Higgs Boson; To be, or Not to Be

Amir REZAIE, Arash ASKARI, Antonin VIANEY, *EPFL Lausanne, Switzerland*

Abstract—**Due to rapid decay, observing Higgs Boson particles is not feasible. However, with the help of machine learning algorithms, it is possible to detect them using the features recorded at the laboratory. In this article, *logistic regression* is used as the classifier to detect Higgs Boson particles. Various augmentation types are applied to capture non-linearity of the decision boundaries. Among all performed methods, it is observed that using Gaussian basis functions results in a higher accuracy of the prediction of validation and test data. Moreover, it is shown that *bagging* can increase the accuracy and F-1 score to 0.835 and 0.751, respectively.**

I. Introduction

Machine learning methods are used in various branches of *physics* to help to predict the phenomena that cannot be accurately explained with physical-based models. Detecting Higgs Boson particles are one of which that requires the help of machine learning. In this article, regularized logistic regression with augmented features is used to predict the presence of Higgs Boson particles. The effect of different augmentation ways including Gaussian basis functions, square root, differences, ratios, logarithm, and inverse of logarithm on the accuracy of models is explored. It is found that augmenting features using Gaussian basis functions leads to the highest accuracy and also differences between features has the lowest increase of accuracy. Additionally, bagging increases the accuracy of the predictions. In the what follows, the pre-processing of data is firstly explained; secondly, the feature conditioning methods are mentioned, thirdly, the results of trained models are discussed and finally the conclusion of the study is summarized.

II. Pre-processing

The first step in each machine learning setting is pre-processing of the raw data. The main goals are to understand better the type of each feature (i.e. whether it is categorical, ordinal or continuous), the distribution of the data, the interaction between features and treating missing values and outliers.

In the current database, almost 70% of the samples has at least one feature, which is not recorded, missing or meaningless (i.e. equal to -999). Thus, common approaches like replacing theses values with the mean or median of each feature or removing them result in the manipulation of large portion of data, which is not desirable. To handle these values, partitioning of the data is necessary that will be explained in what follows.

Investigating the type of each feature, it was observed that all features are continuous except the feature ”PRI_jet_num”, which can only take four integer values 0, 1, 2 or 3. Therefore, the data was divided into four categories namely ”Group 0”, ”Group 1”, ”Group 2” and ”Group 3”. As a result of doing this, some features could be removed from the design matrix of each group as they were equal to -999.

After categorizing data into four groups and removing features, there are still some samples in which the feature ”DER_mass_MMC” is equal to -999. To handle these missing data, two approaches were applied. In the first approach, the missing value of the feature ”DER_mass_MMC” was replaced with the median. This approach was used for all the analyses except the best submission of the competition part. In the second approach used only for the competition part, each category of data, i.e. Group 0, Group 1, Group 2 and Group 3, was divided into two subcategories, according to the existence (i.e. a meaningful value) or non-existence (i.e. equal to -999) of the feature ”DER_mass_MMC”.

Th next step was removing the outliers. For that, the box plot of each feature was plotted and samples where at least one of their feature was completely separated from the rest of the data, was removed from the database.

The last performed pre-processing step, was removing correlated features. In this project, only the features with the correlation factor of almost one (more than 0.99) was removed.

III. Feature conditioning

For problems where the data is not linearly separable, it is common to use some non-linear mapping functions to unfold the data and to be able to benefit from linear classifiers like logistic regression. In this report, the data was augmented with engineered features and the effect of each of which was elaborated on the accuracy of the prediction of training, validation and test data. The functions used to augment the data are mentioned in Table I.

TABLE I: Feature augmentation types

Model	Engineered feature type
Aug. model 01	Gaussian basis function* $\exp[-1/2\sigma_i^2(\mathbf{x}-\mathbf{c}_i)^\top(\mathbf{x}-\mathbf{c}_i)]$
Aug. model 02	Square root $\sqrt{\mathbf{x}}$
Aug. model 03	Difference $x_i - x_j$
Aug. model 04	Ratio x_i/x_j
Aug. model 05	Logarithm $\log(\mathbf{x})$
Aug. model 06	Inverse of Logarithm $1/\log(\mathbf{x})$

* σ_i and $\mathbf{c}_i$ are standard deviation (set to one) and center of basis functions (set randomly), respectively.

IV. Building models

The ”logistic regression” with the l_2 norm regularization term was used as the classifier of our binary classification problem. The details of the utilized learning model is summarized in Table II.

