Predicting the signal of Higgs boson using machine learning techniques

Saibo Geng, David Yeung, Sahil Shah

***Abstract*—The Higgs Boson is an elementary particle in the Standard Model of particle physics to explain a fundamental question in physics: why particles have mass. In 2013, European Organization for Nuclear Research (CERN) announced their discovery of the Higgs boson in a particle collision. A further research topic related is the characterization of the occurrence of Higgs boson in a collision event. This problem is difficult because the Higgs boson decays to other particles rapidly and the products of this decay process (decay signature) may look similar to one another. In this paper, classifiers using multiple machine learning mechanism is build, aiming to simulate the discovery of the Higgs boson.**

I. Introduction

From the machine learning perspective, this problem can be formally transformed into a binary classification problem given a number of features. The output is of two categories: **s** and **b**, representing **signal** and **background**. The data set consists of roughly 800,000 data points provided by the official *CERN* particle accelerator data, divided into training (250,000 events) and testing (56,8238 events) sets. Each data point in the training data contains a label and the id together with 30 features describing the physics quantities in the collision event that they represent. A set of regression methods are used in this challenge, including linear, ridge and logistic regression.

II. Exploratory Data Analysis

The data contains of 30 different features each representing a different physics description of the collision event. There exist features which contains values -999, which is outside of the normal value range of all the fields. These values represent variables that are meaningless or cannot be computed for some events. From the analysis, there are 11 fields containing this value and some fields have up to 177457 (around 70% of the training data set) points with value of -999. These values will be transformed in the later feature generation step.

III. Feature Cleaning

A. Baseline

In order to evaluate the effect of data cleaning and subsequent feature augmentation, the baseline model will be the result generated from least square regression with normal equation with raw data. 4-fold cross validation will be used in the following to compared the effect of different feature generation method. Each step below will be cumulative.
Baseline Test Set Accuracy (teAcc): 74.419%

B. Abnormality Removal

As mentioned, the value -999 are invalid entries in the data sets. There are 7 columns with more than 70% of values being -999. It is reasonable to remove these columns to keep only the more meaningful features.

Also, there are entries containing the value -999 but the proportion is not as significant. Outliers (defined here as values outside the range of $25th\ \ percentile - 1.5 * interquartile\ \ range$ to $75th\ \ percentile + 1.5 * interquartile\ range$) also exist in each field. These values are being replaced by the mean of the column.
Post-column Removal teAcc: 74.616%

C. Normalization

Since the input data contain values in different ranges, it is reasonable to re-scale the values with normalization to have a better comparison among data points(especially when the unity system is not necessary aligned). The normalization method being used will be standard score (z-score). We also test the min-max normalization, but it performs not as well as z-score.
Post-normalization teAcc: 74.987%

IV. Multiple Machine Learning Approach

A total of 6 regression mechanism is used, namely *linear gradient descent*, *linear stochastic gradient descent*, *least square with normal equations*, *ridge regression using normal equations*, *logistic gradient descent* and *regularized logistic gradient descent*. The results in figure 1 below are obtained with 4-fold cross validation (additional configuration is specified). Only the feature cleaning part is carried out and the feature augmentation is not applied unless otherwise specified.

(Refer to Fig. 1) The results indicates that the polynomial regression with degree 8 gives the best performance in terms of accuracy. The time used for a 4-fold cross validation test is also reasonably small using ridge regression with normal equations compared with other training method, e.g. gradient descent).

A. Choice of step-size (gamma) and degree

The step size of gradient descent (GD) and stochastic gradient descent (SGD) was first computed using: $1/Lipschitz\ C\ onstant$, computed from the Hessian. However, the step size is too small for reaching convergence. Therefore, it is increased until the largest value before the test (GD or SGD) diverged. The degree search space ranges from 1 to 8, in polynomial basis. For ridge regression (RR) and polynomial regression, each degree value in the space is computed and the results shown is the one with maximum accuracy.

