The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Yi Lan

Supervisor:
Mingkui Tan

Student ID：
201630664673

Grade:
Undergraduate

October 24, 2018

Logistic Regression and Support Vector Machine

Contrasting with linear regression, construct a logistic regression model to implement linear classification. And then apply SVM (support vector machine) mechanism to improve the classifier. All code is written with python and tested in python3 platform.$\boldsymbol{h_w(x) = g(w^\top x)}$

I. Introduction

Different from linear regression, the target values of linear classification are discrete, and a linear classification model's responsibility is to put the sample in correct side.

Linear classification is used to find a discriminant which can separate the samples which are linearly separable (Figure 1.a and 1.b). Otherwise linear classification model may not get satisfy result. To deal with the case that samples are not linearly separable (Figure 1.c and 1.d), we may need boosting algorithm, neural network, etc.

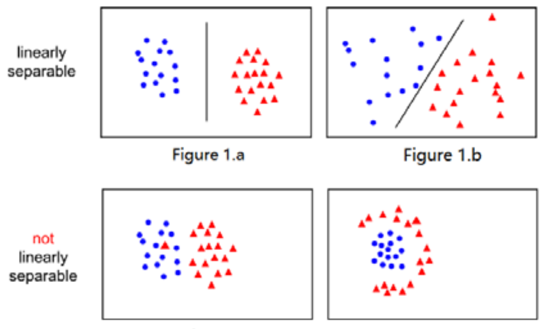


Figure 1.a Figure 1.b

Figure 1.c Figure 1.d

Generally, the discriminant is a line in *2D*, plane in *3D* and hyperplane in *mD*.

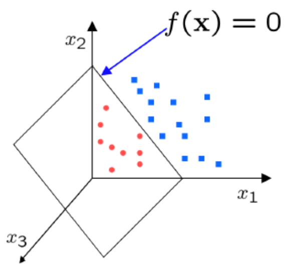


Figure 2. discriminant 3D

However, the perturbations of inputted samples, may be a wrong-classification sample or an outlier, etc. will result in an extreme classification model that has extremely bad robust. It means that our model cannot classify new samples that never seen before, in other words, the generalization is no good. (Figure 3.a)

Therefore, we need Max-margin Method to help us to measure the generalization of our model. And then we will try to use SVM to make a tradeoff between the precision of classifying known samples and the accuracy of classifying samples that never seen before.

In this experiment, we use the logistic regression and SVM classifier to solve the linear classification problem. What's more, GD and MSGD, two gradient descent optimization algorithms are used to optimize the model parameters.

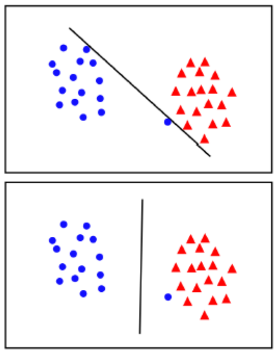


Figure 3.a

Figure 3.b

More experiment details are in Part II.

II. Methods and Theory

1. Logistic Regression

When we need to calculate the probability or the likelihood of the classification problem, we will use the logistic function as below to get the probability

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

Obviously, if $z \to +\infty$, then $g(z) \to 1$; if $z \to -\infty$, then $g(z) \to 0$. So the likelihood result would be

$$h_w(x) = g(w^\top x)$$

$$P(y|x) = \begin{cases} h_w(x) = g(w^\top x) & y = +1 \\ 1 - h_w(x) = 1 - g(w^\top x) & y = 0 \end{cases}$$

According to the Maximum Likelihood Estimate Theory, we maximize the probability

$$\max \prod_{i=1}^{n} P(y_i|x_i) = \min \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \mathrm{e}^{-y_i * w^\top * x_i}\right)$$

The complete loss function should be

$$J(w) = \frac{1}{n}\left[\sum_{i=1}^{n} y_i \log \mathrm{h}_w(x_i) + (1 - y_i)\log\left(1 - \mathrm{h}_w(x_i)\right)\right]$$

The gradient of loss function should be

$$\frac{\partial J(w)}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{h}_w(x_i) - y_i) * x_i$$

2. Support Vector Machine

The purpose of SVM is to find two parallel hyperplanes which can separate the two classes of samples, and at the same time, maximize the distance of the two hyperplanes. The region bounded by them is called margin, which can be calculate as

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

According to Max-margin Method, the optimization can be formulated as

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

Equivalently:

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

$$s.t. \quad y_i(w^\top x_i + b) \geq 1 \qquad \mathrm{i} = 1,2,\cdots n$$

Training data may not be simply linearly separable, although it is in global view. 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 this sample is at correct side; if $0 < \xi_i < 1$, then it is correctly classified, but with a smaller margin than $\frac{1}{\|w\|}$; if $\xi_i > 1$, then it is incorrectly classified. (Figure 4)

