

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.***

## 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.

## METHODS AND THEORY

### 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.

### 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

The mathematical expression for cross-entropy loss is

The average loss across the entire dataset is

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

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

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

Then, we calculate the gradient of the loss function

Where is the input data, is the labels, is the model parameters, and is the regularization parameter.

### 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

### 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.

### 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.

## EXPERIMENT

### 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.

### 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.

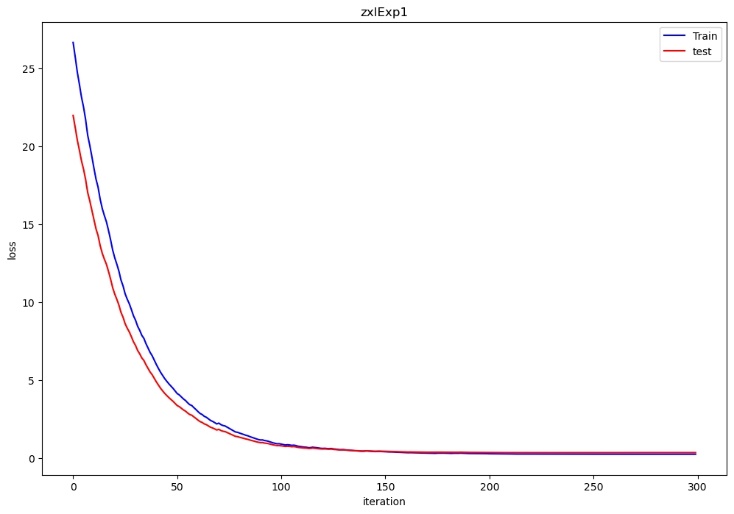
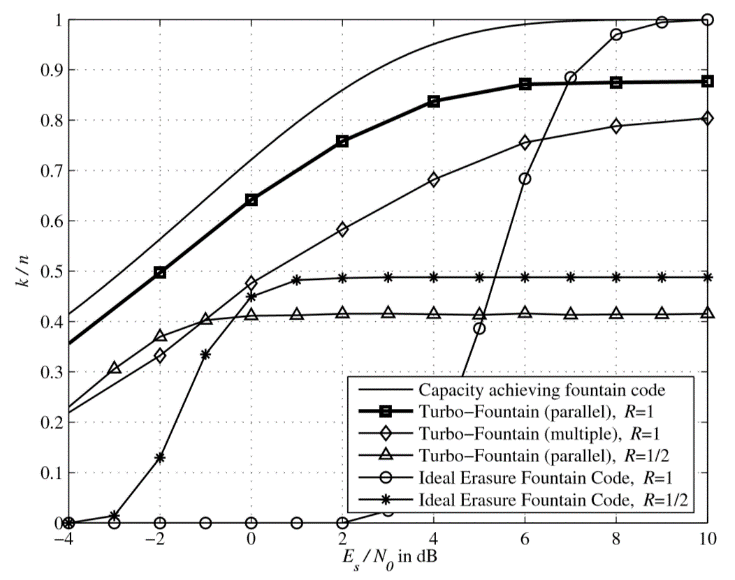
Figure. 1. Loss curve (learning rate 0.0005, iterations 300).

Figure. 2. Loss curve (learning rate 0.001, iterations 300).

## CONCLUSION

This section summarizes the paper. In our experiments, you can also write your gains and inspirations in here.

Figure. 1. Simulation results on the AWGN channel.