Project 1: Machine Learning Higgs

Jiaan Zhu, Lei Wang, Qinyue Zheng
EPFL, Switzerland

***Abstract*—The Higgs boson, an elementary particle in the Standard Model of physics [1], produced by collision decays rapidly into other particles. The process of the decay is hard to observe directly. Therefore, machine learning methods can be usefully applied to verify whether the detected features are from a Higgs boson or just something else. Here we applied 6 classification techniques to answer the question. In our project, we found that with Ridge Regression, an accuracy of 0.831 and F1 value of 0.742 can be achieved.**

I. Introduction

In our works, we tried to answer the question whether the signal of Higgs Boson can be detected by machine learning methods or not. Applying 6 methods we discussed in class to the real-world dataset, we finally landed on the ridge regression, and distinguished Higgs Boson signals from background signals with an accuracy of 0.831 and an F1 value of 0.742.

In the following sections, our works in data processing, algorithm implementation and cross validation are presented.

II. Models and Methods

A. Data Processing

We firstly looked at the distribution of the individual features in the train set by using matplotlib.pyplot.hist. Noted that only 22nd feature is the categorial feature. So we divided each row of data in the train set into two groups with the given label{-1,1} and observed the distribution of the 22nd feature data in the same plot.

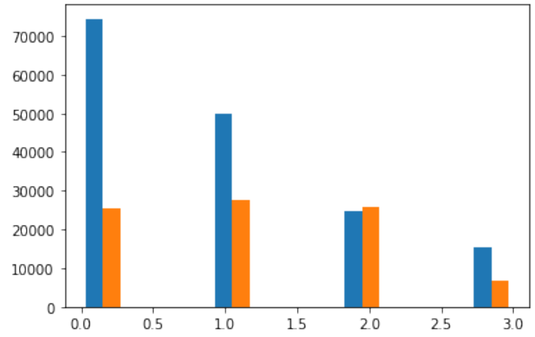


Figure 1: Distribution of 4 numbers in higgs group(label = 1, color = yellow) and background group(label = -1, color = blue)

We found that the distributions of {-1,1} were quite different in four categories as shown in Figure 1. Then we decided to divide the train set into four groups based on these category differences.

Observing the distribution of four sets of data using Boxplot, we found that in the first group and second group, the whole column of some feature was -999(which means NaN), so we only need to delete these features. Then going back to the remaining features, we noticed that -999 exists in only one feature at this time. We tried to use average value and median value to replace these NaN, however, we found that the accuracy of these methods was not as comparable as that of again splitting data into two groups by -999 and non-999 in first feature. So in the end we divided the train set into 8 groups as shown in Table I and trained separately. Now, there's no NaNs in each group of data.

Table I: 8 subgroups used in our train

group index	PRI_jet_num	DER_mass_MMC
1	0	999
2		others
3	1	999
4		others
5	2	999
6		others
7	3	999
8		others

We also noted that there are some outliers in the training data. Since the distribution of some features in train set are not normal distribution, so we apply IQR method to process these outliers. We treated any data other than $Q1 - ind * IQR(Q3 - Q1)$ and $Q3 + ind * IQR(Q3 - Q1)$ as outliers and removed them. The ind value selection process will be described in the Cross Validation section.

We also shifted the 22nd column of group2.

B. Function Implementation

We trained 8 sets of data which we have already pre-processed by using required functions, all functions are provided in the implementation.py document attached. Our results showed that gradient descent, stochastic gradient descent, and logistic algorithms were not as accurate as least-squares and ridge regression, so we ultimately chose to optimize them based on the ridge regression. We also noticed a significant increase in the accuracy of our predictions using the Data Augment method [2].

C. Cross Validation

A 4-fold cross validation method was used to verify the accuracy of our algorithms. $3/4$ of the data in the original train set is used for train(train set) and the remaining data is for validation(test set). Our plan is to optimize the following three parameters: 'ind' in IQR, 'degree' in data augment, and 'lambda' in ridge regression function.

First, we use the function check_ind_demo to find the best "ind", which ranges between 1 and 31 with an interval of 2. The train set after-removing outliers is used for train, and test set is for test. The best 'ind' value will be used for subsequent testing. Noted that in this step we set the degree of the augment to 1 and lambda to 0.

