$$g_{\mathbf{w}}(\boldsymbol{x}_i) = \begin{cases} -y_i\, \boldsymbol{x}_i & 1 - y_i(\mathbf{w}^T \boldsymbol{x}_i + \mathrm{b}) \geq 0 \\ 0 & 1 - y_i(\mathbf{w}^T \boldsymbol{x}_i + \mathrm{b}) < 0 \end{cases}$$

$$g_b(\boldsymbol{x}_i) = \begin{cases} -y_i & 1 - y_i(\mathbf{w}^T \boldsymbol{x}_i + \mathrm{b}) \geq 0 \\ 0 & 1 - y_i(\mathbf{w}^T \boldsymbol{x}_i + \mathrm{b}) < 0 \end{cases}$$

I. EXPERIMENT

A. Dataset

This a9a data set used in the experiment, in which the training set contains 32561 samples and the verification set contains 16281 samples. Each sample has 123 Implementation

B. Realization

In the experiment, we first use the logistic regression model. First, we use all zero initialization for W. batch_ size =100; Learning rate = 0.001; Cycle 200 times.

Calculate the accuracy of the model prediction verification set after each cycle (where the judgment condition is g() ≥ 0.5 and the loss function value, as shown in Figure 1 and Figure 2.

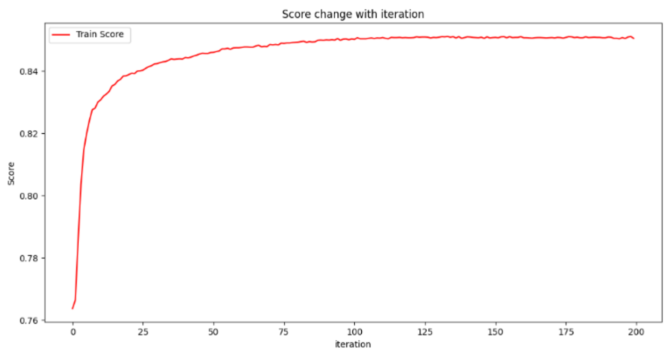

Figure1 Score of logistic regression model prediction verification set

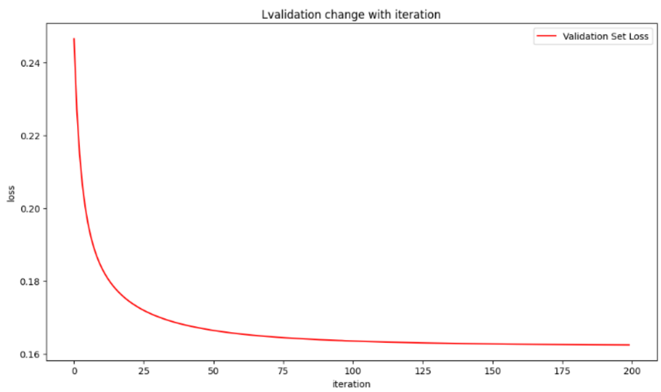

Figure 2 Loss function value of validation set corresponding to logistic regression model

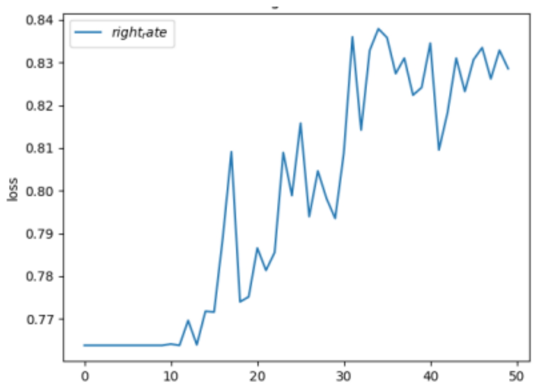

Figure3 Accuracy of support vector machine model prediction verification set

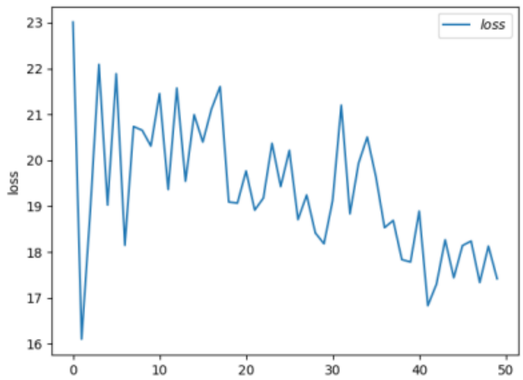

Figure4 Loss function value of validation set corresponding to support vector machine model

II. CONCLUSION

This The work done includes:

- ✧ It is found that there is an error in the code in baseline. The loss function does not correspond to the gradient. The solution is to change the sample: - 1:1 distribution to a 0-1 distribution, map - 1 to 0, and then change the loss function;
- ✧ Only one batch can be taken at a time, and there is no sample available: change the random sampling code to ensure that all samples can be taken;
- ✧ Adjust the parameters to increase the accuracy rate from 0.7637737239727289 to 0.8505005835022419

From this experiment, I learn a lot!!! Thanks for assistants' help!!!

