The Higgs boson machine learning challenge

Bruno Magalhaes[1], Marie Drieghe[2], Nicolas Casademont [3]

Abstract
Decay signatures of proton-proton collisions led to the discovery of the Higgs Boson. This new particle was discovered by analysing areas with a significant amount of unknown events (signals) against others that contain known ones (background). To tackle the challenge, we present a Machine Learning (ML) algorithm for the classification of these areas. Our algorithm reaches more than 80% accuracy by combing feature engineering and logistic regression. Due to the high-dimensionality of the solution, we perform iterative stepping via the stochastic gradient descent method. We detail a set of pre-processing methods that allow for shorter data set, smaller dimensionality, faster execution time, filtering of outliers, and imputation of missing values.

Email (SCIPER): [1] bruno.magalhaes@epfl.ch (212079) [2] marie.drieghe@epfl.ch (273688) [3] nicolas.casademont@epfl.ch (223485)

We present the sequence of pre-processing filtering steps tested in Section 1, where we explain their rationale and detail the results. The data after filtering is then used in the Machine Learning methods described in Section 2 to create a classification model. Results and final remarks are detailed in Section 3.

1. Pre-processing

1. Removal of features missing values We tried filtering out the columns referring to features with a number of unknown elements exceeding a given threshold (we tried 30% as well as 50%). This allows us to filter out dimensions that seemingly do not offer any added value. This feature is exposed via the `clean_data_by_removing` method in `clean_data.py`. For the remaining missing values mean imputation is used. The removing of features did not turn out to give better results than a mean imputation for all features.

2. Mean Imputation The technique of assigning the average value of a feature to the missing values [1]. This avoids the unfavourable effect of the large -999 values. Testing showed that this technique for all features offers superior results compared to the removing of features.

3. Outliers from median filter A common technique applied to non-accurate multi-dimensional data, particularly pictures. It performs filtering of outliers based on a median filter, a nonlinear digital filtering technique often used to remove noise [2]. The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighbouring entries. This feature is executed by activating the `filter_median = True` flag. Nevertheless, the results of applying this filter are slightly degraded, as multi-dimensional *smoothing* of data makes sense when dimensions carry similar information (e.g. pixels gradient or positions) which is not our case.

4. Mahalanobis Distance We apply a filtering process based on the Mahalanobis distance (MD)[3], a measure of the distance between a given point and a distribution. The MD is unitless and scale-invariant, and takes into account the correlations of the data set. It is also shown to be efficient in multi-dimensional distance-based approaches filters, particularly compared with Euclidian distance filters [1]. In general, a small threshold only keeps values close to the mean while a large value will also keep samples that are further away. The implementation is available in the method `MD_removeOutliers(x, y, threshold_scale = 1.5)` or by setting the flag `filter_outliers = True`. The default threshold value utilized is 1.5, leading to a removal of 13284 outliers or 6.3% of data. We implemented cross-validation to verify if the outlier removal has a negative effect on the classification and it does so we did not include it in the final filtering.

5. Principal Component Analysis The PCA method [5] is a widely applied filtering for reducing the complexity of multi-dimensional data. The outcome is a dataset of reduced dimensionality where the least meaningful components are removed and reflected on the remaining dimensions. When applied to our dataset, the PCA revealed that: (a) the covariance matrix (not displayed for brevity of report) shows that most dimensions are unrelated; and (b) approximately half of dimensions are relevant in capturing the data organization. Our PCA results were validated against the `sklearn.decomposition.PCA` library and are presented in Table 1. When applied to our dataset, the PCA filtering did reduce the loss on our classification. This together with the results from the Mahalanobis Distance filtering leads us to believe the data has been pre-filtered.

6. Data Standardization The final process performs data standardization by subtracting the mean and dividing the values by the standard deviation. This algorithm is particularly helpful to avoid computational overflows. It is only required when PCA is not executed beforehand, as the PCA implicitly executes the standardization.

