
# The Experiment Report of Machine Learning

SCHOOL: SHIEN-MING WU SCHOOL OF INTELLIGENT ENGINEERING

SUBJECT: The super robot Everest class

*Author:*

Xilang Zeng

*Supervisor:*

Mingkui Tan

*Student ID:*

202130461984

*Grade:*

Undergraduate

2024-3-28

# Experiment 2: Logistic Regression and Support Vector Machine

***Abstract*—This experiment involves using logistic regression models and support vector machine models to handle binary classification problems with the a9a dataset in LIBSVM Data. In this experiment, we tried the method of stochastic gradient descent to update the model parameters and used the validation set to test the loss value after training.**

## I. INTRODUCTION

This section introduces the problem to solved and leads the reader on to the main part. Detailed motivation is necessary. What's more, you can show your expected results and contributions.

This experiment utilizes two methods, logistic regression models and support vector machine models, to conduct binary classification experiments on the a9a dataset within LIBSVM Data. In logistic regression, we used the sigmoid activation function as the logistic function. Afterwards, we calculate the cross-entropy loss for positive and negative samples and update the model parameters using stochastic gradient descent.

To accommodate some samples that do not meet linear constraints, we use the hinge loss function in the support vector machine experiment and calculate its gradient. By updating the parameters of the support vector machine through stochastic gradient descent, we achieve the purpose of classification. Finally, we are able to effectively implement binary classification problems through these two methods.

## II. METHODS AND THEORY

### A. Reading the experimental data

The a9a dataset from LIBSVM Data is read using the Python library sklearn. The dataset has already been pre-divided into a training set and a validation set. After reading, it is divided into feature data and label data. An additional dimension is added to the end of the feature data, with all data in this dimension being one, introducing a bias term to the model.

### B. Choosing the loss function

For the logistic regression experiment, we selected the cross-entropy loss function as the loss function. Cross-entropy is a measure of "surprise," quantifying the average level of "unexpectedness" when we know the true value of y. When the output matches our expectations, our level of surprise is relatively low; when the output does not meet our expectations, our level of surprise is higher. The mathematical expression for the sigmoid function is

$$\hat{y}_i = \sigma(z_i) = \frac{1}{1+e^{-z_i}}$$

The mathematical expression for cross-entropy loss is

$$L(y_i, \hat{y}_i) = -[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

The average loss across the entire dataset is

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

From this, we can derive the gradient function needed for gradient descent

$$\nabla_\theta J(\theta) = \frac{1}{N} X^T \cdot (\hat{y} - y)$$

For the support vector machine experiment, our loss function is the hinge loss function. In machine learning, hinge loss is a loss function that is commonly used in maximum margin algorithms, which are important algorithms used by support vector machines. The mathematical expression for the hinge loss function is

$$\max\left(0, 1 - y^{(i)}\left(w^T x^{(i)} + b\right)\right)$$

Here, we add a regularization term to the loss function, the mathematical expression becomes

$$L(\theta) = t \cdot \frac{1}{N}\sum_{i=1}^{N} \max\left(0, 1 - y_i \cdot (X_i \cdot \theta)\right) + \frac{1}{2}\sum \theta_i^2$$

Then, we calculate the gradient of the loss function

$$\nabla_\theta L(\theta) = \theta - C \cdot \frac{1}{N} \sum_{i:1-y_i(X_i\cdot\theta)>0} (-y_i X_i)$$

Where $X$ is the input data, $y$ is the labels, $\theta$ is the model parameters, and $C$ is the regularization parameter.

C. Update Model Parameters

We update the model parameters using the method of stochastic gradient descent. For logistic regression and the support vector machine, the mathematical formula for updating parameters is

$$\theta = \theta - \alpha \cdot \nabla_{\theta} L(\theta)$$

D. Validating with the validation set

For the part about validation on the validation set, we calculate the loss value using the samples and their labels from the validation set, thereby determining the effectiveness of this round of gradient descent. The samples from the validation set are used only for validation and are not used to update the parameters.

E. Printing the loss curve

Use Python's Matplotlib library to plot how the training and validation loss change with the number of iterations. Through the graph, we can visually see how the loss on the training and validation sets changes with the number of iterations, which is very helpful for tuning and evaluating the model.

## III. EXPERIMENT

A. Dataset

Both experiments utilized the a9a dataset from LIBSVM Data, consisting of 48,842 entries. Data was transformed from 14 original features to 123, and split into training and testing sets at a 2:1 ratio, with a9a serving as the training set for classifier model training; a9a-t was the test set for model classification performance evaluation. The dataset contains two categories, with labels -1 and 1 indicating whether an individual's annual salary exceeds 50K, where 1 signifies exceeding 50K and -1 does not.

