Project report: Higgs Boson Challenge

Riccardo Cadei, Raphaël Attias, Shasha Jiang
Department of Computer Science, EPFL, Switzerland

***Abstract*—In this report we present our approach to solve Higgs Boson Machine Learning Challenge: a classification problem on a huge dataset simulated by the ATLAS experiment from CERN.**

I. Introduction

The task of the Challenge was to predict whether a collision event can generate a Higgs boson, an elementary particle in the Standard Model of physics, or not. In fact, scientists cannot really observe this particle, but only its decay signature through different measurements. We analyzed, pre-processed, transformed those decay signatures and use them as features in our Machine Learning algorithms. In Section 2 we present our analysis and elaboration of the dataset. Furthermore we present the 6 models that we built and how we selected the best one with the best parameters combining Grid Search and Cross Validation. In Section 3 we report the results and a final discussion is in Section 4.

II. Models and Methods

We have studied and solved this classification problem splitting it in two sub-task: data analysis and algorithmic design. First, we have studied the data set and the features provided, which consisted in cleaning the sample set and proceeding in an explanatory analysis. Second, we focused on multiple algorithmic and numerical methods to create a good classifier and set its best parameters.

A. Data Analysis and Exploration

The training set consists of 250'000 data points with 30 features and their corresponding labels (-1 for "background" and 1 for "signal"). The test set consists of 568'238 data points with the same 30 features and we have to predict their labels.

1) Categorical feature: We noticed that all variables are continuous, except *'PRI Jet num'* which is categorical. From the challenge documentation we understood that some features only made sense for a specific number of Jets. Hence we decided to split our data set into three distinct classes (subsets) having *'PRI Jet num'* respectively equal to 0, 1 and 2 or 3.

2) Missing Values: The dataset is full of missing values. We managed the majority of them by deleting, for each class, the features containing only missing values. We imputed the remaining missing values with the median, a robust estimator for outliers. Then we plotted the empirical features distribution per each Jet class.

3) Outliers: First of all we observed that there are several outliers, we plotted them and we capped the extreme values of the elements of each feature to a minimum (maximum) α-percentile (1-α). Then, once we built our best model we found the best parameter α per each class, maximizing the accuracy predicted through Cross Validation.

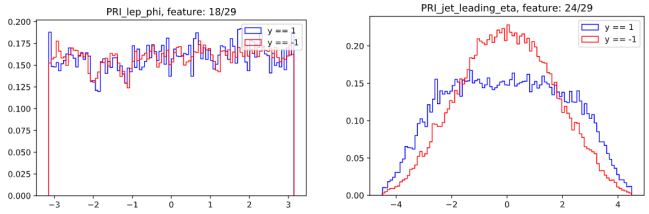

Figure 1: Empirical distribution plots of the features 18 and 24 for Jet class 0, with respect to the label of the samples.

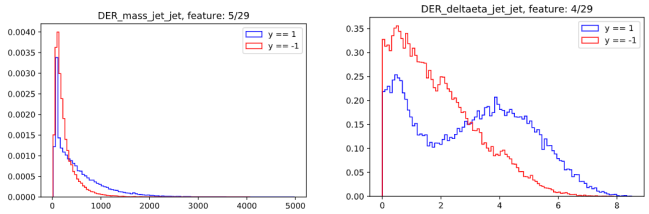

Figure 2: Empirical distribution plots of features 5 and 4 for Jet class 0, with respect to the label of the sample.

4) Feature Engineering: Distributions plots are also important for feature selection and transformations. For each feature of the train set we compared the empirical distributions given different labels. We deleted the features [15,16,18,20] because the empirical distributions for different labels plots were too similar (see an example in Figure 1A). We computed the absolute value of the features [14,17,24,27] with symmetric distribution with respect to zero (see an example in Figure 1B) and we compute the $log(1+x)$ transformation to the positive features with a long tail to the right (see an example in Figure 2A). Figure 2B is an example of a good features i.e. the distribution of this feature is highly dependent on the value of the label.

5) Standardization: A good practice in Machine Learning is to standardize the dataset, in particular when you use numerical methods for optimizations (since the features have different scales, the features matrix could be ill-conditioned)

6) Balancing: The dataset is quite unbalanced (66%-34%) but not excessively. Under and Over Sampling algorithms like Random Over Sampling did not seem useful to improve the accuracy.

7) Polynomial Expansion: Finally, for each 'class Jet', we added an intercept (constant column) to the feature matrix, we added the polynomial expansion until the appropriate 'degree' for each feature, we added the root and the cubic square of the feature matrix element per element, and for each couple of features we added a new feature with their product. We tuned 'degree' by maximizing the accuracy predicted through cross validation.

