CS5014 P2 Classification

180007044
University of St Andrews
St Andrews, UK

INTRODUCTION

As the requirement of the practical, there are two classification models implemented. The first one is a binary logistic regression model, which is implemented with gradient decent. Another one is a multi-classification model implemented by the multi-layer perception classifier provided by the neural network library of sklearn. This report will indicate the detail of implementation, evaluation and performance of two models respectively and critically compare two methods.

BINARY CLASSIFICATION MODEL

Logistic regression models produce bounded results in the boundary of [0, 1], hence it could be a virtue choice for binary classification tasks.[1]

Data Preparation

Clean Data

Basically, there may be supposed to be 3 elements taken into consideration in this phase: missing values, outliers and data types.

However, since there are 768 columns in the input dataset, it is difficult to explore outliers for each feature. Besides, according to the output printed by functions *info()* and *isnull(),* there are no missing values in both training and testing dataset and all features contain only float data.

Feature Scaling

To scale all features into a similar range, all input features in the training dataset are normalized by their means, so that they are kept in the range of [-0.5, 0.5].

Feature Selection

Because of the large number of input features, it might be impossible to select features according to the correlation coefficients among them.

Instead, in this practical, the selecting strategy is divided 768 features into 3 subsets according to their meanings. Specifically, 3 sets contain mean values, minimal values and maximal values of signals produced by 256 components respectively. This decision is made based on the assumption that all components have equal status, which means they pose similar influence on the output. In later parts, these 3 data sets would be separately used to fit the logistic regression model. The model with best performance would be chosen to predict for the test data in *XToClassify.csv.*

Split Dataset

To evaluate the performance of models, it may be helpful to split data to a training set and a testing set. As there are only 80 entries in the dataset, the proportion is set to 8: 2, in case of insufficiency of data for training.

To distinguish this testing set from the final one for predicting, this split testing set would be named as a validation set.

Visualization

Data are not visualized in this practical because visualizations are generally used to explore some attributes of certain features. However, since there are 768 similar features involved in this case, it may be impossible to select some features for visualization. Although comparing all or some features in a coordinate system is also an executable option, it is difficult to judge if it is useful, since all features have different meanings and each component may have own working mechanisms, which may make it unreasonable to compare them horizontally.

Model Training and Evaluation

As that indicated before, a logistic regression model would be trained for binary classification. The algorithm utilized in this case is gradient descent.

Parameter Initialization

Combined with regularization, the cost which needs to be minimized is:

$$\mathrm{J}(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(h(x^{(i)}) - (1-y^{(i)})\log(1-h^{(x)})] + \frac{\lambda}{2m}\sum_{j=1}^{m}{\theta_j}^2$$

There are 4 parameters relate to the model need to be taken into consideration: coefficients of the regression model, which is also known as θ, the learning rate, known as α, the regularization parameter λ and the threshold of the gradient.

To make sure that the training process steps towards a positive direction, all coefficients of the model are initially set to 0, which means all features would not influence the result at the start. In following phases, weights of features would be adjusted by training.

The learning rate is also initialized as 1, which would change in training phase. Similarly, the regularization parameter would also be set 0 and changed based on the optimization requirement.

The threshold of gradient is set to 0.01 initially. It would also change for optimization if it is necessary

Evaluation

It may be vital to decide the evaluation strategy before training models, so that they could be optimized according to their performance.

In this practical, the performance of models would be judged by accuracy of the prediction for the validation set, which is calculated by:

$$Accuracy = \frac{True\ Positive + True\ Negative}{All\ Prediction}$$

This justification is based on that accuracy is a kind of general way to evaluate classification model. [1] Besides, according to the result of evaluation, which would be demonstrated in later parts, accuracy is an index which may be convincing enough for the evaluation of classification models trained in this practical.

Besides, time spent on training would also be recorded as a reference of optimization. It would not be a definition index because it could be influenced by hardware.

Training

Using gradient descent algorithm, 3 models fitted by 3 feature subsets are initially trained with learning rate 1, regularization parameter 0 and gradient threshold 0.01. Their performance is listed in Table 1.

