CS-433 Machine Learning - Project 1

Yan Fu[a], Shengzhao Xia[a], Wan-Tzu Huang[b]
[a]*The Institute of Electrical Engineering, EPFL Lausanne, Switzerland*
[b]*Faculty of Computer Science and Communication System, EPFL Lausanne, Switzerland*

I. Introduction

In March 2013, the discovery of Higgs boson particles was announced at the Large Hadron Collider at CERN. The production of a Higgs particle is very rare and when it happens, scientists cannot observe it directly as it decays rapidly into other particles. Therefore, Higgs boson particles' decay signature is measured instead as an alternative way to distinguish whether the particle is a Higgs boson particle or not.

In this report, we present our first Machine Learning project and show how we used binary classification techniques to estimate the likelihood of a given decay signature is the result of a Higgs boson or background noise.

II. Feature Selection and Data Processing

The data set used in this project consists 250000 events for training and 568238 events for testing. The first column is the ID column, then followed by the label column which indicates whether the signal is generated by Higgs Boson particle or from the background noise, and finally 30 feature columns. In each event, if feature cannot be computed or is meaningless, their values are set to -999.0.

A. Feature Selection

1) **Features deleting with Phi variable**
We plotted the histogram of 30 features to learn their characteristics generally. We found that the 5 features with phi variable had uniform distribution. And these features could be compared to white noise, which did not provide useful information. So we dropped out these 5 phi features.

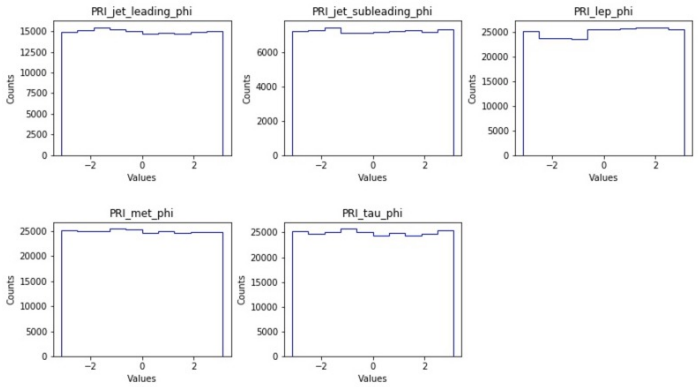

Fig. 1: Density plots of 5 phi features

2) **Data set grouping by PRI_jet_num**
According to the Higgs Boson document, features would be defined or not defined depending on the value of PRI_jet_num[1]. When jet number equaled to 0 or 1, 10 or 7 features would be undefined respectively. And all given features were defined when jet number was 2 or 3. Therefore we split the data set into 3 groups with jet number 0, jet number 1 and jet number 2 or 3 respectively. After then we dropped undefined features, features with all 0s and also PRI_jet_num.
3) **Linear correlated features**
Under the assumption that strong linear-related features would be redundant, we hoped dropping linear correlated features would help improve model performance. However in Fig.2 when degree grew, although the test accuracy also improved, there was a gap between model dropping correlated features and model that did not drop correlated features. The reason may be that after polynomial, some correlated features became non-linear related and may have new physical meanings. And dropping them would lose polynomial information. In the end, we decided to preserve these correlation features.

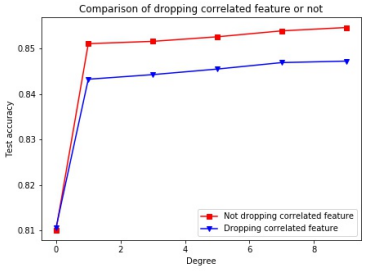

Fig. 2: Comparison of dropping correlated features or not

B. Data Processing

1) **Undefined data replacement**
We noticed that other than the 10 undefined features related to PRI_jet_num, there were still some missing values in the columns DER_mass_MMC. To fix these missing value, we calculated the mode of DER_mass_MMC in background or in signal and replaced -999 with mode respectively.
2) **Logarithm of data**
We took logarithm of positive data for each feature column to narrow down the data range.

3) **Standardization**
Both training data set and test data set were standardized by subtracting the mean and dividing their standard deviations.
4) **Polynomial features building**
We built polynomial features for each group by squaring, root-squaring and multiplying every two feature .

III. Model

A. Model Selection

The performance of different models were compared and the model with best accuracy was chosen to do further parameter tuning. 10-fold cross validation was used to validate the performance of each model.

