The Machine Learning Higgs Boson Challenge

Allan Bellahsene, Lionel Brodard, Antoine Marchal
CS-433 Machine Learning, EPFL, Switzerland

***Abstract*—The goal of this paper is to study the characteristics of a well-known physics phenomenon whose existence was experimentally confirmed in 2012: the Higgs Boson. In this article, we try to fit a model that can predict the presence of a *signal* using machine learning methods. No physics theory is used.**

I. Introduction

Predicting the presence of a *signal* or a *background* is a classification (binary) problem, as the output variable can only take two discrete values. Hence, the theory would suggest classification methods are more suited for this problem that standard regression models. However, we show that ridge regression using polynomial basis function produces an acceptably high accuracy rate.

II. The Data

A. Train and Test Data

The data used in this project is an original dataset from CERN. As is any machine learning project, the data is divided into train and test data. Train data is used to construct our model while the test data is used to simulate how our model would perform on unknown/future data. Hence, unless stated otherwise, every variable mentioned in this project will refer to the train data set, as the test data is only used at the end to estimate the performance of our models.

B. Dimensions

The label, $Y = (Y_1, ..., Y_N)^T$, is an $N \times 1$ vector, where N = 250,000 is the number of observations. Each Y_i can only take two values, *b* or *s*, respectively *background* and *signal*. The feature matrix, $X = (X_1, ..., X_D)$, is the matrix of independent variables (or features) where D = 30 and $X_K = (X_{K,1}, ..., X_{K,D})^T$, for $K \in \{1, ..., 30\}$.

C. Exploratory Data Analysis

The missing values, replaced by -999 in the original dataset, represent 21.06% of the values. After a more in-depth analysis, we observed that 11/30 features present missing values. After going through the documentation about the challenge and in particular through the detailed description of the features [1], we conclude that these missing values arise from different causes. For this reason, we distinguished two categories of missing values and treated them differently.

DER_mass_MMC_NaN

Prediction	PRI_jet_num	
b	0	24564.0
	1	6857.0
	2	2485.0
	3	1373.0
s	0	1559.0
	1	705.0
	2	467.0
	3	104.0

Fig. 1. Detailled repartition of NaN in the first feature.

1) Undefined values of the 'DER mass MMC' feature: The *DER mass MMC* feature presents undefined values when the topology of the event is too far from the expected topology. The first inspection of this feature revealed that in 92% of the cases, a missing value in the *DER mass MMC* feature corresponds to a background event (see Fig.1). Therefore, we found judicious to replace these missing values by the average mass of the background events (This is realised by the *'data_preprocessing1'* function that we implemented in the *implementations.py* file.).

2) Undefined values in the features 5-7, 13 and 24-29: The number of undefined values in these features depends on the number of jets. Furthermore, this dependence is similar for some features. Hence the missing values will simultaneously occur in different columns for a given number of jets. This motivates us to separate the data set according to the number of jet events. We distinguished three categories:

- First category: number of jet$= 0$
- Second category: number of jet$= 1$
- Third category: number of jet≥ 2

This separation resulted in three data sets S_1, S_2 and S_3, with data sizes $N_1 = 99913$, $N_2 = 77544$ and $N_3 = 72543$, corresponding respectively to the three categories above. As we can deduce from Fig.2, after the separation, the features columns either consist only of missing values or have simply no missing values at all. Therefore, the processing of the *NaNs* (Not a Number) becomes much simpler, as we delete the columns made only of *NaNs*. The separation, according to these three categories, also engenders some columns only made of the same constant, we delete them as well. Indeed, they will become redundant knowing that we will introduce a constant column (column of ones) when using the polynomial basis expansion.

PRI_jet_num	DER_deltaeta_jet_jet_NaN	DER_mass_jet_jet_NaN	DER_prodeta_jet_jet_NaN	DER_lep_eta_centrality_NaN	PRI_jet_leading_pt_NaN	PRI_jet_leading_eta_NaN	PRI_jet_leading_phi_NaN	PRI_jet_subleading_pt_NaN	PRI_jet_subleading_eta_NaN	PRI_jet_subleading_phi_NaN
0	99913.0	99913.0	99913.0	99913.0	99913.0	99913.0	99913.0	99913.0	99913.0	99913.0
1	77544.0	77544.0	77544.0	77544.0	0.0	0.0	0.0	77544.0	77544.0	77544.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 2. Number of NaN by jet value