B. Implementation

First, we load the dataset through the Python library sklearn, which has been pre-divided into training and validation sets. Next, we split the dataset into feature data and label data. Additionally, we append a column of ones to the end of each row of the feature data, to introduce a bias term and simplify mathematical representations.
Then, we define and calculate the logistic loss function and its gradient. The model parameters for logistic regression are updated through stochastic gradient descent, with each iteration's training and validation set loss values recorded. Finally, the loss values are plotted as curves. We also adjusted the learning rate to test the classification performance of logistic regression under different learning rates. The curves for the loss function are shown in Figures 1, 2, and 3, respectively.

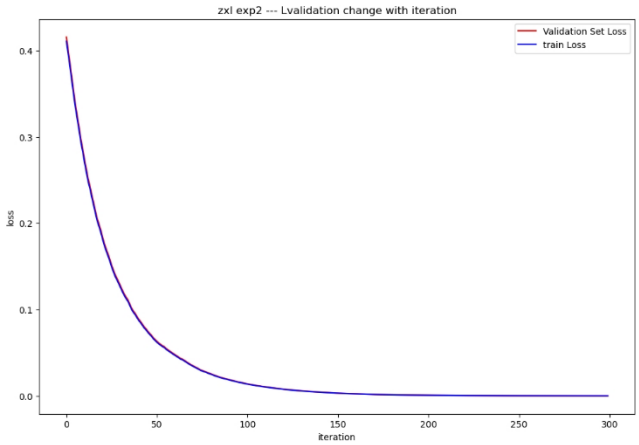


Figure. 1. Loss curve (learning rate 0.0001, iterations 300).

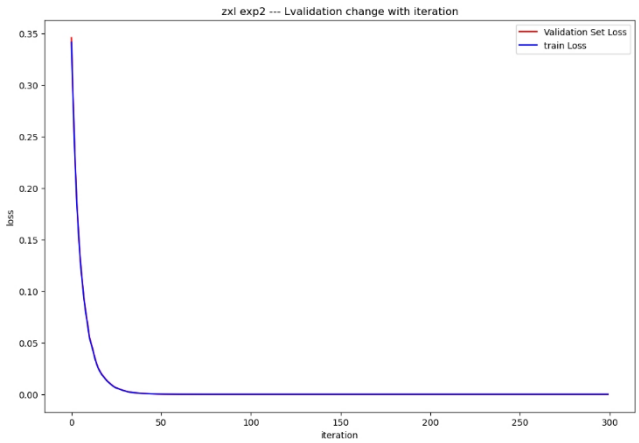


Figure. 2. Loss curve (learning rate 0.0005, iterations 300).

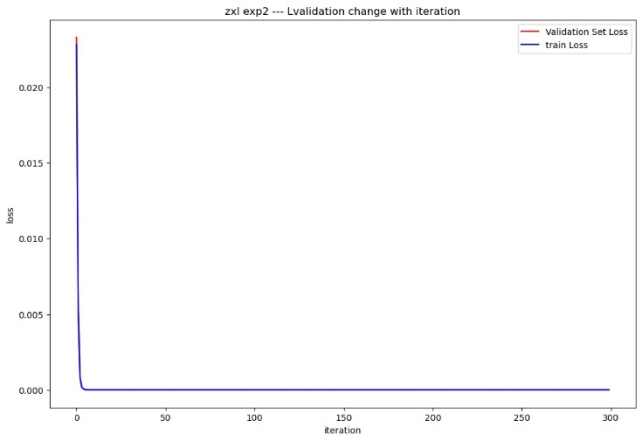


Figure. 3. Loss curve (learning rate 0.005, iterations 300).

From the curve graph, we can observe that the higher the learning rate, the faster the loss function decreases, and the final loss value converges within a very low range. This indicates that logistic regression performs very well in classifying this dataset.
We also calculated the classification accuracy of the logistic regression model by computing the predicted values using the sigmoid function, and then transforming these predictions into class labels (1 or -1) based on a threshold (0.5). Afterwards, the function compares the

predicted class labels with the actual class labels `y`, calculating the proportion of matches, which is the model's classification accuracy. The classification accuracy at different learning rates is shown in Table 1.