Table 1. Performance of initial models

Features	Loss	Accuracy	Time (s)
Mean value of signals	5.18e-6	1.0	0.041
Min value of signals	3.17e-4	1.0	0.039
Max value of signals	1.23e-4	1.0	0.039

Optimization

As per Table 1, all 3 models could accurately and correctly classify the input in the validation set, hence it may be difficult to judge if there are overfitting or underfitting problems. So, it is probably not necessary to optimize them by changing regularization parameters.

In this case, optimization would focus on decreasing loss and elapsed time by changing learning rate and the gradient threshold.

Learning rate: Currently, the learning rate is set to 1. To enhance the performance of models, the learning rate would be changed to 0.5 and 2 to detect the better direction. Then according to corresponding result, it would be adjusted by rule of dichotomy until that the loss would not decrease no matter enlarge or shrink it. After several iterations, learning rates are finally set to 2.7, 4.1 and 3.3 respectively.

Table 2. Performance of models trained by most suitable learning rates

Features	Most Suitable Learning Rates	Loss	Accuracy	Time (s)
Mean value of signals	2.7	6.23e-14	1.0	0.041
Min value of signals	4.1	7.56e-14	1.0	0.039
Max value of signals	3.3	6.94e-12	1.0	0.039

Threshold of Gradient: Similar to learning rate, the threshold would change through 2 directions. However, during the optimization, the performance of models does not become better in an obvious extent. Therefore, the threshold is kept as 0.01 finally.

Choose a model

According to Table 2, the feature subset containing mean values of signals. Because it has the lowest loss and its elapsed time is also quite similar to others.

Classify Test Input

The final step is to classify the test input data with the chosen model.

The result is documented into Appendix 1.

MULTI-CLASS CLASSIFICATION MODEL

Although use one-vs-all algorithms could easily train logistic regression models, it requires a lot of iteration processes to identify multiple classes. The complexity of the algorithm would sharply increase with larger number of class indexes.

Under the circumstances, multi-layer perception (MLP) classifier which has been well implemented by sklearn's neural network library could be a more appropriate choice. It could make coding much easier for training and optimizing models.

Data Preparation

Basically, data preparation, including data loading and cleaning, feature scaling and selection and dataset splitting, is as same as the binary classification model. The only extra process is output data transformation. Because the raw output data is indexes of classes, it may be simpler to process if it could be transformed into a group of binary features.

Model Training and Evaluation

As that mentioned before, an MLP classifier is created to train models.

Parameter Initialization

All parameters discussed in this part are all those of the MLP classifier.

Solver: This parameter presents the training algorithm of models. Since the dataset is extremely small in this occasion, 'lbfgs' may performance best among three possible values. [2]

Activation function: Although logistic function is widely utilized in classification tasks, it may not be a proper choice in this case, because it yields slow for it only produces positive values. Tanh is an eligible alternative sigmoidal function, a kind of functions with higher sensibility, since it could produce both positive and negative values. [3]

Hidden layer size: This parameter is initially set to 1, which means 1 hidden layer containing 1 neural unit. It would change during the optimization.

All these parameters are related to the size of the dataset; hence it may be reasonable to use the same classifier to train models fitted by 3 subsets with the same size.

Evaluation

Evaluation strategy is basically as same as that of binary classification models. However, according to performance of models, there may be randomness in training, leading to the instability. All performance data listed later would be average value of 5 tests.

Training

3 models are initially trained by the MLP classifier created before. Their performance is shown in Table 3.

Table 3. Performance of initial models

Features	Loss	Accuracy	Time (s)
Mean value of signals	1.47	0.41	0.085
Min value of signals	1.57	0.325	0.061
Max value of signals	1.72	0.22	0.051

Optimization

The optimization strategy is much simpler than that of the binary classification models as there is only one parameter, hidden-layer size could change for optimization.

Currently there is only 1 hidden layer with 1 unit involved. The basic strategy is to firstly find a best size of the first layer by doubling current number of units in each iteration, until there is no obvious increase on performance after doubling. The try to obtain more precise results with the rule of dichotomy.

Next, an attempt would be made to add number of hidden layers and determine a most proper number in the similar way. The result of optimization and corresponding performance of models are listed in Table. 4.