Method	Testing Accuracy	Time
Least Squares (GD)	74.23%	106.4s (iter=3000, gamma= $2.195e-7$)
Least Squares (SGD)	65.69%	118s (iter=1000, gamma= $2.195e-9$)
Least Squares normal equation	74.99%	0.55s
Ridge Regression (nor eq, polynomial basis)	81.0276%	7.79s (degree=5, lambda=$10e-6$)
Logistic Regression (GD)	78.48%	396s (iter=4000, threshold=$1e-5$, step-size=$1e-8$, degree=6)
Regularized logistic Regression (GD)	77.80%	278s (iter=2000, lambda=$1e-4$, threshold=$1e-5$, step-size=$1e-8$, degree=6)
Least Square (nor eq, polynomial basis) feature grouping	82.80%	7.82s (degree=5)

Fig. 1. Accuracy and time obtained by different regression methods

B. *Choice of lambda*

The lambda item in ridge regression determines the penalisation for complex models. The lambda used is determine by a ridge regression loss analysis with degree 1 and root mean square loss.

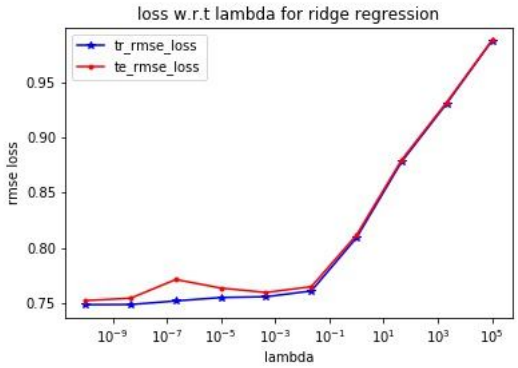

Fig. 2. RMSE loss under different lambda in ridge regression with degree 1

V. Under / Over - fitting analysis

After data cleaning, there are only 23 fields remaining, which may not be able to fully capture the underlying distribution, hence under-fitting. Polynomial feature augmentation can be used to increase the dimension of the data points. However, the problem of over-fitting may arise. Therefore, another 4-fold cross validation analysis will be done to compare the effect of different dimension.

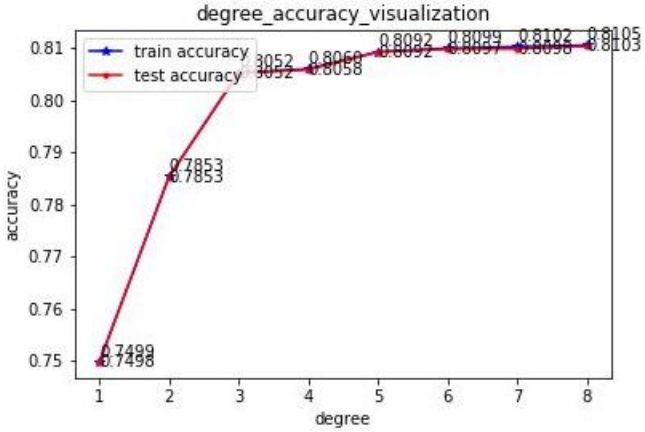

Fig. 3. accuracy vs degree of augmentation

From the graph, no over-fitting behavior can be observed , but while comparing the accuracy calculated by AI crowd, an evident over-fitting arises from degree 5. Hence the degree was set to be 5, having $5\times23features$.

VI. Feature Grouping

A close observation on the data again reveals a special column, namely "PRI_jet_num", which takes on only 4 different values, i.e. 0, 1, 2 and 3. A grouping on this column can be done, where 4 different models are build corresponding to each group. A cross validation analysis is done with degree 6 polynomial feature augmentation with grouping. The results is as follows. **Training Set Accuracy**: 83.15%, **Testing Set Accuracy**: 82.80%. The result shows a increase in performance in the overall classification and will be our final submission model.

VII. Discussion

This project can be seen as 3 principal steps:data analysis and treatment, implementation of various machine learning algorithms and finally tuning the hyper-parameters of each models. The first step helped us to gain a slight improvement of prediction accuracy (+0.8%). On the second step, we saw that the accuracy of each algorithm along with their runtime were distinct. In our case, least squares using normal equation with polynomial basis (degree 8)performs best.Not only does it give the best testing accuracy but also it takes least training time (due to no iteration fact). To our surprise, logistic regression which is generally used in binary classification tasks only gives the 3rd best accuracy. We think this is due to that the gradient descent didn't minimize the log loss to the limit. In the last step, we discovered that a polynomial basis could give a boost to the accuracy of our model, but the this boost attenuated when the degree went up, and the complexity of calculations grew fast as well. With feature grouping, our final highest accuracy is **82.07%**.