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:
Dewei Su

Supervisor:
Mingkui Tan

Student ID:
201530612736

Grade:
Undergraduate

December 14, 2017

Stochastic Gradient Descent for Solving Classification Problems

***Abstract*—This experiment has three main motivations. First compare and understand the difference between gradient descent and stochastic gradient descent. Second compare and understand the differences and relationships between Logistic regression and linear classification. Third further understand the principles of SVM and practice on larger data.**

I. Introduction

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.Two of the most used classifier for the linear classification problem is logistic regression and support vector machine.

Logistic regression is a regression model where the dependent variable (DV) is categorical in statistics.It covers the case of a binary dependent variablethat 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. 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.

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis in machine learning. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). 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. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. In other words, SGD tries to find minima or maxima by iteration.

In this experiment, we use the logistic regression and SVM classifier to do the linear classificaiton problem. A number of different gradient descent optimization algorithms are used to optimize the gradient such as SGD,NAG,RMSPropAdaDelta and Adam.The details of these different algorithm are in the Methods section.Finally we get the expected result and hope to do some future work about linear classification problem and the optimization algorithm.

II. Methods and Theory

In this section, we will introduce the theory of logistic regression and support vector machine with their loss function and the calculation of gradient.Then the different gradient optimization algorithms will be present.

A. Logistic Regression

In traditinal linear classifier problem we use $s = w_T x$ to do the classification.But sometimes we want to output the probability or the likelihood of the problem.So we define the logistic function as below to address the probability

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

The function is a continuous function.If $z \to +\infty$, then $g(z) \to 1$; If $z \to -\infty$, then $g(z) \to 0$.So the likelihood of the predicted result will be

$$h_x(x) = g(w^T x) \tag{2}$$

$$P(y|x) = \begin{cases} h_w(x) = g(w_T x) & \text{y=0} \\ 1 - h_w(x) = g(-w_T x) & \text{y=-1} \end{cases}$$

Still, we can uniform the two expression into one as $P(y|x) = g(y * w^T x)$.Notice that the data set $(x_1, y_1), ..., (x_n, y_n)$ are independently generated which means the probability of getting the $y_1, ..., y_n$ in D from the corresponding $x_1, ..., xn$ will be $P(y_1, ...y_n|x1, ..., x_n) =$

$\prod_{i=1}^{n} P(y_i|x_i)$. According to the Maximum Likelihood Estimate Theory in statistics we maximize the probability

$$max\prod_{i=1}^{n} P(y_i|x_i) = min\frac{1}{n}\sum_{i=1}^{n} log(1+e^{-y_i*w^T*x_i}) \tag{3}$$

The final loss function and the gradient of it(for all the samples) will be

$$J(w) = \frac{1}{n}[\sum_{i=1}^{n} y_i log h_w(x_i) + (1-y_i)log(1-h_w(x_i))] \tag{4}$$

$$\frac{\partial J(w)}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}(h_w(x_i)-y_i)*x_i \tag{5}$$

B. Support Vector Machine

In svm we hope to select two parallel hyperplanes that separate the two classes of data and let the distance between them as large as possible.The region bounded by these two hyperplanes is called the margin.The margin can be calculated as

$$\frac{x}{||w||}*(x_+ - x_-) = \frac{2}{||w||} \tag{6}$$

Learning the SVM can be formulated as an optimization:

$$max_{w,b}\frac{2}{||w||} \tag{7}$$

Or equivalently:

$$min_{w,b}\frac{||w||^2}{2} \tag{8}$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 \quad i = 1,2,...n \tag{9}$$

In general there is a trade off between the margin and the number of mistakes on the training data.Moreover, training data may not be linearly separable.So we introduce variable $\xi_i \geq 0$, for each i, which represents how much example i is on wrong side of margin boundary.If $\xi_i = 0$ then it is ok.If $0 < \xi < 1$ it is correctly classified,but with a smaller margin than $\frac{1}{||w||}$.If $\xi > 1$ then it is incorrectly classified.The optimization problem becomes

$$min_{w,b}\frac{||w||^2}{2} + C\sum_{i=1}^{n}\xi_i \tag{10}$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1,2,...n \tag{11}$$