Table 4. Performance of models trained by most suitable hidden-layer size

Features	Most Suitable Hidden-Layer Size	Loss	Accuracy	Time (s)
Mean value of signals	28	0.00069	1.0	0.043
Min value of signals	28	0.00086	0.995	0.045
Max value of signals	28	0.00077	0.99	0.046

It could be noticed that the number of layers is still 1 because single layer is enough for simple classification like this one which just has 5 classes. Extra layers would also sharply increase the computation time. [3]

According to Table 4, the result calculated by a single layer is quite accurate, hence it may be not necessary to import other layers.

Choose a model

Table 4 shows that the model fitted by mean values of signals still has the best performance among 3 models.

Classify Test Input

The result of classification of test data is documented into Appendix 2.

COMPARISON

Finally, advantages and disadvantages would be summed up here.

Logistic regression model does have some advantages such as simplicity and controllability. It is based on a simple sigmoidal function which is easy to be manually transformed into codes. It also has advantages of sigmoidal functions such as sensitivity to change of inputs. [4] However, as it discussed in previous chapters, logistic regression yields slowly, and it would be quite complex and time-consuming if it is used to handle multi-class classification problems.

Compared with logistic regression models, MLP classifier is more eligible and easier to practically apply. Programmers could rapidly customize and train models by setting some parameters and sklearn also provides plenty of functions and attributes for evaluation. It could efficiently fit all kinds of classification tasks and run efficiently if parameters are appropriate. Whereas, the largest disadvantage is probably that it is a 'black box' system. Most of users may not know how it works and how to control it subtly. For example, in this practical, it may be difficult to find other ways to optimize models besides change hidden-layer sizes. Moreover, in this case, they could be confused about problems if something goes wrong. Another defect is that as the cost of high accuracy and efficiency, it may take more elapsed time and computational resources. [5]

In conclusion, for some simple classification tasks such as binary classification, logistic regression may be a simple but effective way to train models, while for those complex classification or problems which could not be solved by logistic regression, some neural network algorithms like MLP classifiers could be better choices.

EXTENSION-KNN ALGORITHM

Introduction

As an extension of this practical, a K-Nearest-Neighbors (KNN) algorithm is implemented and used in the multi-class classification.

KNN is an extremely simple non-parametric algorithm for classification. The basic methodology is to choose k entries whose input have smallest difference from the data waiting to be classified. Then the class mostly labeling these k entries would be considered as the result of classification.

Data Preparation

All data preparation work is as same as that of the binary classification discussed in previous chapters.

Although KNN is used for multi-class classification in this case, output data transformation is redundant because they would not be involved into numeric calculation.

Model Training and Evaluation

Parameter Initialization

K is the only parameter to set in this algorithm. However, the value of k may not have an obvious relation with performance of data, which more depends on the distribution of training data. In this occasion, its value is set to 1, which may be the safest choice because larger value would more likely to cause bias.

Evaluation

Since it is a non-parametric algorithm and it does not learn a parametric model, there is no concept of loss in this implementation. Besides, the complexity of the algorithm is mostly decided by the size of dataset, elapsed time is also meaningless in this case. So, accuracy is the only evaluation index.

Training

Similar to the other 2 implementations indicated before, 3 models are initially trained. The accuracy of all 3 models is 0.975.

Optimization

The current performance is quite virtue and the optimization would be skipped, because it may be difficult to optimize it by changing values of parameters.

Choose a model

As all 3 models have the same performance, like the other 2 solutions, the mean values of signals would be chosen to classify the testing dataset.

Classify Test Input

The result of classification of test data is documented into Appendix 3.

Conclusion

In this part, a comparison would be made between KNN algorithm and the other 2 parametric algorithms.

It is not difficult to find that KNN algorithm has some typical strong points of non-parametric algorithms. Compared with parametric ones, it could finish the classification more precisely, since it does not to balance the cost for produce a parametric model. Another reason is that the bias caused by outliers could be avoided in a large extent, and data cleaning is much easier. Besides, as a classification algorithm, unlike the other 2 algorithms discussed before, the number of possible classes would not influence the complexity. Specifically, it is much more friendly to rookies than the other 2 algorithms. In fact, it may be one of the simplest classification algorithms. The theory does not involve any advanced mathematic knowledge and formulas, making it extremely easy to understand and implement.