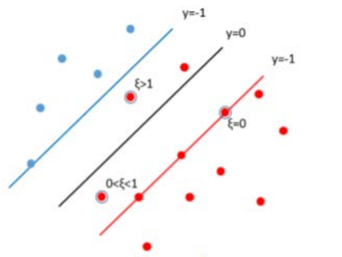


Figure 4

Now, the optimization problem has trans to:

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

$$s.t. \quad y_i(w^\top x_i + b) \geq 1 - \xi_i \qquad \mathrm{i} = 1,2,\cdots n$$

Now, we use Hinge Loss function as:

$$Hinge\ Loss = \xi_i = \max\big(0,1 - y_i(w^\top x + b)\big)$$

The complete final loss function and the gradient of the classification model is:

$$J(x) = \min_{w,b} \frac{\|w\|^2}{2} + C\sum_{i=1}^{n} \max\big(0,1 - y_i(w^\top x + b)\big)$$

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

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

3. GD: Gradient Descent

True gradient descent is a batch algorithm, slow but sure

$$w := w - \eta \nabla L(w) = w - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla L_i(w) \qquad \text{(a)}$$

η: the hyper-parameter learning rate

Simple procedure:

i. Initialize parameter w and learning rate η.
ii. Loop: while an approxiamte minimum is no obtianed do (a)
iii. end

However, when the dataset is so large, doing a fully gradient descent will cost too much time and memory. So we could try MSGD.

4. MSGD: Minibatch Stochastic Gradient Descent

Rather than using a single point (SGD), use a random subset where the size is less than the original data size

$$w := w - \eta \frac{1}{|S_k|} \sum_{i \in S_k} \nabla L_i\,(w), \qquad where\ S_k \subseteq [n]$$

Like the single random sample, the full gradient is approximated via an unbiased noisy estimate. Random subset reduces the variance by a factor of $\frac{1}{|S_k|}$, but is also $|S_k|$ times more expensive.

Simple procedure:

i. Initialize parameter w and learning rate η.
ii. Loop: while an approxiamte minimum is no obtianed do
   a) Randomly select $|S_k|$ examples in the training set
   b) $w := w - \eta \frac{1}{|S_k|} \sum_{i \in S_k} \nabla L_i\,(w)$
iii. end

III. Experiment

Data Set Source:
https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a

The data set, with 123 features, includes 32561 samples for training and 16281 for testing

Experiment Steps:

a) Load and preprocess the experiment data.
b) Define and initialize hyper-parameters: learning rate, the max epochs of training, batch size.
c) Initialize model parameters 1-diemond matrix as ones.
d) Code the loss function and optimizer.
e) Update the model parameters w.
f) Calculate the training loss and testing loss using current model. Record them.
g) Output monitor information.
h) Based on GD or MSGD repeat (e) to (f) until the train end.
i) Output the model's performance information.
j) Drawing graph of training losses and testing losses, as well as with the number of iterations.

Experiment Hyper-parameters Selection Scheme:

1. Logistic Regression

Hyper-parameters Tables

Learning Rate	0.01
Iteration Time	1000
Batch Size	100

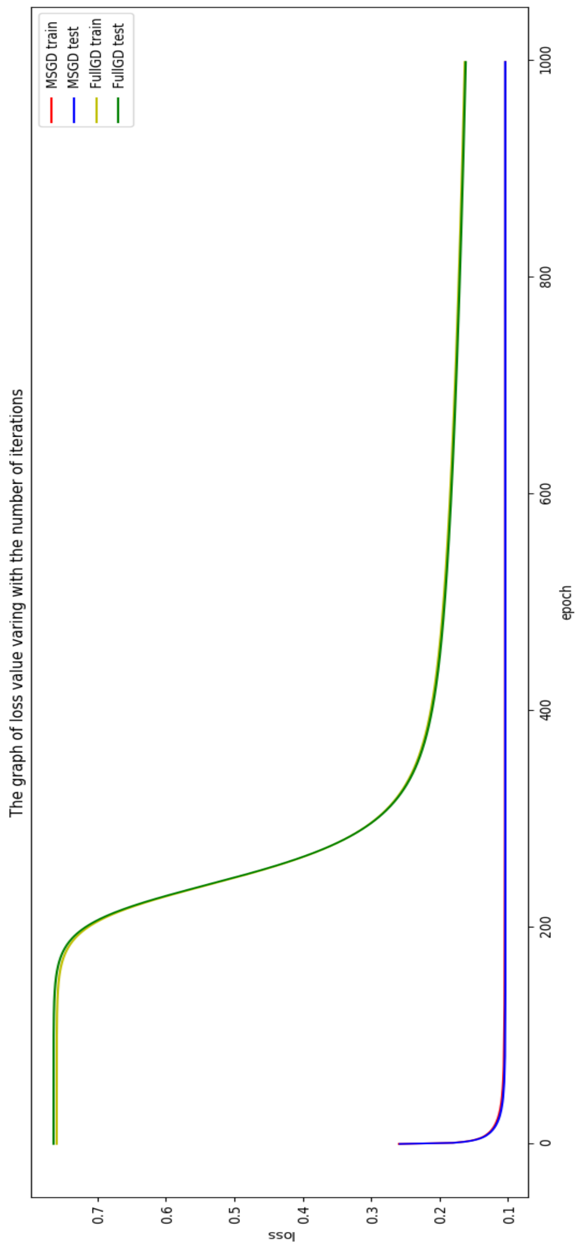
Result reports:

