Machine learning Project 1

Wei JIANG Xiaoyu LING Yao DI
EPFL, Switzerland

***Abstract*—Recently, the rapid development of machine learning algorithms makes it possible to be utilized for numerous areas and it creates significantly impacts on both the academia and industry field. The project aims at developing a machine learning method to distinguish Higgs boson from the background noise from several recorded physical parameters.**

I. Introduction

The Higgs boson is discovered at the Large Hadron Collider at CERN through the observation and analysis of the "decay signature" generated by proton collision. The repaid decay of Higgs boson and the by-product of the crash makes it hard to decide whether the given event's signature is from Higgs boson. This project aims at developing a machine learning method to distinguish Higgs boson and background through the event's signature.

The report presents the our method developing processes including data prepossessing, feature scaling, algorithm selection and parameter tuning in Sec II. A discussion based on the project result (Sec III) and conclusion (Sec IV) are given at the end of the report.

II. Models and Methods

A. Data Preprocessing

Data pre-processing especially data cleansing is a necessary step to minimize the prominent errors caused by the incomplete and inaccurate records. The following actions were taken according to our investigation on the raw data set.

1) Data Regrouping: The raw dataset we have includes a training set of 250,000 events and a testing set of 568,238 events, where all the null values are record as -999. Among all 30 features, **PRI_jet_num** is an important categorical parameter. There is a strong correlation between the value of this integral feature and the null values in several specific features measuring the jet, which is shown in Table I. The raw data set could be divided into three groups accordingly.

2) Group-based Null Feature Processing: In the regrouped data-set, some features would contain only null values. Those values are meaningless and will lead to zero standard deviation in standardization process. Thus, we could minimize the impact from the incomplete data and reduce the calculation complexity by removing the null features from the first two groups according to Table I.

Meanwhile, in the first two groups, **PRI_jet_num** becomes a constant feature, which will be removed in order to improve the calculation efficiency. As a biasing feature is added in the feature scaling later, this removal will cause no information loss.

PRI_jet_num	0	1	other
DER_deltaeta_jet_jet	null	null	normal
DER_mass_jet_jet	null	null	normal
DER_prodeta_jet_jet	null	null	normal
DER_lep_eta_centrality	null	null	normal
PRI_jet_leading_pt	null	normal	normal
PRI_jet_leading_eta	null	normal	normal
PRI_jet_leading_phi	null	normal	normal
PRI_jet_subleading_pt	null	null	normal
PRI_jet_subleading_eta	null	null	normal
PRI_jet_subleading_phi	null	null	normal
PRI_jet_all_pt	null	null	normal

TABLE I
Correlation between the feature textbfPRI_jet_num and several other features. Null means all the feature values are equal to -999

3) Outlier Processing: The initial dataset also contains several outliers. The outliers are identified with interquartile range method as we are not sure whether the physical parameters follows Gaussian distribution. IQR is calculated as the difference between the 25th and the 75th percentile of the data, and the outlier is identified when it is 1.5 times of IQR below the 25th percentile or above the 75th percentile.

We tried to process the detected outliers in the following ways:

1) use mean of non-outliers to replace outliers
2) use median of non-outliers to replace outliers
3) use closest non-outlier to replace outliers

The trial results show that the third method leads to the best prediction accuracy.

It is noticed that there still exits some null value in feature **DER_mass_MMC**. Those null values are processed as outliers. A new feature is added later to record the removal of the null values to avoid information loss.

4) Standardization of initial values: For subsequent feature extension processes, such as logarithm and exponent, it is a beneficial step to standardize the data, i.e. subtract the mean and divide by the standard deviation for each feature. For the test set, it is subtracted and divided by the value obtained from train set.