However, some disadvantages could also be drawn from the process in this practical. First one is that it would be time-consuming and computationally expensive to classify lots of data, which may be quite common for most of non-parametric algorithms. It needs to repeat the whole process for each input since it does not learn a model. Besides, it may be more likely biased if the classified data is close to the boundary, since its nearest neighbors would be variety. And as there are no weights for different features, there is also a risk that the result is influenced by irrelevant features. [6]

To sum up, KNN algorithm may be a better choice for simple classification tasks. Features of non-parametric algorithms make it not suitable to classify a large number of data. Besides, high possibility of bias make its performance not stable, which means it is not a robust algorithm compared to the other 2 parametric algorithms.

REFERENCE

[1] Teaching materials of the module CS5014

[2] "Sklearn. Neural_Network. Mlpclassifier — Scikit-Learn 0.20.3 Documentation". 2019. Scikit-Learn.Org. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

[3] Almahasneh A. “What are the advantages and disadvantages of the single-layer feed-forward, multilayer feed-forward and recurrent network in artificial neural network?” 2017. Quora. https://www.quora.com/What-are-the-advantages-and-disadvantages-of-the-singlelayer-feed-forward-multilayer-feed-forward-and-recurrent-network-in-artificial-neural-network

[4] "Comp.Ai.Neural-Nets FAQ, Part 2 Of 7: Learningsection - Why Use Activation Functions?". 2019.

Faqs.Org. http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-10.html.

[5] "Pros And Cons Of Neural Networks". 2019. Towards Data Science. https://towardsdatascience.com/hype-disadvantages-of-neural-networks-6af04904ba5b.

[5] "A Quick Introduction To K-Nearest Neighbors Algorithm". 2019. Noteworthy - The Journal Blog. https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.

CS5014 P2 Classification – Appendix 1

Predictions for Testing Data of Binary Classification

	Class ID	Class Name
0	1	plastic case
1	1	plastic case
2	1	plastic case
3	1	plastic case
4	1	plastic case
5	1	plastic case
6	1	plastic case
7	1	plastic case
8	1	plastic case
9	1	plastic case
10	0	book
11	0	book
12	0	book
13	0	book
14	0	book
15	0	book
16	0	book
17	0	book
18	0	book
19	0	book

CS5014 P2 Classification – Appendix 2

Predictions for Testing Data of Multi-Class Classification

	Class ID	Class Name
0	2	hand
1	2	hand
2		can’t identify
3	2	hand
4	2	hand
5	2	hand
6	2	hand
7	2	hand
8	2	hand
9	2	hand
10	0	air
11	0	air
12	0	air
13	0	air
14	0	air
15	0	air
16	0	air
17	0	air
18	0	air
19	0	air
20	3	knife
21	3	knife
22	3	knife
23	3	knife
24	3	knife
25	3	knife
26	3	knife

27	3	knife
28	3	knife
29	3	knife
30	1	book
31	1	book
32	1	book
33	1	book
34	1	book
35	1	book
36	1	book
37	1	book
38	1	book
39	1	book
40	4	plastic case
41	4	plastic case
42	4	plastic case
43	4	plastic case
44	4	plastic case
45	4	plastic case
46	4	plastic case
47	4	plastic case
48	4	plastic case
49	4	plastic case

CS5014 P2 Classification – Appendix 3

Predictions for Testing Data Predicted by KNN Algorithm

	Class ID	Class Name
0	2	hand
1	2	hand
2	1	book
3	2	hand
4	2	hand
5	2	hand
6	2	hand
7	2	hand
8	0	air
9	2	hand
10	0	air
11	0	air
12	0	air
13	0	air
14	0	air
15	0	air
16	0	air
17	0	air
18	0	air
19	0	air
20	3	knife
21	3	knife
22	3	knife
23	3	knife
24	3	knife
25	3	knife
26	3	knife

27	3	knife
28	3	knife
29	3	knife
30	1	book
31	1	book
32	1	book
33	1	book
34	1	book
35	1	book
36	1	book
37	1	book
38	1	book
39	1	book
40	4	plastic case
41	4	plastic case
42	4	plastic case
43	4	plastic case
44	4	plastic case
45	4	plastic case
46	4	plastic case
47	4	plastic case
48	4	plastic case
49	4	plastic case