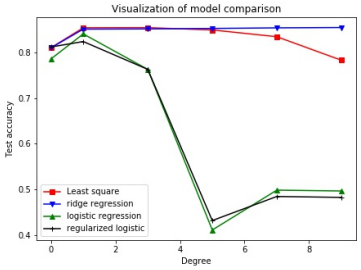

Fig. 3: Model comparison

Least square performed very well but, when we applied least square on test.csv and uploaded the result to Kaggle, the score was very low with only 0.51944 , implying high over-fitting. Ridge regression improved least square by adding lambda as over-fitting penalty and we could see that when lambda = 0.001, ridge regression also has a good performance, and on Kaggle we got a score on 0.82236 for this model, showing that ridge regression balanced well between improving model accuracy and avoiding over-fitting. And theoretically, logistic regression should be the best solution for binary classification problem. And here we saw it did have good performance when polynomial degree = 1, and the Kaggle score for logistic regression is 0.81866, though a bit lower than ridge regression, which may attribute to inadequate iteration times. And performance of logistics dropped dramatically when degree grew, mainly because we did offer enough iterations. Taking the computing ability of our computer into account, finally we decided to use ridge regression.

B. Hyper-parameters Tuning

Hyper parameter tuning on polynomial degree and lambda were performed for ridge regression using grid search. We set the degrees from 1 to 14, and lambdas = 0.0005, 0.001, 0.01 and 0.1, then we ran ridge regression exhaustively using every possible combination of degree and lambdas. In fig 3 we could see how the accuracy on test set changes according to the change of degree and lambda. And the best combination was lambda = 0.0005, degree = [9, 9, 12] and the test accuracy was 0.855, 0.820 and 0.851 respectively for each group.

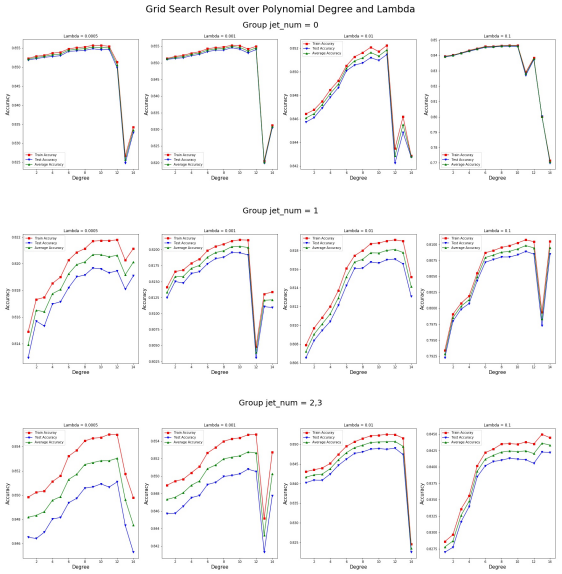

Fig. 4: Grid search result over polynomial degree and lambda

IV. Discussion

In this project, the most annoying thing is the poor computation ability of our laptops. To solve this problem, on the one hand we can set up and buy a cloud service machine (like Amazon AWS); on the other hand we can consider further feature selection. For example, after 14-degree polynomial feature building, we get a new data set with over 1400 features. If we can know more physical relations between features we can select polynomial features more selectively, hence reducing data set dimension and computation time.

V. Conclusion

During this project, the feature selection and data processing played an important role in improving the prediction accuracy. By using ridge regression and the help of turning hyper-parameters, we achieved kaggle score of 0.82769 with lambda = 0.0005 and Degrees = [9, 9, 12].

References

[1] Adam-Bourdarios C, Cowan G, Guyon I, et al. The Higgs boson machine learning challenge[C] International Conference on High-Energy Physics and Machine Learning. JMLR.org, 2014:19-55.

[2] Pontus Giselsson and Stephen Boyd. Monotonicity and restart in fast gradient methods. IEEE 53rd Ann. Conf. Decision and Control, pages 50585063, 2014.

[3] Bishop, Christopher M, and N. M. Nasrabadi. Pattern Recognition and Machine Learning. Pattern recognition and machine learning. Springer, 2006:461 - 462.

[4] Boyd, Vandenberghe, and Faybusovich. "Convex Optimization." IEEE Transactions on Automatic Control 51.11(2006):1859-1859.