B. Feature Scaling

Since the real physical parameters are non-linearly correlated, the feature scaling step is necessary. The 6 types of external features are introduced in addition to the initial features:

- *Bias.* $[xT_{bias}]_n = 1$
- ***DER_mass_MMC** null value indicator.*
  $[xT_{null}]_n = \{1|xT_{n,0} = -999; 0|otherwise\}$

- *Polynomial (argument)*:
  $[xT_{poly}]_n = [xT_n^2, xT_n^3, ..., xT_n^{degree}]$ [1]
- *Sine and cosine.*
  $[xT_{sin}]_n = [\sin(xT_n)]$ $[xT_{cos}]_n = [\cos(xT_n)]$
- *Exponent and logarithm.*
  $[xT_{exp}]_n = [\exp(xT_n)]$ $[xT_{log}]_n = [\log(|xT_n|)]$
- *Cross multiply.*
  $[xT_{multi}]_n = [xT_{n,i} * xT_{n,j}](i < j)$

The last 4 types of added features are standardized before attached to the data from the previous procedure.

C. Algorithms

1) Algorithms Comparison: We compare the accuracy of six methods after feature extensions using the simplest dataset, i,e, the set with PRI_jet_num = 0. The dataset is split into two sets: 80% for model training and 20% for the accuracy test. The prediction accuracy with six implemented functions could be found in the table below.

Methods	GD	SGD	LS	RR	LR	RLR
Accuracy(%)	84.85	84.40	85.23	85.24	84.65	84.66
Degree	5	5	5	5	5	5
γ	5e-03	5e-04	/	/	1e-05	1e-05
Max_iters	1500	3000	/	/	3000	3000
λ	/	/	/	5e-07	/	/

TABLE II
PREDICTION ACCURACY WITH DIFFERENT METHOD

The result is similar and great. All methods have accuracy above 84%. When implementing logistic regression and regularized logistic regression, we change the label [-1,1] to [0,1] first and then change back to get accuracy. But considering their low speed for large dataset as we have, we decide not to choose these two methods.

The Least Square Gradient Decent and Least Square Stochastic Gradient Decent have the same problem. They always have the problem of choosing γ and it works slowly. And there is no penalty for the high polynomials so it may cause over-fitting where the Least Square method has the same problem.

We can see the Ridge Regression has the highest accuracy. It is not accidental if we analyse it. The ridge regression has the all advantage of Least square but also with additional penalty for high polynomials thus avoid over-fitting. Thus we choose this as our model. We still need to decide the hyper parameters, i.e. the polynomial degree, the λ. Cross validation is a good way to select best hyper parameters as it covers whole training set.

2) Parameter Tuning with Cross Validation: We split the dataset into four folders: three of them for training and the rest one for validation. We choose to test some sets of λ and degrees to find out some good models.

a. The ridge factor of λ with degree = [5, 8, 11]:
    Lambdas = np.logspace(-10,-2,80)

b. The degree of polynomial base with λ = [1e-10, 1e-7, 1e-3]:
    Degrees = np.arange(2,22)

we take the test loss(mean square error) as the index to evaluate the combination of hyper parameters. The chart shows some good models we get.

Combination	#comb 1	#comb 2	#comb 3	#comb 4
Degree	5	7	8	11
λ	1e-7	1e-3	1e-10	1e-07
Test loss(mse)	0.681	0.682	0.6798	0.6801

TABLE III
TEST LOSS WITH DIFFERENT HYPER PARAMETERS

We put two example plots(Appendix figure 1 with degree = 8 and figure 2 with λ = 1e-7) showed in the appendix for discussion. Generally, we get a higher accuracy through the increased flexibility of our model. However, the plot is not smooth with some peaks where overfitting can be a reason since the peaks always appear when λ is extremely low or degree is extremely large.

D. Final Model

The diagram concludes the previous methodology sections and describe the process to generate final classification results could be found in the Appendix (Fig 3).

III. Results Discussion

With the methodology described previously, we could generate a model with could achieve 83.4% accuracy in the classification prediction of the Higgs Boson based on the test set. It would be possible to improve the accuracy in the following ways.