TABLE II: Hyper-parameters settings

Optimizer	Gradient descent
Maximum number of iterations	5000
Learning rate	2×10^{-6}
Regularization strength parameter	$\{0, 10^{-10}, 10^{-6}, 10^{2}, 10^{4}\}$
Ratio of the training to validation set size	0.8

One base model, i.e. without feature augmentation, plus six models each of which with a specific type of feature augmentation

TABLE III: Accuracy results

	Group 0		Group 1		Group 2		Group 3	
	Train acc.	Valid acc.	Train acc.	Valid acc.	Train acc.	Valid acc.	Train acc.	Valid acc.
Base model	0.825	0.828	0.713	0.713	0.740	0.737	0.729	0.741
Aug. model 01 (3 basis function)	0.839	0.845	0.773	0.777	0.794	0.792	0.773	0.780
Aug. model 01 (5 basis function)	0.845	0.849	0.807	0.805	0.828	0.827	0.818	0.822
Aug. model 01 (8 basis function)	0.846	0.850	0.808	0.808	0.832	0.831	0.829	0.831
Aug. model 01 (10 basis function)	0.846	0.850	0.808	0.808	0.833	0.833	0.831	0.831
Aug. model 01 (15 basis function)	0.846	**0.850**	0.809	**0.808**	0.834	**0.833**	0.833	0.833
Aug. model 01 (20 basis function)	0.832	0.834	0.763	0.766	0.831	0.831	0.834	0.835
Aug. model 01 (30 basis function)	0.813	0.814	0.722	0.719	0.788	0.784	0.836	**0.836**
Aug. model 02	0.832	0.836	0.761	0.762	0.791	0.790	0.767	0.775
Aug. model 03	0.825	0.827	0.713	0.712	0.740	0.738	0.727	0.742
Aug. model 04	0.837	0.840	0.755	0.753	0.793	0.789	0.780	0.791
Aug. model 05	0.835	0.840	0.777	0.778	0.809	0.803	0.796	0.804
Aug. model 06	0.834	0.839	0.776	0.778	0.808	0.802	0.801	0.806

method (see Table I) were trained with a similar set of hyperparameters (see Table II). The accuracy of the trained models on the ”training set”, ”validation set”; and ”test set”, and ”F-1 score” on the ”test set” are summarized in Table III. It must be mentioned that for all augmentation types, models with the regularization parameter equal to zero had the highest accuracy on the validation set. Therefore, for brevity, only the results corresponding to models with zero regularization parameter are reported. Moreover, for the ”Augmented model 01”, the features were augmented using 3, 5, 8, 10, 15, 20, 30 Gaussian basis functions to investigate the optimum number of Gaussian basis functions to use.

According to Table III, it is observed that generally augmenting data using each of the proposed methods, can increase the validation accuracy of of the prediction (except the augmented model 03, which resulted in a slightly lower accuracy of predicting the validation set for the Group 0 and Group 1).

Based on the results of the accuracy of the prediction of the validation and test data, the applied augmentation methods can be ranked from the highest to the lowest increase in the accuracy metric as follows: 1) Gaussian basis function (at least 5 basis function); 2) Inverse of log; 3) Log; 4) Ratios; 5) Square root; and 6) Differences.

As mentioned earlier, for the competition part of the project, the data of each group, was subdivided into 2 more sub-groups based on the existence of the feature ”DER_mass_MMC”. Therefore, 8 models with augmented features using 15 and 10 Gaussian basis functions for the sub-groups with and without the feature ”DER_mass_MMC”, respectively, were trained. The hyperparameters for each of these models were determined heuristically by trying various ranges of each. The concept of ”bagging aggregation” was utilized to increase the accuracy of the predicting model. For that, 301 models were trained each of which using 10% of the data as the training data (that were selected randomly with replacement). The final prediciton was the aggregation of these models (majority voting). This procedure resulted in a slightly higher accuracy (0.835) and F1-score (0.751) on the test data compared to the highlighted result shown in Table IV. It should be mentioned that various approaches, including down-sampling the data and weighted cross-entropy loss function to penalize more the models that cannot detect signal, were applied to account for the unbalanced data (especially for the Group 0 and Group 1), but they did not show a significant improvement in the accuracy and the results are not reported.

TABLE IV: Accuracy and F1-score of test data

Model	Test data	
	Accuracy	F1-score
Base model	0.765	0.621
Aug. model 01	**0.832**	**0.746**
Aug. model 02	0.797	0.681
Aug. model 03	0.765	0.624
Aug. model 04	0.797	0.686
Aug. model 05	0.809	0.705
Aug. model 06	0.809	0.707

V. Summary

Detecting Higss Boson particles using machine learning methods is a challenging task, which requires adequate pre-processing, features engineering and model selection. In this report, logistic regression was used to classify signals from backgrounds. For that, first some pre-processing steps including removing outliers, categorizing data into groups and removing irrelevant features were performed. In the second step, data was augmented using different mapping functions, such as Gaussian basis functions, log, inverse of log, differences, ratios and square root of features. The results showed that using Gaussian basis function and differences between features has the highest and lowest increase in the accuracy of predictions, respectively.

For the competition part of the project, each group was divided into two sub-groups according to the presence of the feature ”DER_mass_MMC”. Data was augmented using Gaussian basis functions. To increase the robustness of predictions, bagging was used resulted in a higher accuracy (0.835) and F1-score (0.751) of the prediction of test data, compared to the case where data was not divided into two more sub-groups and bagging was not used (accuracy = 0.832, F1-score = 0.746).