華南理工大學

South China University of Technology

机器学习实验报告

学院：软件学院

专业：软件工程

作者:

王子通

指导教师:

谭明奎

学号:

202030484151

班级:

2020 级 2 班

2022-10-12

Logistic Regression and Support Vector Machine

Abstract: In this experiment, we use a9a data in LIBSVM Data, including training set and verification set. Next, logistic regression and support vector machine models are used respectively, and the corresponding parameters of the model are obtained after training with the training set. Finally, we use the trained model to predict the data on the verification set, and calculate the accuracy and loss function value for comparison.

I. Instruction

The a9a data sets used in this experiment include training set and verification set. In this experiment, the training set will be used to obtain the model parameters, and the verification set will be used to calculate the value of the loss function and prediction accuracy of the trained model, which will be used as the evaluation of the trained model.

II. Method and theories

In this experiment, we will use logistic regression and support vector machine models.

1.For the logistic regression model, we use a probability function as follows

$$h_{\mathbf{w}}(\mathbf{x}) = g(z) = g(\mathbf{w}^T\mathbf{x})$$

Where z = $\mathbf{w}^T\mathbf{x}$, and

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

So we can get

$$P(y|\mathbf{x}) = \begin{cases} g(\mathbf{w}^T\mathbf{x}) & y = 1 \\ 1 - g(\mathbf{w}^T\mathbf{x}) & y = -1 \end{cases}$$

$$P(y_i|\mathbf{x}_i) = g(y_i\mathbf{w}^T\mathbf{x}_i)$$

After derivation, it is found that

$$\max_{\mathbf{w}} \prod_{i=1}^{n} P(y_i|\mathbf{x}_i) = \min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \log(1+e^{-y_i\mathbf{w}^T\mathbf{x}})$$

Thus we determine the loss function

$$\boldsymbol{J}(\mathbf{w}) = L(\mathbf{w}) + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

Among

$$L(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log(1+e^{-y_i\mathbf{w}^T\mathbf{x}})$$

The parameters are also updated using the random gradient descent algorithm, where α Learning rate

$$\mathbf{w} = \mathbf{w} - \alpha\frac{\partial J(w)}{\partial w}$$

The derivative of $\boldsymbol{J}(\mathbf{w})$ is obtained:

$$\frac{\partial \boldsymbol{J}(\mathbf{w})}{\partial \mathbf{w}} = -\frac{1}{n}\sum_{i=1}^{n} \frac{y_i\mathbf{x}_i e^{-y_i\mathbf{w}^T\mathbf{x}_i}}{1+e^{-y_i\mathbf{w}^T\mathbf{x}_i}} + \lambda\mathbf{w}$$

2. Support vector machine wants to find a hyperplane and select two hyperplanes parallel to it to divide the data into two categories to realize classification. The two hyperplanes we choose are

$$\mathbf{w}^T\text{x+b=+1}$$
$$\mathbf{w}^T\text{x+b= -1}$$

Then we want to correctly classify and maximize the spacing between the two hyperplanes, and the spacing between the two planes is

$$d_{margin}=\frac{2}{||\mathbf{w}||}$$

We need to meet

$$\max_{\mathbf{w},b} \frac{2}{||\mathbf{w}||} \text{ s.t. } \mathrm{w}^T\text{x+b}\begin{cases} \geq 1 & y_i = +1 \\ \leq -1 & y_i = -1 \end{cases}$$

Equivalent to

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

$$\text{s.t. } y_i(\mathbf{w}^T\boldsymbol{x_i}\text{+b})\geq 1, \quad \text{i=1,2,3,…,n}$$

However, for better classification, relaxation variables are introduced ε_i. And make

$$\varepsilon_i\text{=max}(0,1\text{-}y_i(\mathbf{w}^T\boldsymbol{x_i}\text{+b}))$$

Finally, the problem becomes

$$\min_{\mathbf{w},b} L(\mathbf{w},b) = \min_{w,b} \frac{||\mathbf{w}||^2}{2} + \frac{C}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w}^T\boldsymbol{x_i} + \text{b}))$$

Where n is the sample size and C is the super parameter, which determines the influence of our constraints.

Then we use the random gradient descent algorithm to update the parameters

$$\mathbf{w} = \mathbf{w} - \alpha\nabla_w L(\mathbf{w},b)$$
$$\text{b} = \text{b} - \alpha\nabla_b L(\mathbf{w},b)$$

Among α Learning rate

Let $g_{\mathbf{w}}(\boldsymbol{x_i})=\frac{\partial \varepsilon_i}{\partial \mathbf{w}}$, $g_b(\boldsymbol{x_i})=\frac{\partial \varepsilon_i}{\partial b}$, thus

$$\nabla_w L(\mathbf{w},b) = \mathbf{w} + \frac{C}{n}\sum_{i=1}^{n} g_{\mathbf{w}}(x_i)$$

$$\nabla_b L(\mathbf{w},b) = \frac{C}{n}\sum_{i=1}^{n} g_b(x_i)$$

Then find out $g_{\mathbf{w}}(\boldsymbol{x_i})$, $g_b(\boldsymbol{x_i})$