Machine Learning Higgs Boson Challenge

Adam Elodie, Des Courtils Philippine, Eyraud Pauline

***Abstract*—The Higgs boson is an elementary particle whose existence was announced in 1964 and observed for the first time in 2013 at the Large Hadron Collider of CERN. During a high speed collision event between billions of particles, it is possible to identify the Higg boson thanks to its decay signature. Data of a collision event can be collected and used as a training set for machine learning models. These models enable us to predict whether a Higg boson was emitted or not during a collision, based on the recorded parameters. This paper will explore various machine learning optimization models and assess their performance in predicting the presence of a Higgs Boson. The best model was the one using ridge regression, since it showed the best overall accuracy.**

I. Introduction

The Higgs boson is considered by physicists as the keystone of the matter fundamental structure. This particle gives mass to all the other particles in our universe. However, the Higgs boson is very hard to observe directly. Indeed, it only appears for a fraction of a second in very high speed collisions between particles. Luckily, the boson can be indirectly identified via its decay signature.

The CERN announced in 2013 the discovery of Higs Boson at the Large Hadron Collider (LHC), almost 50 years after its existence was theorized in 1964. Since a collision event creates many similar decay signatures, machine learning can help determine whether a signature is due to a Higg Boson or to another collision background event. This paper will explore and assess the performance of machine learning models trained on original CERN data in finding the Higg Boson in a collision event.

II. Exploratory Data Analysis

The challenge can be considered as a binary classification problem. The training set is composed of original CERN data, referencing the features and predictions of 250000 events. The raw data contain 30 features (d=30), which characterize events detected during experiments at the LHC of CERN [2]. All features are continuous, excepted the number of jets PRI_jet_num. This feature divides the data into four categories and defines which ones are meaningful for this given number. Correlation between the features was studied, in order to remove highly correlated features and simplify the model if the user wishes to do so.

III. Data preprocessing

A. Train and test sets

First of all, the entire data set was split into a train and a test set. Each of these sets has itself been divided into 4 groups, each corresponding to a specific value of the categorical feature PRI_jet_num (0,1,2 or 3). The meaningless features regarding the number of jets were discarded according to the documentation on the dataset [2].

Even if most of meaningless data were due to the partition of data depending on the value of the feature PRI_jet_num, the remaining missing or meaningless values were replaced by the median, more robust to outliers than the mean.

B. Preprocessing

We also included three optional preprocessing pipelines which are realized in the following order if jointly applied:

- correlated features removal
- standardization since some methods such as least squares gradient descent are really sensitive to high variance discrepancies between features
- normalization which we mostly applied to the logistic methods to prevent loss divergence

Another implemented treatment is feature augmentation with polynomial features because some models perform better with that type of input. We tuned the best degree for each method by grid search (See below IV).

C. Workflow

We firstly performed a 5-fold cross validation on our train set to obtain the best hyper parameters possible for each model. Those models were then evaluated against our hold out test sample, from which we derived the expected accuracy for the $AICrowd$ submission platform.

IV. Model training

Various models were implemented using 6 different methods:

1) **Least squares**
2) **Least squares gradient descent** (GD)
3) **Least squares stochastic gradient descent** (SGD) using an adapted time step
4) **Ridge regression** using normal equations
5) **Logistic regression** using an adapted time step
6) **Regularized logistic regression** using an adapted time step

For the relevant methods, grid search optimization was performed in order to find the best lambda and/or degree. Two criteria of optimality were studied and are discussed in point VI, namely the parameters with the *highest k-fold test set accuracy* and the parameters with the *smallest k-fold test set loss*.

The different training parameters used in the various methods are: γ (waiting factor), λ (regularization factor), the degree for polynomial expansion, the maximum number of iterations and the batch size for stochastic gradient descent.

Degrees across lambdas were tuned to select models with the highest accuracy. Gamma for each method was defined experimentally.

Cross-validation on the training data was performed to avoid under and overfitting the model, along with regularization methods such as ridge and penalized logistic regression.