Method	Train Accuracy	Test Accuracy	Variance on Test
Least Squares GD	0.785777	0.785727	$2 \cdot 10^{-6}$
Least Squares SGD	0.737212	0.737296	$8 \cdot 10^{-6}$
Least Squares	0.843021	0.840316	$2 \cdot 10^{-6}$
Ridge Regression	0.842766	0.840568	$2 \cdot 10^{-6}$
Logistic Regression SGD	0.720633	0.720975	$9 \cdot 10^{-5}$
Reg Logistic Regression SGD	0.720631	0.720975	$9 \cdot 10^{-5}$

Table I: Training and test accuracy obtained when performing 3-fold cross-validation on the training set.

B. Methods

Once we analysed and cleaned the dataset we moved to the models. We focused our classification on linear models and logistic regression. Both models can be approximately solved using numerical methods such as Gradient Descent (GD), or the less computationally expensive Stochastic Gradient Descent (SGD). To minimize overfitting on the training set, we tried a variant of linear and logistic problems by adding a regularization term λ. Among the different solutions to linear models, with or without regularization term, we preferred the analytical solutions rather than numerical methods that could be computationally expensive and diverge with such a large dataset.

After implementing the aforementioned methods, the difficulty of this challenge was to figure out the best method on which to focus on. We used two metrics to measure the quality of our models, the accuracy and F1-Score. To have a good estimate of those metrics, we used cross-validation on the training set. Finally we used grid search on a range of hyper-parameters (λ) of the method and the parameters of the pre-processing (α) and polynomial expansion (degree) to maximize the accuracy.

Note that since we want to counter the overfitting on the training set, during our cross-validation estimate of the test accuracy we aim to also minimize the variance of the test accuracy. High variance on the test accuracy would indicate that our model may note generalize enough.

III. Results

A. First analysis of the 6 methods

First we needed to choose among all the 6 methods described which one we should focus on. From Table I, all methods perform with an accuracy on the test set greater than 0.72. We also observe that the best training accuracy is achieved by *Least Squares*, while the best test accuracy is reached by *Ridge Regression*. The suspect is that Least Squares is overfitting, and since we want to maximize the test accuracy, we focused on Ridge Regression. Ridge Regression is also less computationally expensive than the iterative numerical methods.

B. Tuning parameters for Ridge Regression

The linear model with a regularization term (or *Ridge Regression*) was performing with the best accuracy in our first trials. So we decided to focus on this model for the second step of this report. We used grid search to find, for each Jet class, the hyper-parameter of our model (the regularization term λ) and the appropriate parameters for our features expansion (the degree d and the α-quantile).

Each jet category would require different parameters, from Figure 3 we observe that for jet category 0, the parameters $\alpha = 4$, $d = 5$, $\lambda = 10^{-6}$ generates the best test accuracy. Proceeding on the other jet categories, we obtain 3 set of parameters that produces a mean accuracy on the test set of 0.841 (5th-best acccuracy on the public ranking of AIcrowd).

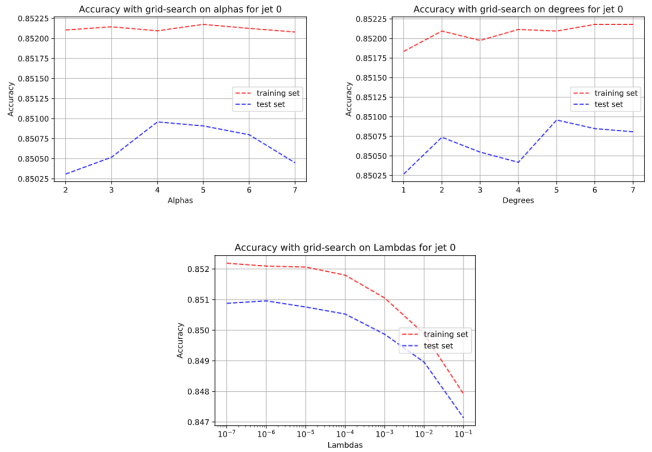

Figure 3: Results of grid-search on the parameters of the feature expansion and Ridge Regression, for Jet category 0

IV. Discussion

As we have seen from the results of section III-A, a Ridge Regression based model was best suited for our task. This model has an analytical solution that can be computer almost instantaneously, and the regularisation term allows us to counter overfitting. This last point is particularly important considering the scale of the test set compared to the training set. Creating a model for different Jet category was also a strength of our analysis, since it was the most reasonable way to deal with the missing values. From this observation, the final model was more finely tuned to each of the Jet category. However, we still observed a disparity in the test accuracy between the Jet categories due to their different balancing. The perfect accordance between the accuracy predicted on the training set through cross validation and the accuracy that we got on the test prediction submitted on AI Crowd keeps us away from thinking we are overfitting.

V. Conclusion and prospects

This report emphasizes the rule of the Data Analysis, Preprocessing and Feature Engineering in ML systems. We have shown that even a very-standard classification algorithms, with the right interpretation and manipulation of the data, can perform well on a specific task.