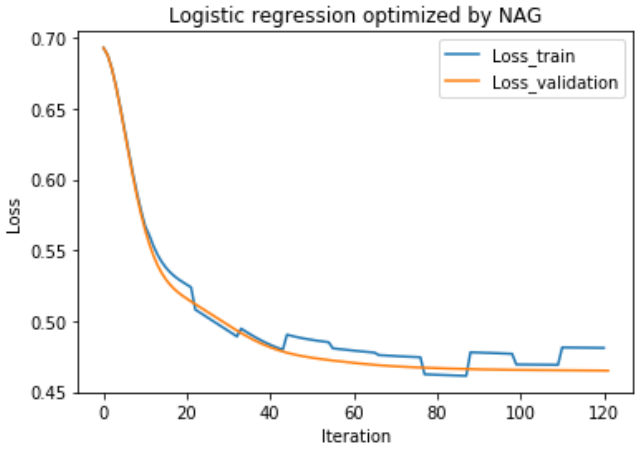

Fig. 1. Curve of logistic Regression loss optimized by NAG

2. RMSProp

The parameter theta is initial as zero, and the learning rate is 0.01, epoch is 500, gamma is set as 0.999 as well as 0.1 for lambda. In additional, there is a constant epsilon of which value is e-8. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 2.

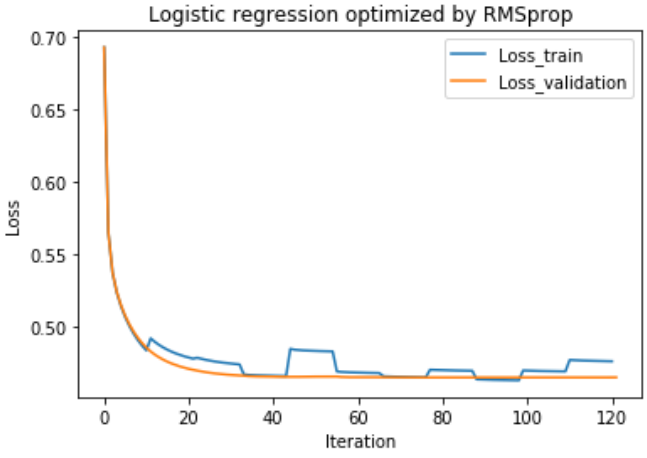

Fig. 2. Curve of logistic Regression loss optimized by RMSProp

3. AdaDelta

The parameter theta is initial as zero, and the learning rate is 0.01, epoch is 50, gamma is set as 0.999 as well as 0.01 for delta_t. In additional, there is a constant epsilon of which value is e-8. Notice in the AdaDelta there is no need for learning rate. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as

well as L_{train} is show as Fig. 3.

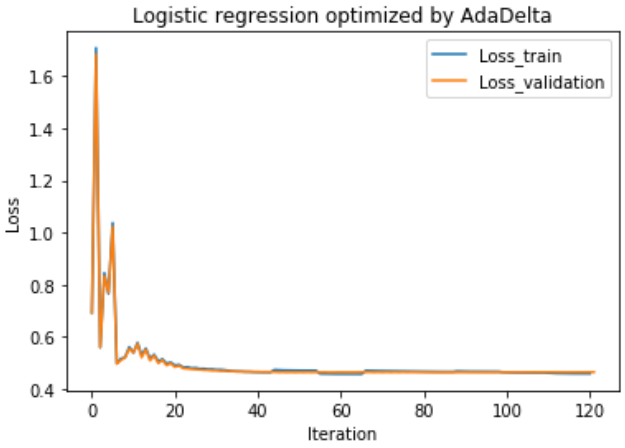

Fig. 3. Curve of logistic Regression loss optimized by AdaDelta

4. Adam

The parameter theta is initial as zero, and the learning rate is 0.2, epoch is 50, beta is initial as 0.9, gamma is set as 0.99 as well as 0.1 for lambda. In additional, there is a constant epsilon of which value is e-8. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 4.

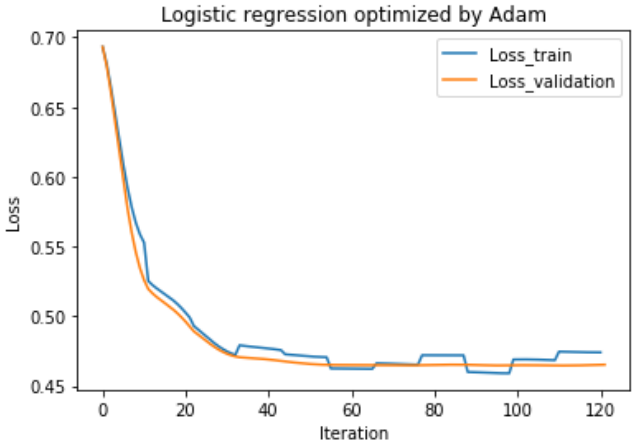

Fig. 4. Curve of logistic Regression loss optimized by Adam

Liner Classification and Stochastic Gradient Descent

1. NAG

The parameter theta is initial as zero, and the learning rate is 0.01, epoch is 50, gamma is set as 0.9 as well as 0.1 for lambda. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 5.

