Scientific Report : The Higgs Boson

Ekaterina Trimbach, Saad Charkaoui, Sofia Blinova
École Polytechnique Fédérale de Lausanne

***Abstract*—The Higgs boson is an elementary particle that explains why other particles have mass. It is important to be able to classify whether the particle decay signature belongs to the Higgs boson, because it fills other particles with a mass confirming the physical theory. In this report we introduce six methods for this classification task. After the training and validation of our classifiers, we achieved the a 82.3% of accuracy and 72.9% of F1 score with the Ridge regression on the official testing platform. We show that applying some additional pre-processing to the data set and polynomial feature expansion improve our predictions.**

I. Introduction

The aim of the project is to determine whether this decay signature was the result of the Higgs boson. Since many decay signatures look the same, it would be useful to have an automated classification solution. In this work we compared six methods for resolving our classification task: least squares using gradient descent, least squares using stochastic gradient descent, exact least squares, ridge regression, logistic regression and logistic regression with regularization. We used cross validation to tune the hyper-parameters and select the accurate models. Moreover, the performances of the classifiers was improved by applying pre-processing of the data. Finally, we applied feature expansion with polynomial functions to achieve competitive results.

II. Baseline model

To obtain preliminary results of all six models we used raw data with normalization for training and testing processes. For selecting best gradient step size parameter and number of iterations we validated all models with different parameters and found the best for each one.
We split the data set for training and validation with a ratio of 0.8. Thereafter, with the use of cross-validation on the training data set, we tuned the hyper-parameters of the models. Next, using the selected parameters, we tested the selected models on the validation set.

In the table I we provide final base accuracy for every model on the validation set.

For all methods except Least squares SGD, we chose the number of iterations of 1000 and there were no significant changes in the loss function when we increased it. Since the Least squares SGD method is stochastic, more iterations are required for its convergence, so we set 1300 iterations.
The table shows that for raw data using only normalization, the non-iterative exact methods perform better in this case. But in order to improve the prediction of these models, we applied some additional pre-processing and feature expansion on the data set.

Method	γ	λ	Accuracy
Least squares GD	0.07	x	0.6957
Least squares SGD	0.5	x	0.6873
Exact Least squares	x	x	0.7456
Ridge regression	x	1e-5	0.7454
Logistic regression	0.5	x	0.6958
Reg Logistic regression	0.5	1e-6	0.6958

Table I: The best baseline models results.

III. Enhancement of the predictions

A. Data pre-processing

During data analysis under data set we noted some hints that allowed us to achieve better results. In this part we describe our data pre-processing. While researching the data, we noticed that all cells are filled with float except the column [PRO_JET_NUM]. This column takes only four integer values 0, 1, 2, 3, so that can be interpreted as 4 classes of the data set. Hence, we decided to split the data into 4 different sub-data set. We noted that every sub-data set contains columns with identical values and, by dropping it, it would be reasonable to work with these classes separately. For every sub-dataset we individually execute following steps:

Data split. We split every sub-dataset into train and validation samples with proportion 0.8 and 0.2 respectively. Further all statistics and constants obtained during train set pre-process were used for the validation and test sets cleaning. That splitting allows us to tune our hyperparameters and validate our models. Also, track the overfitting.

Identical columns. We noticed that for every sub-dataset exist columns filled only one similar values for every row. These columns are useless in the classification process, because they don't convey extra information about how to distinguish the samples. Hence, they should be dropped. After deleting these columns, the number of features decreased from 30 to 18, 22, 29, 29 for first, second, third and forth sub-dataset respectively.

Missing values. we noticed that for some features the measured value is -999.0. In this case, the parameters are

meaningless or cannot be calculated as specified in the function description. For further experiments, the missing values are imputed with the average of the column in the training sample.

Pearson coefficient. Highly correlated features do not convey extra information and it makes sense to find and remove them. We computed Pearson coefficients for all pairs of features in train samples and drop one column in that pair where Pearson coefficient is greater than 0.8. Thereby the number of features to 16, 28, 23, 25 for first, second, third and forth sub-group respectively was decreased.