When we use the Hinge Loss as the ξ,we address the final loss function and the gradient of it:

$$J(w) = min_{w,b}\frac{||w||^2}{2} + C\sum_{i=1}^{n} max(0, 1-y_i(w^T x_i + b)) \tag{12}$$

$$\frac{\partial J(w)}{\partial w} = w + g_w(x_i) \tag{13}$$

$$g_w(x_i) = \sum_{i=1}^{n}\begin{cases} -y_i x_i & 1-y_i(w^T x_i + b) \geq 0 \\ 0 & 1-y_i(w^T x_i + b) < 0 \end{cases}$$

C. Stochastic Gradient Descent (SGD)

In theory we use gradient descent to optimize the weight after each iteration.However using all samples to calculate the gradient means redundant information might be used and leads to the slow convergence.So we use random part of the samples to calculate the gradient which is called the mini-batch stochastic gradient descent

$$g_t = \bigtriangledown J_i(w_{t-1}) \tag{14}$$

$$w_t = w_{t-1} - \eta g_t$$

η represents the learning rate which user can set. g_t means the gradient of the loss function which only use some samples to calculate.

D. Momentum

Imagine a ball rolling down from the mountain, started at the speed of 0.The ball will roll stumble(concussion).After a period of accumulation of momentum, the concussion will decrease and the ball will go straight down the hill.This phenomenon could be expressed as

$$g_t = \bigtriangledown J_i(w_{t-1}) \tag{15}$$

$$v_t = \gamma v_{t-1} + \eta g_t$$

$$w_t = w_{t-1} - v_t$$

Compare to the previous formula the current gradient is the weighted average of the prevous gradient. γ is a hyper parameter which could decrease the concussion and make the function converges more quickly.

E. Nesterov Accelerated Gradient(NAG)

The key idea of the NAG is to use the momentum to predict the next gradient rather than the current w.

$$g_t = \bigtriangledown J_i(w_{t-1} - \gamma v_{t-1}) \tag{16}$$

$$v_t = \gamma v_{t-1} + \eta g_t$$

$$w_t = w_{t-1} - v_t$$

When we calculate the gradient we did not use w_{t-1} directly but add the γv_{t-1} which means we do a prediction using the momentum.

F. RMSProp

We can notice that among all the above algorithm each elements in the weight matrix were updated by the same learning rate. For updating the elements with different learning rate the AdaGrad and RMSProp algorithm were proposed to solve this problem.We only introduce the RMSProp algorithm here which is the improved version of the AdaGrad.

$$\begin{aligned} g_t &= \bigtriangledown J_i(w_{t-1}) \qquad (17)\\ G_t &= \gamma G_{t-1} + (1-\gamma) g_t \odot g_t \\ w_t &= w_{t-1} - \frac{\eta}{\sqrt[2]{G_t + \epsilon}} \odot g_t \end{aligned}$$

This algorithm use the previous gradient information and make sure different elements in the weight matrix have differnt learning rate.Moreover there is no problem with the learning rate tending to 0.

G. AdaDelta

This algorithm is another Variant of the AdaGrad which do not need the initial learning rate. Intuitively it use the previous step size to predict the next step size.The mathematical fomula is

$$\begin{aligned} g_t &= \bigtriangledown J_i(w_{t-1}) \qquad (18)\\ G_t &= \gamma G_{t-1} + (1-\gamma) g_t \odot g_t \\ \triangle w &= -\frac{\sqrt[2]{\triangle_{t-1} + \epsilon}}{\sqrt[2]{G_t + \epsilon}} \\ w_t &= w_{t-1} + \triangle w_t \\ \triangle_t &= \gamma \triangle_{t-1} + (1-\gamma) \triangle w_t \odot \triangle w_t \end{aligned}$$

The $\triangle_{t-1} + \epsilon$ is used to estimate the learning rate.

H. Adam

Adam makes use of the advantages of AdaGrad and RMSProp on sparse data. The correction of the initialized deviation also makes the Adam better.Adam also adaptively adjusts 1 order moments (mean) and 2 order moments (variance).

$$\begin{aligned} g_t &= \bigtriangledown J_i(w_{t-1}) \qquad (19)\\ m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ G_t &= \gamma G_t + (1-\gamma) g_t \odot g_t \\ \alpha &= \eta \frac{\sqrt[2]{1-\gamma^t}}{1-\beta^t} \\ w_t &= w_{t-1} - \alpha \frac{m_t}{\sqrt[2]{G_t + \epsilon}} \end{aligned}$$

III. Experiments

A. Dataset

Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features.