III. The Model

A. Motivation

We decided to fit a ridge regression using a polynomial basis function. Using an augmented feature allows us to catch more complexity in our model than a simple linear model, but this could also lead to over-fitting which could in turn lead to multicollinearity. Furthermore, the fact that some features are derived (the ones with a *'Der'* in their names) may also be a possible cause of multicollinearity. Hence, this is mainly what motivated us to use Ridge regression instead of Ordinary Least Squares regression, as it penalizes models with large weights and performs better in presence of multicollinearity. The optimal weights matrix is given by:

$$w = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

B. Optimal parameters

Hence, two parameters have to be optimally found: the number of degrees M of the polynomial basis function and the optimal λ of the ridge regression. Our goal is therefore to find an optimal pair (λ_i, M_i), for each subset S_1, S_2 and S_3 (respectively shown in Fig.3, Fig.4 and Fig.5) defined above, that would maximize the accuracy of our predictions. Accuracy in this case refers to the accuracy of a subset of each S_i that we used as a test data. Once we got the optimal parameters for each category, we used them to make our predictions.

IV. Results

The best submission on AIcrowd is referenced as number 23620. It gives an accuracy of 82.4%.

V. Discussion

There are ways to improve the accuracy of our model. First of all, we could have removed outliers. Our objective function is very responsive to outliers, so removing them could have improved the prediction. We tried but did not have enough time to finish it. Also, we could have used more time to find the optimal parameters. Finally, as mentioned in the introduction, Ridge regression was used over Logistic Regression which in theory is better suited for classification problems. We would have implemented it first by setting the -1 values of the label to 0 values, but we decided to focus on an algorithm that we managed better and which gave us a reasonable prediction rate of 82.4%.

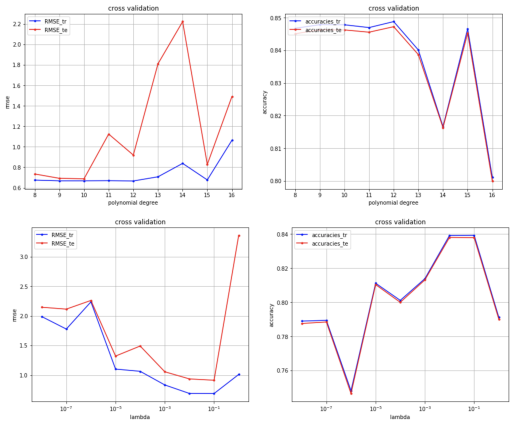

Fig. 3. Optimal number of degrees and optimal lambda for jet $= 0$

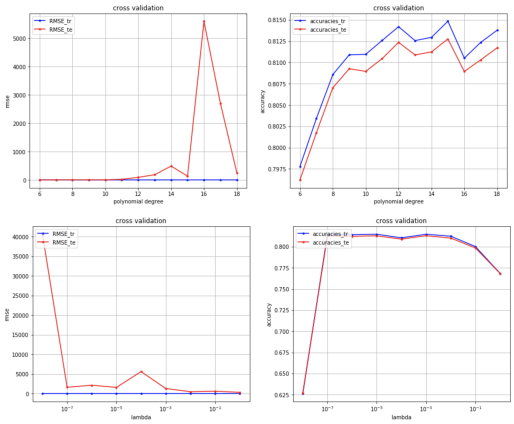

Fig. 4. Optimal number of degrees and optimal lambda for jet $= 1$

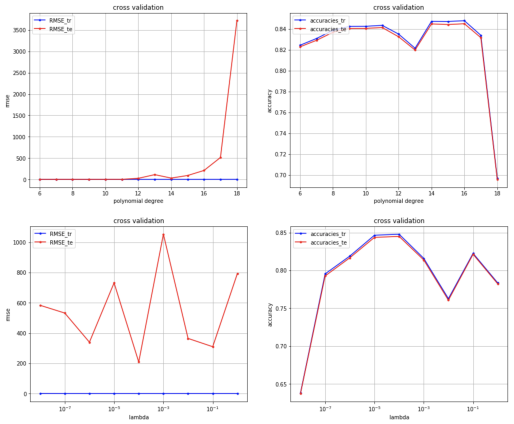

Fig. 5. Optimal number of degrees and optimal lambda for jet≥ 2

References

[1] The higgs boson machine learning challenge. In *Proceedings of the 2014 International Conference on High-Energy Physics and Machine Learning - Volume 42*, HEPML'14, pages 19–55. JMLR.org, 2014.