- *Increase the computing power:* It is common to have a complex non-linear relation among the different physical parameters. However, for lack of the computation power, we could only do feature scaling with limited order and basic transformations. With a higher computation model, we could test a more adequate feature set. It is important to keep cross validation in order to avoid over-fitting.
- *Improve the theoretical knowledge:* Domain-specific knowledge could help to build a better feature set without sacrificing the computation efficiency. Given a more detailed knowledge about the physic background of Higgs Boson, we could remove or cut down the weight of some unrelated features (e.g. sinusoidal transformation may only be reasonable for angular features) and find a more reasonable feature cross-multiple combination.

IV. Conclusion

During this project, we managed to use the machine learning algorithm learnt in the course to approach the binary classification problem of Higgs Boson with a reasonable accuracy. We also learnt that a good machine learning model could not be constructed without proper data-prepossessing, sufficient feature scaling, suitable regression algorithm and fine parameter tuning.

[1] **xT** represents the original feature

Appendix A
Cross validation results

The results of cross validation in parameter tuning process.

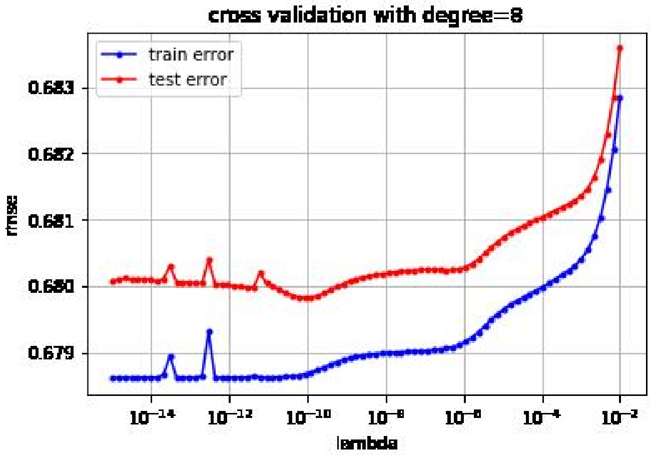

Fig. 1. Cross validation with degree = 8

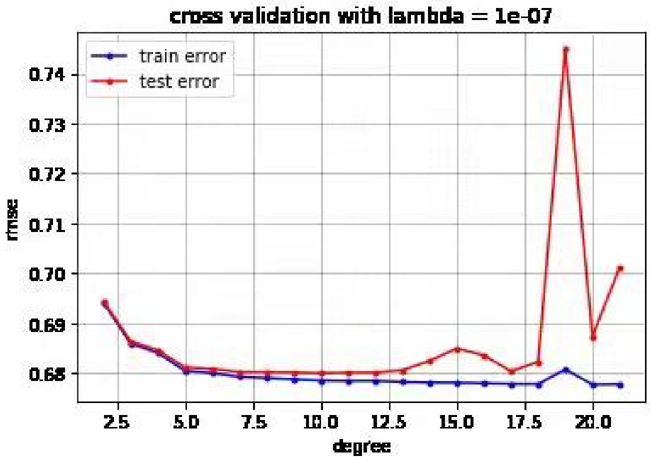

Fig. 2. Cross validation with lambda = 1e-7.

Appendix B
Flow chart for final program

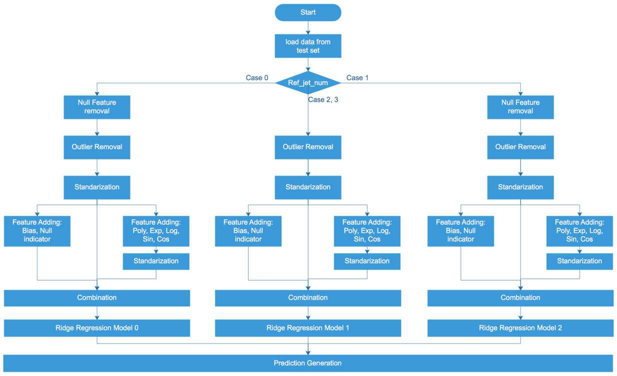

Fig. 3. Flow chart for prediction generate process.