Discovering the Higgs boson using machine learning

Lars Quaedvlieg, Anton Pirhonen, Ana Serrano Leiva
Machine Learning (CS-433), École Polytechnique Fédérale de Lausanne — Fall 2022
Group CheRuben

***Abstract*—This paper studies different machine learning models applied to the data collected from the experiments performed with the CERN particle accelerator with the aim of discovering the Higgs boson particle. The logistic regression using Accelerated Gradient Descent with Restart (AGDR) was found to perform the best, achieving a categorical accuracy of 0.819 and F1-score of 0.729 in the benchmark.**

I. Introduction

The Higgs boson is an elementary particle in the Standard Model of particle physics which explains why other particles have mass [1]. It can appear momentarily after particles are collided with each other at high velocities. The boson cannot be observed directly as it decays rapidly into other particles. Thus, its presence must be inferred from its decay signature that result from its decay process.

This paper aspires to estimate the likelihood of a collision producing a Higgs boson given a vector of features that represents the decay signature of a collision event. This paper utilizes different machine learning models for binary classification. The models are trained and tested with CERN particle accelerator data from [2].

***Keywords*-Higgs boson classification, data cleaning, feature engineering, linear regression, logistic regression**

II. Methodology

A. Data pre-processing and feature engineering

Two data sets: the testing set and the training set, are used for this experiment. The training set consists of 250,000 labeled data points and the test set of around 570,000 unlabeled data points, where each data point is the measurement data from a single particle collision experiment. This data contains 30 features from which 17 are raw quantities (primitives) from the collision measured by the detector and the 13 other (derived) features are quantities derived from the primitive features. Additionally, the training set contains binary labels. The signal label 's' denotes the presence of a Higgs boson and the background label 'b' its absence in an experiment. From the training data, around 34% are signals.

The first steps of this paper establish data pre-processing and feature expansion. During these steps, the data is filtered, normalized and new features are added.

One feature of the data point, "PRI jet num", is categorical, and and it affects many other feature values [2]. The categorical variable and the affected variables are augmented into multiple variables, which contain the corresponding one-hot-encoded values of the initial variable. This way, the applied models can construct multiple weights corresponding to the values of the categorical variable.

As mentioned by Adam-Bourdarios et al. [2], the data set contained undefined values which were marked as having value -999.0. To prevent the missing values from affecting the normalization of the data, these values were set to the Python NaN, denoting a missing number. Afterwards, the data was normalized to have zero mean and unit variance. Finally, the missing values were set to zero which corresponds to the empirical means of the features after standardization. This normalization was applied to all columns except the one-hot-encoded feature columns.

After standardizing the features, the absolute value of the Pearson correlation coefficients between each feature pair is computed. Table I shows the number of feature pairs that have at least a certain correlation. This minimum correlation will be referred to as "correlation cutoff threshold" later in the paper.

Minimum correlation	0.8	0.6	0.4	0.2	0.0
Number of feature pairs	40	51	91	137	561

Table I: The number of feature pairs given a certain minimum correlation for each pair

This paper experiments with adding a differing amount of interaction terms of certain features (including the feature with itself) to create a simple but powerful degree 2 polynomial basis for the model. This basis allows the model to fit the data better and circumvent underfitting. To avoid overfitting or producing multicollinear features, it is ensured that the model performs well in both the training set and the test set.

B. Models

Multiple approaches are implemented and benchmarked in this paper. Initially, linear regression is employed and performed using the linear least squares approach (LS) for approximating the solution to the linear equations. Then, ridge regression is used and tested to apply regularization to the linear regression. Finally, (regularized) logistic regression is implemented with two optimizers: gradient descent (GD), and a custom algorithm called accelerated gradient descent with restart (AGDR). The latter algorithm uses momentum to help the weights convergence more quickly [3], and resets its momentum when it appears to be going in a direction that does not optimize the loss function [4]. The loss function for logistic regression is adapted in order to avoid numerical instability by adding a sufficiently small constant $\epsilon = 10^{-8}$ inside the logarithm terms. It is important to note that all optimization problems mentioned above are (strongly) convex and smooth, meaning the optimization is guaranteed to converge to a global minimum.

III. Experiments and Results

The first experiment performed is a benchmark between the algorithms that were mentioned in section II-B. The tests are performed by splitting the labeled data into two sets: one used for training the model, and the other used for measuring the models performance. This test is performed ten times for each model. The error margins are calculated from the test results by calculating the greatest absolute deviation from the mean of the experiments. The test results are shown in Table II. Additionally, 4-fold cross-validation was performed for the regularized logistic regression and ridge regression for selecting the optimal regularization coefficients (λ), where it was noted that the regularization did not improve the models' performance.

