Methods in machine learning applied to high-energy physics

Catarina Pires, Artur Stefaniuk & Lisa Warners
Project 1, CS-433 Machine Learning
EPFL

***Abstract*—Methods in machine learning can potentially increase the efficiency and accuracy of data analysis that is essential to contemporary science. This report discusses pre-processing of data corresponding to the discovery of the Higgs boson and using the subsequent training model based on penalised logistic regression. Model hyperparameters were tuned using 4-fold cross validation. The obtained accuracy when compared to the test data is 81.2%, confirming that the application of basic machine learning models to such a data set was successful.**

I. Introduction

With the arrival of particle accelerators, the analysis of big data sets became essential to the advancement of high energy physics. The primary purpose of the Large Hadron Collider (LHC) was to confirm the hypothesised existence of the Higgs boson. In 2012, a particle with the correct mass was found [1], which was subsequently confirmed to be the Higgs boson in 2013 [2]. Most of the obtained data in high energy physics corresponds to known decay processes and particles, and are thus background for the purpose of particle discovery [3]. However, in the case of undiscovered particles it is possible to find so-called excess data, which are labelled as signal. If found to be statistically relevant, the particle's existence is confirmed. Therefore, separation of background from signal data is an essential step.

In this report, we will set forth an approach to train weights assigned to a selection of features, exploring the potential of machine learning in the analysis of high energy physics. Implementation of such methods can improve separation and subsequent analysis of data. The process includes both feature processing and model design, taking into account the physical properties of the system. Although the data set in this report is reasonably small, all decisions were made with the intention to keep a low cost computationally while achieving a good prediction.

II. Models and methods

To achieve the goal as stated in section I, a process broadly divided into pre-processing and training including hyperparameter selection using cross-validation was carried out. All implementations were done in Python without the use of external libraries (except NumPy).

A. Pre-processing

In this report, the input data will be denoted by X, the labels by y, and the assigned weights by w. As the data and the definition of certain features depends on the physical nature of the event, the data set was divided into four groups based on the number of jet particles detected (between 0 and 3). First, for the estimated mass of the Higgs boson candidate, undefined values appear in all groups, indicating a large deviation from expected topology. The undefined values for the estimated mass were replaced by the most frequent integer value in that column. Non-defined values in the remaining columns were set to 0 following the reasoning that the momentum of a (sub)leading jet must be 0 if this particle does not exist. The features for the four groups were also evaluated in terms of variance, with a threshold of 0, excluding features lacking information content.

To minimise the effect of extreme values due to measurement errors, the data points further than $\tau = 5\sigma$ from the mean were identified. All these values were clipped at τ, as to preserve the fact that they are reasonably large, but prohibit unbalanced effects on the weight calculations.

Extra features were added by building a polynomial basis. The degree d of polynomial was determined automatically in combination with regularization hyperparameter λ for the penalty (see next subsection). This was done to build a model that allows for non-linearity, since there is no specific reason a linear relation should be perceived in these data.

Finally, the data was standardised by dividing by the mean and subtracting the standard deviation for each feature.

B. Training implementation

A number of methods was implemented in order to analyse data: linear regression via (stochastic) gradient descent, least squares and ridge regression using normal equations, and (regularised) logistic regression using full gradient descent. The first three implementations are based on finding the minimum value of the mean squared error and interpreting the resulting output as probability values. The two logistic methods calculate the loss based on the negative log-likelihood which maximises the probability of the correct classification of data points into two categories. As the training of this categorical data involves binary classification, a logistic approach would generally be more applicable for this problem. Both the gradient descent and the closed form solution of the least squares approach are sensitive to extreme values and show issues in regarding the output as a probability due to the range of values

exceeding $[0, 1]$ [4]. To verify this reasoning, the accuracy of all functions was determined with cross-validation.

General ridge regression as implemented here is based on the normal equations as to minimise mean squared error. This solution is shown in equation 1. For the commonly used ℓ_2-regularisation, parameter λ can be applied to avoid overfitting, generally associated to high values of w [5]. This is more applicable here than ℓ_1-regularisation, considering the low number of features and little previous knowledge on correlated features incorporated in pre-processing [6].

$$w_{\ell 2-normal} = (X^T X + 2N\lambda I)^{-1} X^T y \qquad (1)$$

Similarly, application of the log likelihood principle and ℓ_2-regularisation to the logistic function and subsequent rearrangement of terms results in equation 2.

$$L_{\ell 2-logistic}(w) = \lambda ||w||_2^2 + \sum_{n=1}^{N} \ln[1 + e^{x_n^T w}] - y_n x_n^T w \quad (2)$$

This loss function lacks a closed-form solution but was minimised using conventional gradient descent, which was selected for its favourable cost to accuracy balance. Step size γ was chosen to be 0.1 for ridge regression and 0.001 for logistic regression based on a trial-and-error trade-off between convergence and stability. For logistic regression the step size γ was halved every 10 iteration in order to force the convergence of the loss function. The minimal value of $L(w)$ as found by gradient descent depends on the regularization parameter λ used in equations 1 and 2. In order to find the optimal values for both degree d and λ, 4-fold cross validation was applied to all jet groups. The combination (λ, d) resulting in the largest accuracy was determined for $d \in [1, 3]$ and $\lambda = 10^p, p \in [-4, 0]$. The range for d was chosen to balance computational cost and loss minimisation. The values of λ were based on previously reported typical values for this parameter [7].

