Finding the Higgs Boson: From Machine Learning Approaches

Guanqun Liu
Dept. of Computer Science
EPFL
Lausanne, Switzerland
guanqun.liu@epfl.ch

Yixuan Xu
Dept. of Computer Science
EPFL
Lausanne, Switzerland
yixuan.xu@epfl.ch

Xianjie Dai
Dept. of Computer Science
EPFL
Lausanne, Switzerland
xianjie.dai@epfl.ch

***Abstract*—Based on the ATLAS experiment data provided by CERN, we discussed and trained different classification models to predict the Higgs Boson event to extract its signals from background noise. In this paper, we apply ridge regression to predict the occurrence of Higgs Boson from given event samples. Our proposed model utilized feature engineering techniques including data splitting according to dependent features, imputing missing values with k-means clustering, and polynomial expansion. Finally, we achieve a classification accuracy of 80.3% and an F1-score of 0.685 in the aicrowd competition.**

I. Introduction

The discovery of the Higgs Boson particle was finally claimed in the ATLAS and CMS experiments at CERN in 2012 [1] [2] and was awarded the 2013 Nobel Prize in Physics. The Higgs Boson particle can be generated as a by-product of proton collision. However, scientists find it difficult to conduct a deterministic observational analysis, because the Higgs Boson decays rapidly and the tau-tau decay signal of this particle often buries in background noise. The Higgs Boson challenge aims to explore machine learning algorithms on improving the physics discovery of Higgs Boson's decay event with background noise. Each event in the dataset is binary-labeled, with an event id indicating the event order. To dig out a possible relationship pattern from given features to the presence and absence of tau-tau decay events, we develop classification models based on different machine learning algorithms and improve their performance through data cleaning and feature engineering.

II. Data Preprocessing

The training set contains 250,000 event pairs of features and outcome labels. Each sample includes 30 features, and these features determine the label of events: whether it is tau-tau decay *'1'* or background noise *'-1'*. Observing event samples from the very beginning, we notice there exists missing values *(-999)* for some feature columns, representing invalidity of these features for that specific event sample. Grounded on the Higgs Boson data note [3], missing values that occurred in certain feature columns are related to one specific feature *'PRI_jet_num'* (Col. 23). This feature has a fixed value range $\{0,1,2,3\}$, for each event with a specific value, certain feature values will become missing values (Table I). Therefore, we initially split the original training set into 4 subsets. Under each subset, we remove columns that are valued -999 accordingly plus the last feature that depends on invalid columns. As a result, each dataset only contains one independent column with missing values (col. 1). For missing values occurred in this column, we apply the following imputing techniques for comparison:

- '0' filling
- by mean/median
- by linear regression
- by k-means clustering

After imputing missing values, we apply standardization by each column's mean and standard deviation: $X \leftarrow \frac{X-\mu}{\sigma}$. Before starting the training process, the property of subsets is listed below (Table II).

TABLE I
Column Analysis of Jet Numbers

PRI_jet_num	No. of Columns invalid	Position	Relevant Column
0	10 or 11	1, 5-7, 13, 24-29	30
1	7 or 8	1, 5-7, 13, 27-29	30
2	0 or 1	1	N/A
3	0 or 1	1	N/A

TABLE II
Subset Properties

Subset Jet No.	Event Count	Feature Count	Processing
0	99913	18	Std or Norm
1	77544	22	
2	50379	29	
3	22164	29	

III. Model Training

In this subsection, we will discuss our training configurations and model selection. Model performances are tested on jet-number split test subsets with the same preprocessing steps. We build and test the models developed from the following algorithms:

- Linear Regression (Stochastic Gradient Descent)(SGD)
- Least Squares
- Ridge Regression

- Logistic & Regularized Logistic Regression (SGD)

For each algorithm, we fine-tune the hyper-parameters and conduct a comparison study among different feature engineering techniques on the best algorithm. To select the best algorithm for a comparative study on feature engineering techniques, we conduct an initial model selection comparing different algorithms under the same pre-processing configuration (median imputing, standardized). For each algorithm, we select the hyper-parameter set with the least training loss. The training result is shown in Table III.

TABLE III
INITIAL MODEL SELECTION

Algorithm	Train Accuracy	Test Accuracy
Linear SGD	73.7%	75.9%
Least Squares	82.9%	65.9%
Ridge Regression	80.3%	80.3%
Logistic SGD	75.9%	75.7%
Regularized Logistic SGD	75.2%	75.1%

As ridge regression shows the best performance at first glance, we decide to use ridge regression as our optimization model. Our creative work focuses on finding the best way to impute the missing values in the first feature column. Common methods such as '0' replacing and mean/median imputing can introduce bias depending on the mechanism of feature values [4]. Also, given multiple features in this case, we prefer to impute missing values considering the relationship with other featuressince accurate imputing values will help us in classifying the events. [5] [6] offer us a plausible algorithm to realize this character through k-means clustering:

- For each normal (not -999) event, select K samples as original centroids $a = a_1, a_2, ..., a_k, k_{max} = 55$
- Label other normal events to one centroid (a_n) based on the shortest euclidean distance (m: feature numbers): $d(x, x_c) = \sqrt{\sum_{i=1}^{m}(x_i - x_{ci})^2}$
- Recalculate the position of centroids: $a_j = \frac{1}{|c_i|}\sum_{x \in c_i} x$
- Repeat last two steps until reaching maximum iteration

We also apply polynomial expansion to both sets to provide a wider feature learning background: X_n^d, with $d \in \{0, 1, ..., D\}$, where the optimal expansion degree D is determined by checking the least training loss of different degrees.