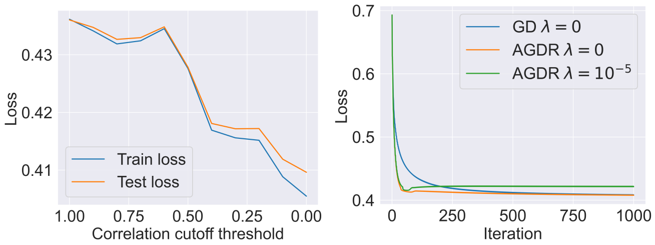
The following models were used in this benchmark:

Model A Always choose background label (ZeroR) (baseline)
Model B Linear regression (LS)
Model C Ridge regression (LS)
Model D Logistic regression (GD, $\lambda = 0$)
Model E Logistic regression + interaction terms (AGDR, $\lambda = 0$)
Model F Logistic regression + interaction terms (AGDR, $\lambda = 10^{-5}$)

Model	Test Accuracy	Test F1-Score
A	0.658 ± 0.003	0.000 ± 0.000
B	0.798 ± 0.002	0.682 ± 0.005
C	0.798 ± 0.002	0.682 ± 0.005
D	0.726 ± 0.003	0.453 ± 0.006
E	0.819 ± 0.002	0.729 ± 0.006
F	0.800 ± 0.003	0.645 ± 0.005

Table II: Accuracy benchmark with different models on a testing dataset with 50,000 samples

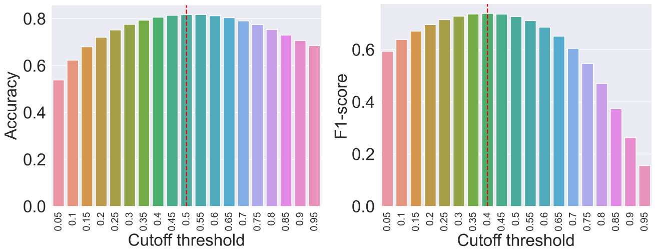
As can be seen in table II, the best model in terms of both accuracy and F1-score uses logistic regression with interaction terms and is optimized using AGDR.



(a) Loss for converged logistic regression with added interaction terms for different correlation cut-off thresholds

(b) Change in test set losses for different optimization algorithms of logistic regression during training

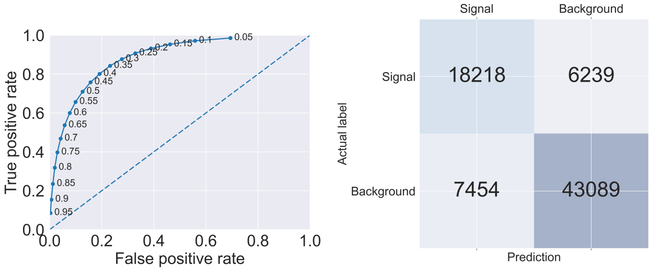
The previous figure shows that when adding more interaction terms, the test error still decreases. This indicates that the model is not overfitting, although the relevancy of the features seems to be decreasing. Hence, all interaction terms are added to the final model. Further, figure 1b shows that AGDR converges much faster than regular gradient descent.



(a) Bar chart showing accuracies over different cutoff thresholds for logistic regression

(b) Bar chart showing F1-scores over different cutoff thresholds for logistic regression

The figures above show that, for logistic regression, the best classification threshold in terms of accuracy and F1-score on the test dataset lies around 0.5, with little variation in performance close to 0.5. However, this still means that the model favors choosing one class over the other.



(a) ROC curve for logistic regression classifier for different cutoff thresholds

(b) Confusion matrix for logistic regression model with cutoff threshold 0.5

The ROC curve in figure 3a shows that there is no large variation in TPR versus FPR for thresholds around 0.4. Furthermore, after fixing the threshold to equal 0.5, figure 3b displays that positive samples are more often misclassified than negative samples on the test dataset.

IV. Discussion and Conclusions

This paper performed regression analysis to the task of Higgs boson detection from the Large Hadron Collider proton collision data. Multiple regression models were considered from which the logistic regression using Accelerated Gradient Descent with Restart (AGDR) was found to perform the best. The model achieved categorical accuracy of 0.819 and F1-score of 0.729 in the test set used for benchmarking. Furthermore, the importance of adding more complex features to address underfitting, in our case by adding interaction terms of features, is highlighted in the experiments.

One interesting idea for further research would be to look into higher-order polynomial terms while preserving memory usage.

References

[1] M. Carena and H. E. Haber, “Higgs boson theory and phenomenology,” *Progress in Particle and Nuclear Physics*, vol. 50, no. 1, pp. 63–152, 2003.

[2] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, “The higgs boson machine learning challenge,” in *NIPS 2014 workshop on high-energy physics and machine learning*. PMLR, 2015, pp. 19–55.

[3] Y. Nesterov, “A method of solving a convex programming problem with convergence rate mathcal {O}(1/kˆ {2}),” in *Sov. Math. Dokl*, vol. 27.

[4] B. O’donoghue and E. Candes, “Adaptive restart for accelerated gradient schemes,” *Foundations of computational mathematics*, vol. 15, no. 3, pp. 715–732, 2015.