Binary Classification with Logistic Regression

Department of Computer Software Engineering
College of Engineering, Hanyang University
Gaon Choi(최가온), Student ID: 2019009261

Introduction

In this experiment, the goal is to create a model that performs binary classification on two-dimensional datasets through Logistic Regression. Implementation methods are largely divided into two types: element-wise learning, vector-wise learning. The former approaches each element directly in the list data structure, and the latter accesses data based on the concept of matrix through numpy.array in the Numpy library.

Experiment

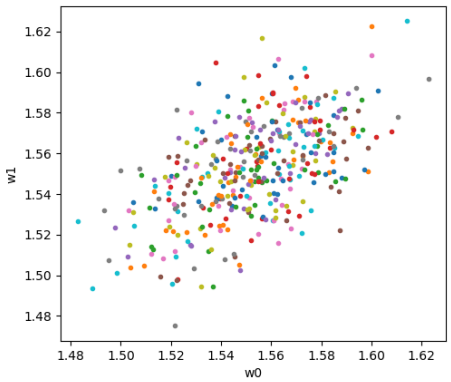
1. Time Comparison

An experiment was conducted to measure the time required in the case of m=1000, n=100, and k=5000 as basic environments. Element-wise method took about 81.51 seconds, which is 80~100 times slower than Vector-wise method, which took about 1.19 seconds.

```
PS C:\Users\USER\PycharmProjects\ai_models> python .\logistic_reg.py 1000 100 2000 element
Training with Element-wise Model..
Elapsed Time: 81.51273989677429s
PS C:\Users\USER\PycharmProjects\ai_models> python .\logistic_reg.py 1000 100 2000 matrix
Training with Vector-wise Model..
Elapsed Time: 1.1929991245269775s
```

2. Estimated unknown function parameters W & b

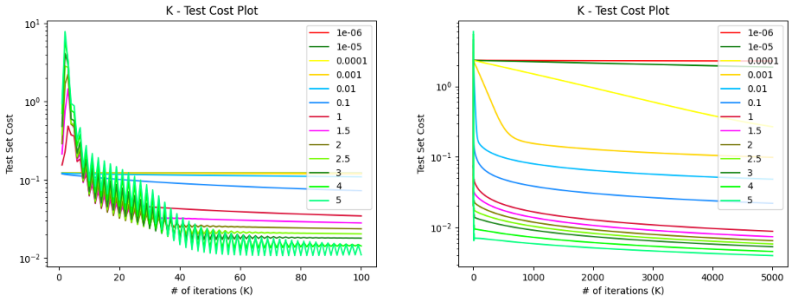
The time required in element-wise method and vector-wise method showed a large difference, but the values of the calculated weights(w, b) were the same. The same results are obtained because the inherent algorithm has the same principle, and the only difference is the format in which the data is processed. As a result of repeating a total of 400 experiments by generating a new dataset every time(m=1000, n=100, k=2500), it was empirically confirmed that the values of w1 and w2 were derived as similar values, respectively. When creating a new dataset, the weights were different every time because randomness existed when creating datasets. but according to the graph, it can be inferred that w1 and w2 have a positive correlation.




```
PS C:\Users\USER\PycharmProjects\ai_models> python .\logistic_reg.py 1000 100 2000
Training with Element-wise Model..
Estimated Parameters:
W:  [1.29034464 1.2929347 ]
B:  -0.01256543007671731
Training with Vector-wise Model..
Estimated Parameters:
W:  [1.29034464 1.2929347 ]
B:  -0.012565430076717314
```

3. Empirically determined (best) hyper parameter, α

From a microscopic perspective, that is, at the beginning of learning, when the value of the learning rate is large, the problem of oscillation is larger. Roughly, it was found that it occurs on a large scale when the learning rate is 1.5 or higher. From a macro point of view, that is, when sufficient learning iteration is made (K≥1000), and when the learning rate is smaller, the degree to which the cost value decreases gradually was found to be insignificant even though the number of iterations is increased. On the other hand, the larger the learning rate, the relatively significantly lower the cost, and the asymptotic value was reached in the subsequent iteration process.



4. Accuracy

4-1. Accuracy comparison by the value of m (the size of train set)

While keeping the value of n and k constant, the trend of accuracy when m is gradually increased is as follows.

	m=10, n=1000, K=5000	m=100, n=1000, K=5000	m=10000, n=1000, K=5000
Accuracy (with 'm' train samples)	100.00%	99.85%	99.93%
Accuracy (with 'n' test samples)	93.21%	99.02%	99.91%

4-2. Accuracy comparison by the value of K (the number of iterations)

One hundred independent experiments were performed on three cases where the value of K was 20, 200, and 2000, respectively. For each experiment, the dataset was newly created[1] and proceeded every time. The accuracy below is the average of the results derived from 100 experiments.

	m=10000, n=1000, K=10	m=10000, n=1000, K=100	m=10000, n=1000, K=5000
Accuracy (with 'm' train samples)	94.82%	99.03%	99.79%
Accuracy (with 'n' test samples)	94.03%	98.76%	99.69%

Discussion

Comparing various implementation methods based on elapsed time[2] is practically significant. In addition, it is considered useful in actual experiments to visually observe how the cost falls for each iteration, and to identify the correlation by associating various factors(m, n, k, etc.) with independent variables, dependent variables, and control variables.

---

[1] If not, the model returns the same accuracy every time.

[2] It is estimated that the performance difference occurred by the optimization technique used in the Numpy.