Outliers. There were several outliers in the train set, We removed them by applying the formula below for a given value x in the dataset: $|x - mean| > thresh * std$, we set the thresh at the maximum deviation, and in our case we used 5. The mean and std is the average and standard deviation of the corresponding column (respectively)

Data normalization. Normalization makes learning less sensitive to the scale of functions, thus we can solve better the task of finding the minimum of the function.

Test set pre-process. We applied the same data transformations to the validation and test sets as for the train set, but used the constants and statistics obtained from the train set cleaning.

After all these actions, for each sub-set, train, validation and test sets have the same structure and are ready for use.

Method	γ	λ	Accuracy
Least squares GD	0.09	x	0.7432
Least squares SGD	0.2	x	0.7325
Exact Least squares	x	x	0.7558
Ridge regression	x	1e-5	0.7558
Logistic regression	0.5	x	0.7462
Reg Logistic regression	0.5	1e-4	0.7462

Table II: The best models results

Number of max iterations for Least squares SGD is 1500, and 1000 for other models.

B. Polynomial expansion

In this part we expanded the features by applying polynomial functions. We used cross-validation for finding the best polynomial degree to the two best models (logistic regression and ridge regression). We expected that such transformation should improve the performance of both models, but the results we obtained were not so unambiguous.

In Figure 1 we can admit that the accuracy of predictions increases with the increase of the polynomial degree. However, after the growth becomes slower after reaching the $degree = 5$ and stops completely after $degree = 8$. So we concluded the optimal $degree$ is equal to 8 with $lambda = 10^{-5}$.

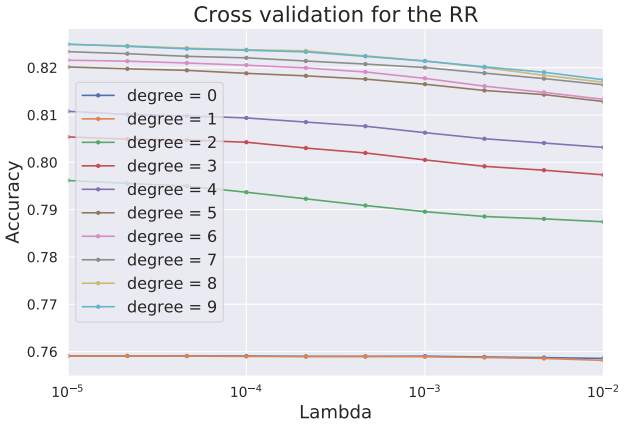

Figure 1: Ridge regression with different degrees

Next step is to explore influence of polynomial expansion on logistic regression with regularization. Results of cross-validation described on the plot 2:

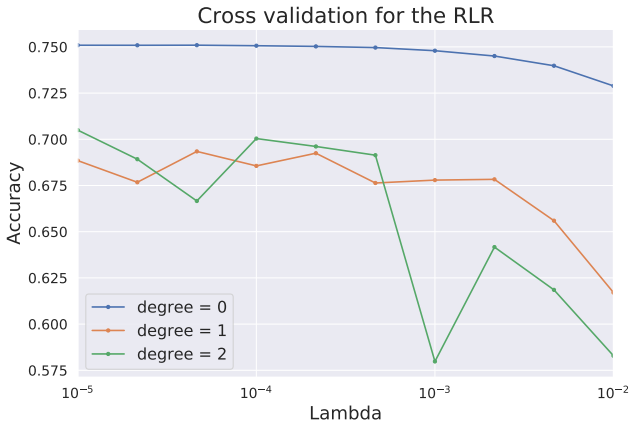

Figure 2: Reg. logistic regression with different degrees

As shown in the Figure 2, the accuracy of the logistic regression is very unstable and sensitive to the degrees higher than 1. Hence, the optimal degree is 0 with an accuracy of 0.750.

Conclusion

After applying the pre-processing and the polynomial expansion on the base model, we can conclude that ridge regression with $lambda = 10^{-5}$ and $degree = 8$ is the most accurate classifier. After testing this model with submission data set, we got an 82.3% accuracy and 72.9% f1 score.

Acknowledgements

The authors thanks Machine Learning and Optimization Laboratory - EPFL contribution in our basic ML knowledge.