To improve our initial result, we conduct the following comparative study on processing the missing value and parameter fine-tuning before model fitting. The configurations and results are shown in Table IV, V. Median and k-means imputing have equal test performance, this implies that the first feature column does not affect much with the prediction result. However, if we apply random seed for each iteration ($i_{max} = 20$), we could reach an 0.689 F1 score. Therefore, we select the ridge regression model with standardized k-means imputing as our final submission (#164074), optimal clusters for each set: $\{85, 15, 25, 35\}$

IV. DISCUSSION

In initial model selection, the test accuracy is closely dependent on the training accuracy except for Least Squares,

TABLE IV
RIDGE REGRESSION OPTIMIZATION

Method	Test Accuracy	F1 Score
'0' Filling	79.8%	0.676
Mean	80.2%	0.684
Median	**80.3%**	**0.685**
Linear Regression	80.1%	0.682
K-means Clustering	**80.3%**	**0.685**

TABLE V
FINE-TUNED HYPER-PARAMETERS

Method	Best Degree	Best Lambda
'0' Filling	2, 6, 5, 4	1.19e-03, 1.0e-02, 1.0e-02, 5.88e-04
Mean	2, 6, 6, 5	1.19e-03, 1.0e-02, 9.43e-03, 1.19e-03
Median	2, 6, 6, 5	1.0e-03, 1.0e-03, 1.0e-06, 1.0e-03
Linear Regression	2, 6, 6, 4	4.38e-05, 9.43e-03, 1.91e-03, 4.64e-04
K-means Clustering	2, 6, 6, 5	1e-06, 1.0e-02, 2.42e-03, 7,44e-04

which has huge best polynomial expansion degrees return by the cross validation. Hence, although Least Squares achieved the best training accuracy over all the other algorithms, overfitting might be accountable for its terrible performance on testing data. We applied different approaches to replace or fit missing values in "DER_mass_MMC" and performed cross-validation for hyper-parameters tuning. Feedback from cross-validation of normalized training data suggests a much higher polynomial expansion degree than that of standardized data. In this case, as we aim to mitigate overfitting, we will only involve standardization in data pre-processing phase for the remaining experiments.

Replacing missing values using median or mean provides a higher training accuracy for small data set, e.g. Jet 3 subset. Fitting missing values using linear regression makes use of the remaining features and works well when the predicted feature has a linear relationship with other features. In reality, relationships may not exist. To overcome this problem, K-mean cluster is applied to replace missing values. Cross-validation helps to find optimal K, which is more computationally expensive. It works well when event count is huge, for example, training accuracy for Jet 0 subset is higher than median replacing, improved by 1.25%.

V. SUMMARY

From the experimental results, it can be concluded that using ridge regression with the missing value imputed using k-means clustering provides the best performance out of the other techniques. Ridge regression with missing value replaced by mean however, has nearly identical performance as k-means clustering did despite it is simple to implement and fast to converge. Thus, it is self-evident that the prediction has less dependency on "DER_mass_MMC".

Further work could involve data augmenting to learn more complex dependencies, we will also apply multivariate imputation methods to the data, such as multivariate imputation by chained equation, or apply different methods on different subsets.

REFERENCES

[1] G. Aad et al., Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC, Phys.Lett., vol. B716, pp. 129, 2012.

[2] S. Chatrchyan et al., Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC, Phys.Lett., vol. B716, pp. 3061, 2012.

[3] Claire Adam-Bourdariosa, Glen Cowanb, Cecile Germain, Isabelle Guyond, Balazs, David Rousseaua, "Learning to discover: the Higgs boson machine learning challenge", 2014-07-21

[4] S. Rosenthal, "Data Imputation", The International Encyclopedia of Communication Research Methods (eds J. Matthes, C.S. Davis and R.F. Potter). 2012.

[5] S. Wang et al., "K-Means Clustering With Incomplete Data," in IEEE Access, vol. 7, pp. 69162-69171, 2019, doi: 10.1109/ACCESS.2019.2910287.

[6] Z. Liao, X. Lu, T. Yang and H. Wang, "Missing Data Imputation: A Fuzzy K-means Clustering Algorithm over Sliding Window," 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009, pp. 133-137, doi: 10.1109/FSKD.2009.407.