V. Results

The following table presents the accuracy obtained for a specific model and its optimal parameters:

TABLE I
Machine Learning Models - Results.

Model	Accuracy	Hyperparameters	γ, iterations
Least Squares GD	0.74346	degree=1	1e-6, 15000
Least Squares SGD	0.74274	degree=1,1,2,4	1e-10, 700
Least Squares	0.77858	degree=4,4,9,9	-
Ridge Regression	0.82446	degree=12 λ=0.001, 0.009999	-
Logistic Regression	0.70682	degree=9 λ= 1e-20	1e-6, 2500
Regularized Logistic Regression	0.65798	degree=1 λ =150	1e-6, 2500

In the following figure, accuracy evolution across for 3 different methods is presented. It points out how well ridge performed for higher degrees where least squares lost accuracy. On the other hand, we can notice an unexpected behaviour for Logistic regression which might arise from loss divergence.

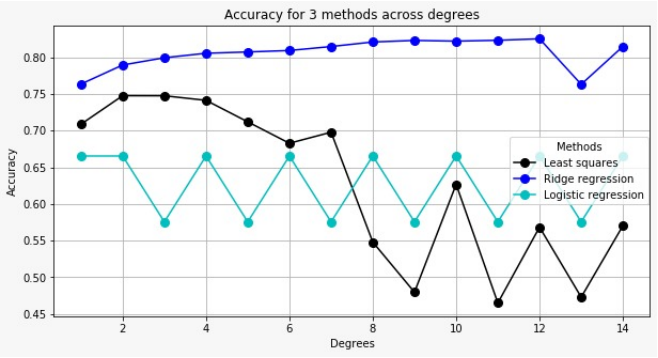

Fig. 1. Accuracy for Ridge regression, Least Squares and Logistic regression depending on the degree for polynomial expansion, for optimal gammas and lambdas

VI. Discussion and model evaluation

Different types of errors were computed as an indicator of the model performance. For the least squares models, the Root Mean Square Error (RMSE) was the reference. For ridge regression, L2 regularized loss was selected. Finally for logistic regression models, normalized negative log likelihood was choosen. Besides, accuracy was favored as comparison criterion between models, since loss is computed differently according to the methods.

In order to be able to compare the performance of the 6 models, accuracy was computed. It is given by the following formula:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Where TP, FP, TN and FN are respectively the number of true positives, the number of false positives, the number of true negatives and the number of false negatives. It gives the number of correct predictions amongst the total number of predictions.

Ridge regression was found to be the best model to train on our data, with lambda parameter 0.009999 and a degree of 12 (Table I), and performed with an accuracy of 0.826 on $AICrowd$ [1]. Optimization of this function was more efficient since it doesn't have a gamma parameter which can make the function diverge easily or converge very slowly if not properly chosen. Ridge regression model performs better than Least squares model (Fig.I) which can be explained by the fact that ridge regression includes regularization of the weights to avoid under- or over-fitting of the model.

Those best parameters were found by grid search for highest accuracy on k-folds. Since this is a classification problem, regularized RMSE minimization didn't result in higher overall accuracy, while optimizing for accuracy on kfold set performed very well at this task for both least squares and ridge methods.

Other parameters to take into account for the model selection are the execution time and the space complexity of the algorithm, which can be crucial depending on the resources made available.

Concerning the removal of highly correlated features in data processing, it didn't impact meaningfully the overall accuracy so it wasn't used systematically in the end.

VII. Conclusion

After discussing the performance of the different models, it appears that ridge regression has the best accuracy on the predictions for the given data. In order to improve the models, some of the following leads could have been further explored:

- Find optimal data preprocessing pipelines in order to yield a better performance for classification models
- In general, a better exploratory data analysis, because a model quality also depends on the quality of the training data
- Investigate other loss types which minimization would result in better accuracy for regression and classification models.

References

[1] EPFL ML 2019. *EPFL Machine Learning Higgs*. https://www.aicrowd.com/challenges/epfl-machine-learning-higgs. 201ç.

[2] CERN Open Data Portal. *Dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014*. http://opendata.cern.ch/record/328. 2014.