2. RMSProp

The parameter theta is initial as zero, and the learning rate is 0.01, epoch is 500, gamma is set as 0.999 as well as 0.1 for lambda. In additional, there is a constant epsilon of which value is e-8. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 6.

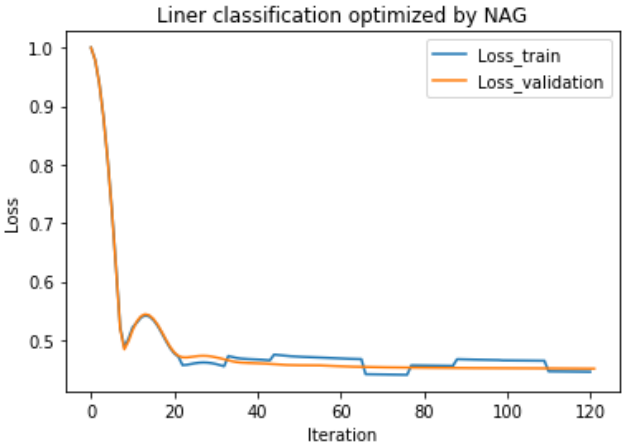

Fig. 5. Curve of liner classification loss optimized by NAG

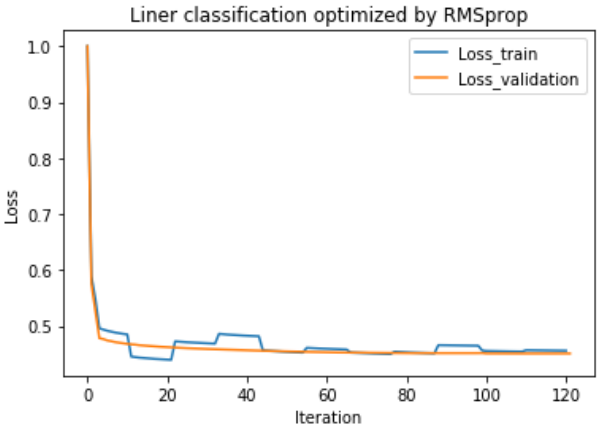

Fig. 6. Curve of liner classification loss optimized by RMSProp

3. AdaDelta

The parameter theta is initial as zero, and the learning rate is 0.01, epoch is 50, gamma is set as 0.999 as well as 0.01 for delta_t. In additional, there is a constant epsilon of which value is e-8. Notice in the AdaDelta there is no need for learning rate. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 7.

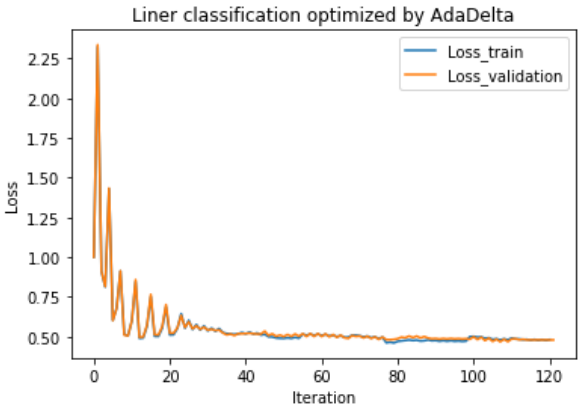

Fig. 7. Curve of liner classification loss optimized by AdaDelta

4. Adam

The parameter theta is initial as zero, and the learning rate is 0.2, epoch is 50, beta is initial as 0.9, gamma is set as 0.99 as well as 0.1 for lambda. In additional, there is a constant epsilon of which value is e-8. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 8.

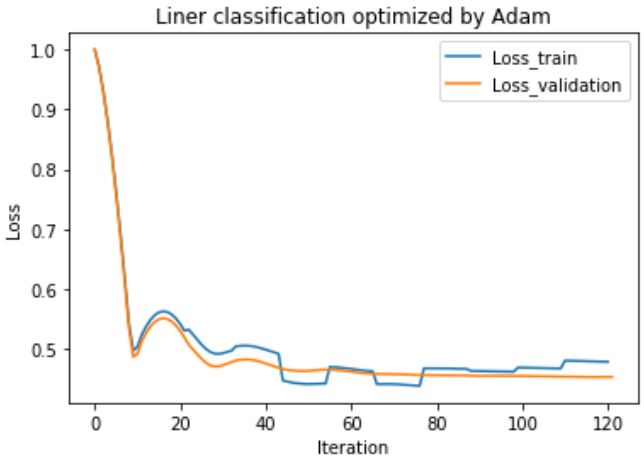

Fig. 8. Curve of liner classification loss optimized by Adam

Overall, the logistic regression and liner classification are essentially the same, that is, the fitting (matching) of the model. However, the **y** value (also known as label) of the classification problem is more discretized, and the same **y** value may correspond to a large number of **x**, which is of a certain range.

Therefore, the classification problem is some x in a certain region corresponds to a singe **y**, and the model of regression problem is more inclined to map x in a very small region or x in general to **y**.

IV. CONCLUSION

This experiment let me further understand the principles of SVM and practice on larger data. I compare and understand the difference between gradient descent and stochastic gradient descent, as well as the differences and relationships between Logistic regression and linear classification. It also raised my level of coding for machine learning.