III. Results

The methods as described in section II were carried out in order to produce the background and signal labels, known for the set of data used for training. To minimise the run time of the final code, the optimal lambdas and degrees were determined separately. There was a definite trend seen in higher accuracy as the degree for the polynomial basis increased, as visualised in figure 1. However, the computational time also strongly increased with the degree, so the final degree selected was 3. The best lambdas per jet group were determined for this separately, and can be found in figure 2. The lambda was found to have a small yet noticeable effect on the final accuracy of the prediction (figure 3).

To determine weights resulting in the highest accuracy, both the regularised normal equations and logistic regression

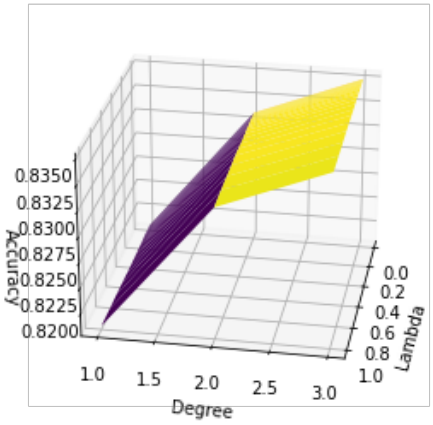

Figure 1. The dependency of the accuracy on degree and lambda selection. Shown for normal equations for the group with jet number = 0.

jet number	0	1	2	3
λ	0.0587	0.492	1.000	0.00100

Figure 2. The optimal lambda values for degree 3 found using a logarithmic range of lambdas between 10^{-4} and 1 using regularised normal equations.

were applied. Contrary to the reasoning as followed in section II, the regularised normal equations resulted in a higher accuracy. This is most likely due to overfitting in the logistic regression, combined with reduced effects of outliers due to the implemented cut-off. The final obtained accuracy on the test data was 81.2%.

IV. Conclusion

Implementation of smart data pre-processing and subsequent application of relatively simple machine learning models results in a prediction of 81.2% accuracy for the separation of background and signal events in data from high energy physics. With a plethora of more sophisticated models available, this report shows the applicability of the general principles to the field of particle physics with many potential improvements.

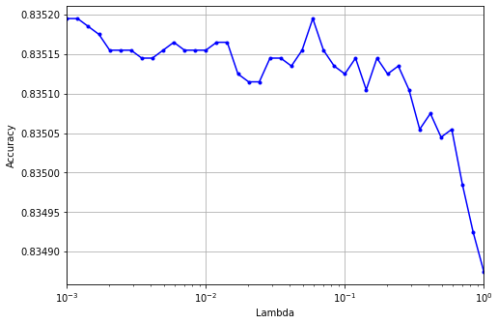

Figure 3. Accuracy depending on lambda selection within the polynomial basis with degree 3. Shown for normal equations on jet number = 0.

REFERENCES

[1] "CERN experiments observe particle consistent with long-sought Higgs boson," Jul 2012. [Online]. Available: https://home.cern/news/press-release/cern/cern-experiments-observe-particle-consistent-long-sought-higgs-boson

[2] "New results indicate that particle discovered at CERN is a Higgs boson," Mar 2013. [Online]. Available: https://home.cern/news/press-release/cern/new-results-indicate-particle-discovered-cern-higgs-boson

[3] C. Adam-Bourdariosa, G. Cowanb, C. Germain, I. Guyond, B. Kégl, and D. Rousseau, "Learning to discover: the Higgs boson machine learning challenge," Jul 2014. [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf

[4] R. Vasudev, "How are Logistic Regression & Ordinary Least Squares Regression (Linear Regression) Related? Why the 'Regression' in Logistic?" Jun 2018. [Online]. Available: https://towardsdatascience.com/how-are-logistic-regression-ordinary-least-squares-regression-related-1deab32d79f5

[5] A. Nagpal, "L1 and L2 regularization methods," Oct 2017. [Online]. Available: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

[6] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," 2009, department of Statistics, Stanford University.

[7] J. Brownlee, "How to use weight decay to reduce overfitting of neural network in keras," Oct 2018. [Online]. Available: https://machinelearningmastery.com/how-to-reduce-overfitting-in-deep-learning-with-weight-regularization/