Then we use the function check_degree_demo to find the best degree, where 'ind' is set to the best value obtained in the previous step. lambda is still set to 0. The test range of Degree is from 1 to 11. The best degree value will be used in the next test.

Finally, we use the check_lambda_demo function to find the best lambda, where "ind" and degree are set to the best values obtained in the previous steps. The test range of 'lambda is set to $(1e(-10), 1)$, and the best lambda will be used for the test we finally submit.

The Figure 2 shows the result of the above optimization(Pick Group 7 as an example).

The optimization result is shown in Table II.

Table II: Optimization results of 8 subgroups

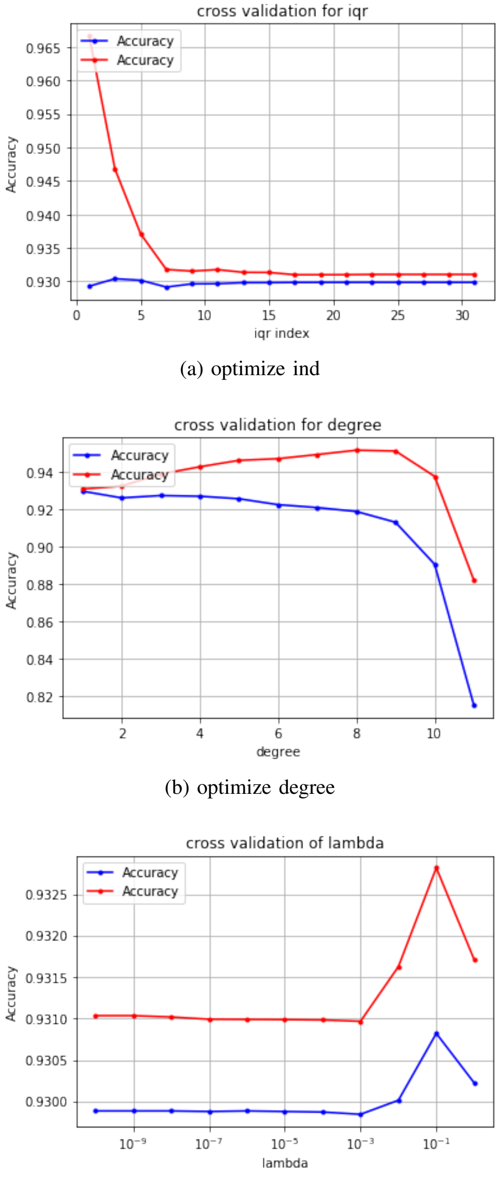
group index	best ind	best degree	best lambda
1	11	7	10^{-3}
2	7	8	10^{-9}
3	11	4	1
4	5	9	10^{-10}
5	17	2	10^{-8}
6	9	8	10^{-9}
7	19	1	10^{-1}
8	15	10	10^{-5}

III. Results

We applied the parameters obtained from the optimization above in our train. However, for optimized lambdas, the accuracy is lower than not setting lambdas(this may because of we only test a small range of lambdas). So we set all lambdas to zero in our final version, which achieved an accuracy of 0.831 and an F1 value of 0.742.

IV. Discussions

1) In theory, the accuracy of the prediction will increase after standardize or normalize the data, which is inconsistent with our results whenever it is processed before or after data augmentation. We speculate that this may be due to the characteristics of our data.



(a) optimize ind

(b) optimize degree

(c) optimize lambda

Figure 2: Optimization result of group 7. Blue line shows test accuracy, red line shows train accuracy. Subfigure (a) shows inds, subfigure (b) shows degrees, subfigure (c) shows lambdas

2) We didn't use feature selection in data processing, and performing feature selection may improve predictions [3].
3) The prediction correctness of Logic regression in our test is significantly lower than that of other methods. We suspect that a light change of the value of the threshold (i.e. in range 0.5 ± 0.1) may lead to better results.

Acknowledgements

The first author thanks Langlang for her moral support.

References

[1] Higgs boson FAQ - UT ATLAS group - UT austin wikis. [Online]. Available: https://wikis.utexas.edu/display/utatlas/Higgs+boson+FAQ

[2] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," vol. 6, no. 1, p. 60. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[3] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.