Machine Learning for Discovering the Higgs Particle

Huajian Qiu, Zhantao Deng, Yinan Zhang
EPFL, Switzerland

***Abstract*—The Higgs boson, which explains why other particles have mass, was discovered at CERN as by-products of high-speed collisions. Here, we apply machine learning techniques to classify the Higgs boson and the background. Different binary classification models and various data preprocessing methods are to be discussed.**

I. Introduction

The Higgs boson is named after Peter Higgs who predicted the existence of such a particle. Its existence was confirmed at CERN in 2013. When using machine learning to reconstruct the process of discovering the Higgs, we regard this challenge as a binary classification problem. Given 30 physical features, we are able to tell the Higgs from the background. Each data processing technique is tested on every one of our existing model. Test error is estimated by using 5-fold local cross-validation set. Our aim is to choose the best model and adopt appropriate data processing methods to improve the accuracy of detecting the Higgs.

The remainder of this report is structured as follows: We put forward several models that are suitable for solving binary classification problems at first. In addition to the methods learned in class, we have tried two more ways. One is Fisher Linear Discriminant Analysis, the other is Maximum Likelihood. Their principles are to be summarized briefly. Then we propose some data processing techniques and test how valid they are. Lastly, we choose the most effective ways to build our final model and give conclusions.

II. Preliminary Models

We use many models to distinguish the Higgs from the background, such as least squares, ridge regression, logistic regression, Fisher linear discriminant analysis and maximum likelihood.

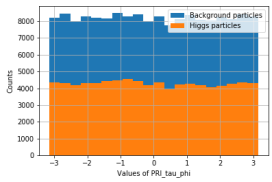
Linear discriminant analysis is a widely known method in statistics and pattern recognition to separate different classes. As one of its specialization, Fisher's Linear Discriminant(FLD) was widely used in face recognition community [1], [2]. The basic idea of FLD is to map high dimension data to a line which maximize the inter-classes distance and minimize inner-classes distance.

Another way is that we simply assume s particle and b particle obey Gaussian distribution but with different parameters. The train set is used to derive mean vectors and covariance matrices. For every data sample of the test set, we compute the likelihood of two set of parameters and find the more likely label.

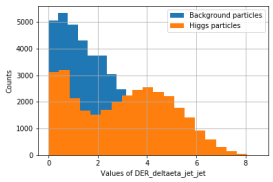
III. Data Preprocessing

A. Dropping Features

Aided by histogram, we try to visualize the distribution of each physical feature. The more distinguishable between the distribution of b particle and s particle, the better the feature is. Here we roughly classify all features into five categories as shown in Table I. Figure 1 shows what is a very good feature or a very bad feature intuitively. When plotting, rows containing missing values have been dropped.



(a) distributions of b particle and s particle are indistinguishable



(b) distributions of b particle and s particle are distinguishable.

Figure 1: Examples of good features and bad features

However, we find that dropping very bad features does not work well. So we decide to drop all features that contain missing values.

B. Filling Missing Values

The presence of missing values influences the process of classification. We intend to merge the train set and the test set, calculate the mean value of each column to replace missing values. When comparing results, mean values of the train set are used because we adopt local cross-validation to estimate our test error.

There is a small trick when classifying b particle and s particle. It is found that while a particle lacks the first feature DER_mass_MMC, it is almost b particle. The accuracy is more than 90%, which is better than any of our algorithms.

C. Feature Augmentation

The steps we build feature augmentation are as follows:

1) We add natural logarithm terms, exponential terms and cosine terms of each feature one by one. Keep those that can enhance the system performance.
2) Then we try three kinds of polynomial expansions. The first one is that we add quadratic terms of each feature. The second one is that the second and the

Very Good Features	DER_mass_vis, DER_deltaeta_jet_jet, DER_mass_jet_jet, DER_lep_eta_centrality
Good Features	PRI_jet_subleading_phi, DER_prodeta_jet_jet
Modest Features	PRI_met, PRI_jet_num, PRI_jet_leading_pt, DER_mass_MMC, DER_mass_transverse_met_lep, DER_pt_tot DER_pt_ratio_lep_tau, DER_met_phi_centrality
Bad Features	PRI_lep_eta, PRI_met_sumet, PRI_jet_subleading_pt DER_sum_pt, PRI_tau_pt
Very Bad Features	PRI_tau_phi, PRI_lep_pt, PRI_lep_phi, PRI_met_phi, PRI_jet_leading_eta, PRI_jet_leading_phi PRI_jet_subleading_eta, PRI_jet_all_pt, DER_pt_h, DER_deltar_tau_lepb, PRI_tau_eta

Table I: Feature Categories

third power of each feature data is taken into account. The third one is that the fourth and the fifth power is also added.

D. Normalization

For each feature data, we use the following formula to achieve normalization:

$$\frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

x_{min} is the minimum value of each column, while x_{max} is the maximum one.

E. Results

For each model, we apply different kinds of data processing methods mentioned above. We only display models that have good initial performances. 'FLD', 'LS', 'RR' refers to Fisher Linear Discriminant, Least Squares and Ridge Regression respectively. In the 'Base' column, raw original data is used. The accuracy of each system based on 5-fold local cross validation is shown in TableII [1]

	Base	A	B	C1	C2	C3	D
FLD	73.45	75.03	72.65	76.22	58.29	51.79	65.53
LS	74.46	77.13	74.72	78.07	52.82	56.85	65.59
RR	73.79	73.94	73.46	76.85	55.17	45.95	70.08

Table II: Performance of different models using various data processing methods based on 5-fold local cross validation.

IV. Our Final Model

Here is how we build our final model.

- LS and RR are proven to perform better than Fisher. Figure2 shows that LS outperforms RR. So we decide to use LS to detect the Higgs.
- Dropping features is a good way to improve our model performance. Features that contain missing values are neglected when building the final model.
- Mean value substitution and normalization are proven to be bad ideas and even impair model performance.
- When it comes to feature augmentation, quadratic, exponential, logarithm and cosine terms can improve accuracy. However, polynomial terms with more degrees fail because of overfitting.

So we drop features that contain missing values and add quadratic, exponential, logarithm and cosine terms. Then we use the least squares model to classify the Higgs and the background.

[1]Logarithm, exponential and cosine expansions have been included. 'A', 'B', 'C', 'D' refers to 'Dropping Features', 'Filling Missing Values', 'Feature Augmentation' and 'Normalization' respectively. 'C1' adds quadratic terms. 'C2' adds the second and the third power. 'C3' also adds the fourth and the fifth power. The unit is %

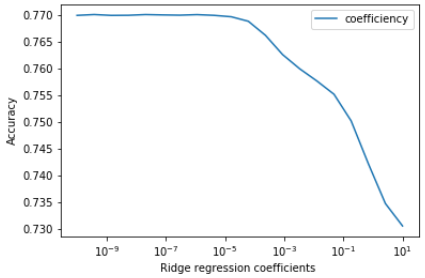


Figure 2: How the accuracy of RR varies with respect to λ.

V. Conclusion

Through two weeks of research, we find that least squares is the best model to detect the Higgs particle. But we may not find the best λ as we only tried a limited number of λ values. It is worth doing further research. We manage to obtain an accuracy of around 78.25% on Kaggle with feature augmentation and feature dropping. However, it is very difficult to further improve the performance of our model. One of possible reasons is the strongly nonlinear distribution of the dataset, which requires better classifiers.

References

[1] R. Fisher, "The use of multiple measures in taxonomic problems," *Ann. Eugenics. v7*, pp. 179–188.

[2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," Yale University New Haven United States, Tech. Rep., 1997.