B. Implementation

The experiment step is as follow

Logistic Regression and Stochastic Gradient Descent

1.Load the training set and validation set.

2.Initalize logistic regression model parameters, we consider initalizing zeros, random numbers or normal distribution.

3.Select the loss function and calculate its derivation.

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 L_{NAG},$L_{RMSProp}$,$L_{AdaDelta}$ and L_{Adam}.

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 Stochastic Gradient Descent

1.Load the training set and validation set.

2.Initalize SVM model parameters, we consider initalizing zeros, random numbers or normal distribution.

3.Select the loss function and calculate its derivation.

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 L_{NAG},$L_{RMSProp}$,$L_{AdaDelta}$ and L_{Adam}.

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.

As for initialization method we choose to set all parameter into ones. And there is no validation set in this experiment.We use trainging set a9a and testing set a9a.t to finish this lab.

Table I and Table II show the parameter we choose for the logistic regression and SVM in this lab.

TABLE I
Parameters of Logistic Regression

Learning rate	$\eta = 0.01$
Iteration time	$epoch = 500$
Parameter γ for all the optimization function	$\gamma = 0.9$
Parameter ϵ for the RMSProp algorithm	$\epsilon = 1e-8$
Parameter ϵ for the AdaDelta algorithm	$\epsilon = 1e-3$
Parameter β for the Adam algorithm	$\beta = 0.9$

Fig. 1 and Fig. 2 shows the loss value of logistic regression and svm using different optimization function.

TABLE II
PARAMETERS OF SVM

Learning rate	$\eta = 0.01$
Iteration time	$epoch = 250$
Parameter γ for all the optimization function	$\gamma = 0.9$
Parameter β for the Adam algorithm	$\beta = 0.9$
Parameter ϵ for the RMSProp algorithm	$\epsilon = 1e - 8$
Parameter ϵ for the AdaDelta algorithm	$\epsilon = 1e - 1$
Parameter C for the relaxation variable	$C = 0.8$

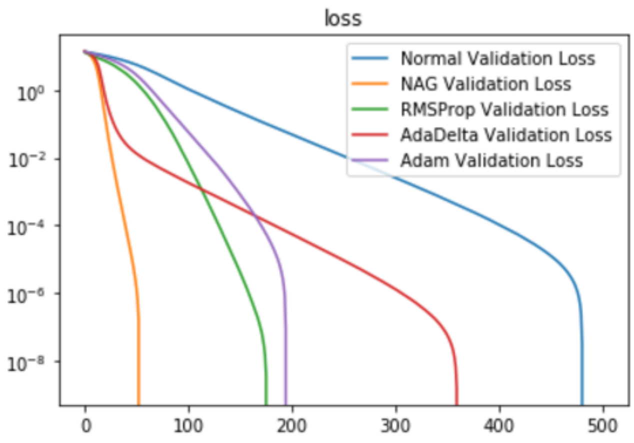

Fig. 1. The loss of logistic regression using different optimization function.

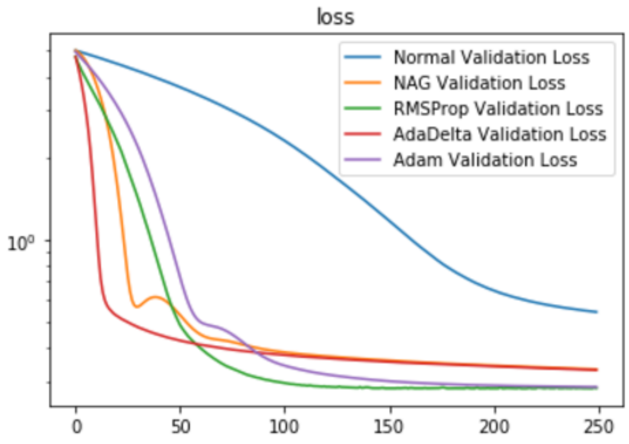

Fig. 2. The loss of svm using different optimization function.

IV. CONCLUSION

Logistic Regression and Support Vector Machine are two of the most important methods in machine learning area.As the basic methods of binary linear classification,both of them make a huge contribution to the development of machine learning algorithm and guid the direction of experiment for a long time.It is believed that many improved methods could be used to both of the algorithm and address a better results.Also, different parameters of different optimization function can lead to various results.So the future work may aim at adjusting the different parameters to get the better results and explore the optimization functions.