TABLE 1
Classification Accuracy of Logistic Regression Model

| Learning Rate | Iterations | Accuracy |
|---|---|---|
| 0.0001 | 300 | 0.76 |
| 0.0005 | | 0.76 |
| 0.005 | | 0.76 |

It can be observed that different learning rates have no effect on the classification accuracy. Even though the loss values of the loss function are low, the classification accuracy does not exceed 0.8, indicating that logistic regression performs moderately when evaluated using classification accuracy.

Next, we defined and calculated the hinge loss function and its gradient. We then updated the model parameters of the support vector machine using stochastic gradient descent, recording the loss values for both the training and validation sets at each iteration. Finally, we plotted these loss values as curves. We also adjusted the learning rates to test the classification performance of the support vector machine at different learning rates. The curves of the loss function are shown in Figures 4, 5, and 6 respectively.

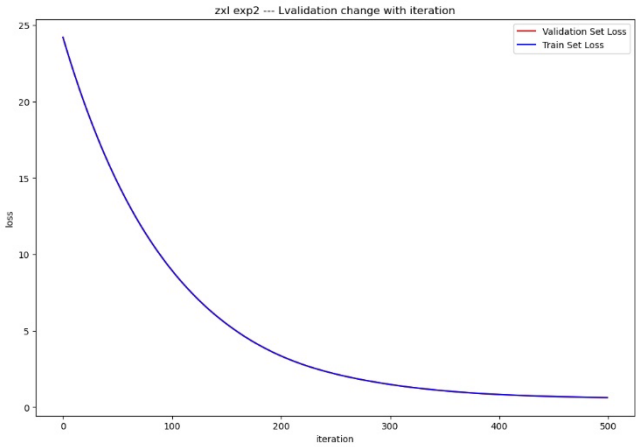


Figure. 4. Loss curve (learning rate 0.005, iterations 500).

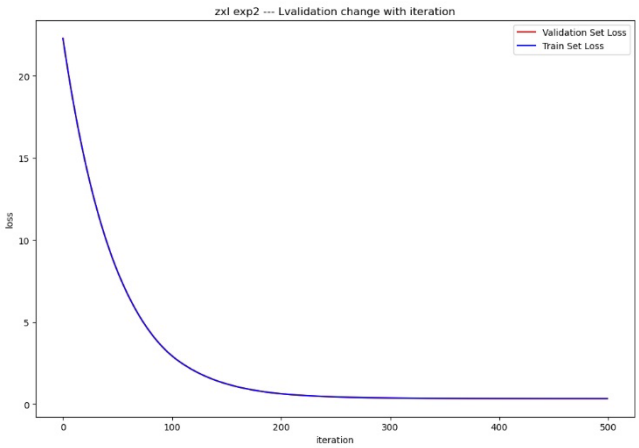


Figure. 5. Loss curve (learning rate 0.01, iterations 500).

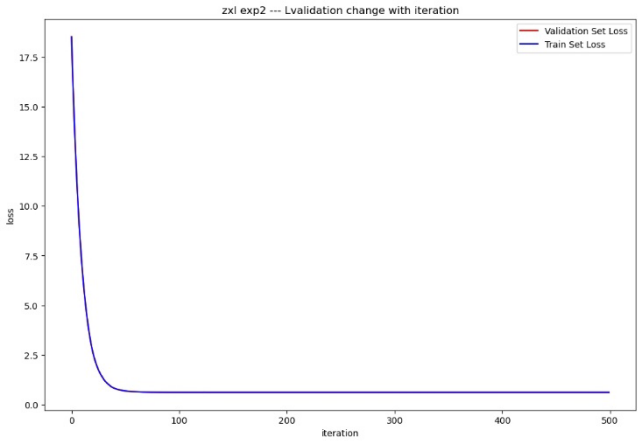


Figure. 6. Loss curve (learning rate 0.05, iterations 500).

Similarly, we also calculated the classification accuracy, and the results are presented in Table 2.

TABLE 2
Classification Accuracy of SVM Model

| Learning Rate | Iterations | Accuracy |
|---|---|---|
| 0.005 | 500 | 0.76 |
| 0.01 | | 0.76 |
| 0.05 | | 0.76 |

From the data in the curve graphs and tables, we can draw conclusions similar to those with logistic regression.

## IV. CONCLUSION

Overall, logistic regression and support vector machines are competent for binary classification tasks. We can optimize the parameters of logistic regression and support vector machines using stochastic gradient descent to enhance their classification capabilities. Although they achieve relatively high classification accuracy, their classification capacity is still lacking. Larger learning rates can accelerate the convergence of stochastic gradient descent but have no impact on classification accuracy, which may be determined by the inherent performance of the models.