i. MSGD

	Precision	Recall	F1-score	Support
Positive	0.8825	0.9270	0.9042	12435
Negative	0.7179	0.6009	0.6542	3846
Avg/total	0.8436	0.8499	0.8451	16281

ii. Full Gradient Descent

	Precision	Recall	F1-score	Support
Positive	0.8261	0.8817	0.8530	12435
Negative	0.5111	0.3999	0.4487	3846
Avg/total	0.7517	0.7679	0.7575	16281



2. SVM

Hyper-parameters Tables

Learning Rate	0.0001
Iteration Time	1000
C	0.7
Batch Size	100

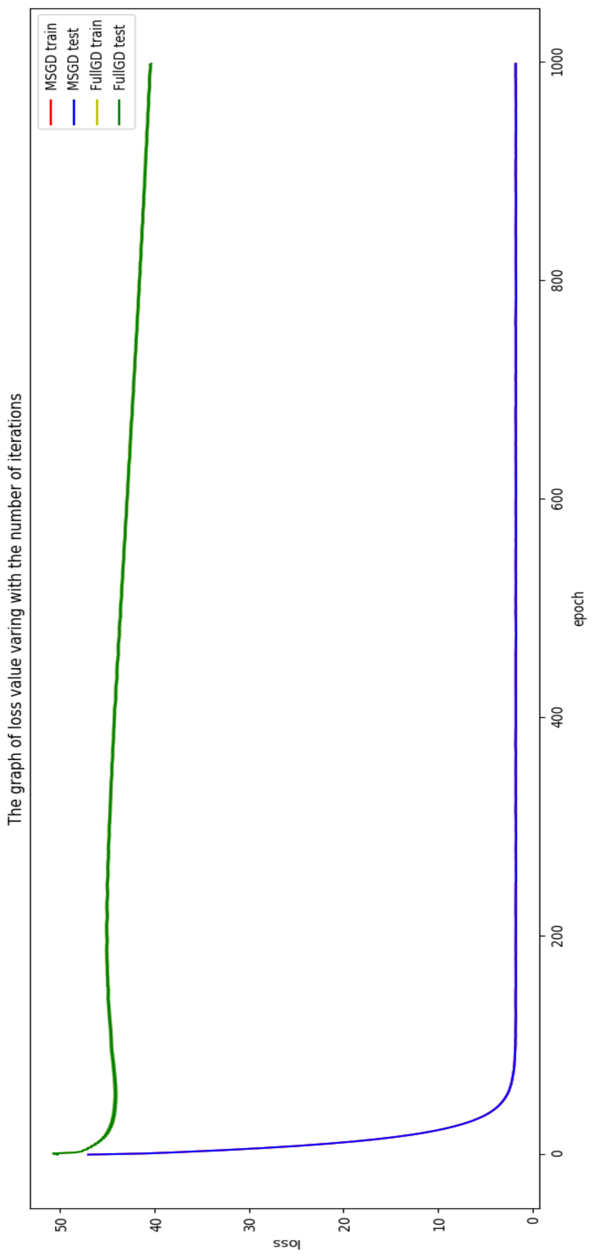
Result reports:

i. MSGD

	Precision	Recall	F1-score	Support
Positive	0.8618	0.9464	0.9021	12435
Negative	0.7460	0.5094	0.6054	3846
Avg/total	0.8345	0.8431	0.8320	16281

ii. Full Gradient Descent

	Precision	Recall	F1-score	Support
Positive	0.9360	0.8096	0.8682	12435
Negative	0.5715	0.8211	0.6739	3846
Avg/total	0.8499	0.8123	0.8223	16281



IV. CONCLUSION

Obviously, the model with SVM is better than logistic regression. And in both methods, using MSGD can converge the loss curve more fast.

Logistic Regression and Support Vector Machine are two of the foundational methods in machine learning area and both of them make a huge contribution to the development of machine learning algorithm. 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.