2. Machine Learning Methods

We implemented the following three least squares methods: `least_squares`, `least_squares_GD` and `least_squares_SGD` for least squares with normal equations, regular and stochastic gradient descent. We also implemented `ridge_regression` for ridge regression using normal equations, `reg_logistic_regression` for regularized logistic regression

[1] It has been shown that typical distance filters fail on complex multi-dimensional data, since as dimensionality increases the distance to the nearest neighbour approaches the distance to the farthest neighbour [4]

Dim.	Eigen value	Explained var. ratio	Dim.	Eigen value	Explained var. ratio
0	1983617.8	7.43 e-1	**15**	3.98	1.49 e-6
1	470787.3	1.76 e-1	**16**	3.49	1.31 e-6
2	152814.2	5.72 e-2	**17**	2.74	1.02 e-6
3	35127.6	1.31 e-2	**18**	2.45	9.18 e-7
4	18184.4	6.81 e-3	**19**	1.65	6.19 e-7
5	2235.3	8.37 e-4	**20**	1.40	5.27 e-7
6	1833.1	6.86 e-4	**21**	0.99	3.73 e-7
7	1266.6	4.74 e-4	**22**	0.74	2.79 e-7
8	944.0	3.53 e-4	**23**	0.67	2.54 e-7
9	519.3	1.94 e-4	**24**	0.61	2.30 e-7
11	384.3	1.44 e-4	**25**	0.18	6.99 e-8
10	374.8	1.40 e-4	**26**	0.11	4.30 e-8
12	178.7	6.69 e-5	**27**	0.05	2.17 e-8
13	138.8	5.20 e-5	**28**	0.024	9.21 e-9
14	45.78	1.71 e-5	**29**	7.48e-8	2.80 e-14

Table 1. Details on the eigen-vectors after PCA filtering. Features are sorted by relevance (explained variance ratio).

Degree	Accuracy (%)
3	60
4	87
5	84
8	79

Table 2. Accuracy of predictions on the test data for different degrees of polynomials

using GD and SGD and `logistic_regression` for logistic regression with GD and SGD.

Our ideal solution relies on a combination of techniques. To obtain the best results we focus on **logistic regression**, the only binary classifier, as it provided better results than the available alternative methods. The final pre-filtering consisted of **mean amputation** over all features and standardization. After that we added a **polynomial interpolation of the fourth degree** that provided better result compared to the linear counterpart (see accuracy results for different degrees in Table 2).

Due to the high amount of computation required, we use logistic regression with **stochastic gradient descent**.The step size gamma used for gradient descent was found with a binary search across a scope of parameters that yield low time to solution and low loss and set to $1.1e-03$. Finally, we tried to find a better solution using regularized logistic regression and 4-fold cross-validation for obtaining the best lambda. But since the difference between the error in the training data and the error in the test data was negligibly small (see Figure 1) we did not pursue the regularized logistic regression further.

3. Discussion and Summary

We applied machine learning techniques to data collected from the CERN particle accelerator aiming at recreating the process of discovering the Higgs particle. We detailed the challenges with the collected data — many missing values and high dimensionality — and presented an exploration of pre-filtering and transformation methods for reducing complexity (least significant dimensions) and sample size (outliers), and auto-completion of missing information (mean imputation). We presented the rationale behind an accurate classification based on a combination of algorithms. Our final algorithm can be executed by running `run.py`. It uses logistic regression with stochastic gradient descent. We cleaned and standardized the data

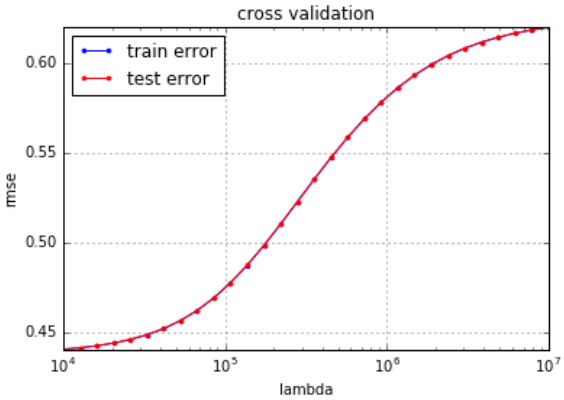

Figure 1. Cross-validation results for test vs training data.

and enriched it by creating a polynomial function of degree 4. This algorithm gave us the following score on Kaggle: 0.81192 .

References

[1] A Rogier T Donders, Geert JMG van der Heijden, Theo Stijnen, and Karel GM Moons. Review: a gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[2] Zhou Wang and David Zhang. Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(1):78–80, 1999.

[3] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[4] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[5] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.