Project 1 - Higgs Boson Challenge

Emilio Fernández, Tyler Benkley, Frédéric Berdoz
Machine Learning, EPFL, 2019

***Abstract*—This short report presents in detail a solution to the Higgs Boson Challenge. It also contains all the choices leading to this solution and a discussion on potential further improvement.**

I. Introduction

The Higgs Boson challenge was invented with the aim of promoting collaboration between data scientists and physicists. The problem is the following: Based on a given set of measurements of an event, construct a model that predicts whether or not an event will give rise to a Boson. Our role was to implement methods introduced in our Machine Learning class to build such a model and fine-tune its associated parameters.

II. Models and Methods

A. Exploratory data analysis

As explained in [1], the data provided in the challenge is simulated using the official ATLAS full detector simulator. The train and test data sets contain 250'000 and 568'238 data points, respectively. Each data point has 30 features:

- 17 directly measured by the detector (primitives),
- 13 computed using the primitive feature (derived).

Some data points have undefined features, generally depending on their jet number (PRI_jet_num). The estimated mass of the Boson candidate (DER_mass_MMD) is also undefined for certain events. Moreover, the scalar sum of the transverse momentum (PRI_jet_all_pt) is 0 if the number of jet is 0.

B. Feature engineering

Based on the description of the features, we know that a jet is a type of pseudo particle that is created as a result of a collision of other particles. We decided to categorize the data depending on the jet number such that for each subset the corresponding undefined features were removed. Furthermore, for every subset there were also Nan values in the feature DER_mass_MMC and we replaced them with the mean of this feature. In conclusion, we ended up with four data-sets devoid of any Nan values. In addition, we also extended our feature vector by building a polynomial basis of degree d from our features in order to increase the flexibility of the regression.

C. Model selection

Once the data had been treated correctly, a model had to be selected in order to perform the machine learning algorithm. For this project, we chose to implement a model that was introduced in class, i.e. *least-squares*, *ridge-regression*, *logistic regression* and *regularized logistic regression*. *Regularized logistic regression* and *ridge regression* have a regularization term, which is important if one wants to limit overfitting by penalizing large weights. They were therefore favored over the *logistic regression* and *least-squares* models, respectively. Their corresponding loss functions $\boldsymbol{\mathcal{L}}(\mathbf{w})$ are given by:

- *Ridge regression*[1]:

$$\boldsymbol{\mathcal{L}}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left[y_n - \mathbf{x}_n^\mathsf{T}\mathbf{w}\right]^2 + \lambda\|\mathbf{w}\|_2^2, \qquad (1)$$

- *Regularized logistic regression*:

$$\boldsymbol{\mathcal{L}}(\mathbf{w}) = \sum_{n=1}^{N}\left\{\ln\left[1+\exp(\mathbf{x}_n^\mathsf{T}\mathbf{w})\right] - y_n\mathbf{x}_n^\mathsf{T}\mathbf{w}\right\} + \frac{\lambda}{2}\|\mathbf{w}\|_2^2. \qquad (2)$$

The goal is to find $\mathbf{w}^\star$ that minimize the loss function $\mathcal{L}(\mathbf{w})$, i.e.

$$\mathbf{w}^\star = \text{argmin}_{\mathbf{w}}\mathcal{L}(\mathbf{w}). \qquad (3)$$

It can be proven that both (1) and (2) are convex in $\mathbf{w}$ [2], [3]. This means that $\mathbf{w}^\star$ can be found by solving the following equation:

$$\nabla\mathcal{L}(\mathbf{w}) = \mathbf{0}. \qquad (4)$$

However, for the *regularized logistic regression*, (4) has no closed-form solution and $\mathbf{w}^\star$ must be found using an iterative method (gradient descent or Newton's method). Such methods require sensitive calibration in the step-size parameter and their precision depends largely on the termination criterion. Moreover, they are usually computationally expensive. For these reasons, we chose to develop a model based on *ridge regression*. The closed-form solution of (4) for such a model is given by:

$$\mathbf{w}^\star = \left(\mathbf{X}^\mathsf{T}\mathbf{X} + (2N\lambda)\mathbf{I}\right)^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}. \qquad (5)$$

[1]The notation used in this document is largely based on the notation developed in [2] and [3].

D. Parameters selection

Once the model is chosen, one must tune its parameters for the benefit of accuracy. For this, 4-fold cross-validation was performed in order to obtain a good measure of the accuracy of the model with only one training data set. First, we tested several different degrees for the feature augmentation of each jet number subset, ranging from 1 to 15. We found that the best degrees were the following:

- PRI_jet_num = 0: $d = 9$
- PRI_jet_num = 1: $d = 10$
- PRI_jet_num = 2: $d = 9$
- PRI_jet_num = 3: $d = 10$

Once the degree was selected for each jet number subset, we tuned the lambdas in such a way that they limited overfitting and improved the accuracy of the model. Fig. 1 illustrates the 4-fold cross-validation plot obtained on the subset for which the jet number was 0.

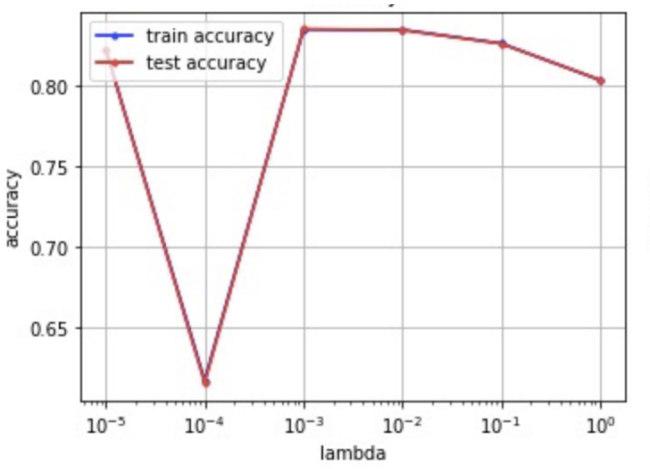

Figure 1. 4-fold cross validation for PRI_jet_num=0, d=9

One can see that $\lambda = 10^{-3}$ maximizes the accuracy of the model (smaller lambdas than the ones shown on Fig. 1 were also tested but they did not improve the accuracy). The optimal lambdas for the other subsets were found in a similar manner:

- PRI_jet_num = 0: $\lambda = 10^{-3}$
- PRI_jet_num = 1: $\lambda = 10^{-9}$
- PRI_jet_num = 2: $\lambda = 10^{-12}$
- PRI_jet_num = 3: $\lambda = 10^{-9}$

III. Results

Once the model and the parameters were chosen, the test data was separated into the 4 subsets corresponding to each jet number. Then, the predicted label y_n of the test data point $\mathbf{x}_n$ was the following:

$$y_n = \begin{cases} 1 & \text{if} \quad \mathbf{w}_i^\mathsf{T}\mathbf{x}_n \geq 0, \\ -1 & \text{if} \quad \mathbf{w}_i^\mathsf{T}\mathbf{x}_n < 0, \end{cases} \tag{6}$$

where i is the jet number of $\mathbf{x}_n$ and $\mathbf{w}_i$ is the set of weights that were obtained by training the model on the train data whose jet number is i. The accuracy of our submission on *AIcrowd* is 0.816 and our F1 score is 0.724.

IV. Discussion

As mentioned earlier, the choice of the model was largely influenced by our limited ability to efficiently implement an iterative method, due to the difficulty in choosing an adequate step-size parameter γ that ensured both convergence and an acceptable rate of convergence. This prevented us from implementing logistic regression techniques which theoretically, were conceived specifically for classification. The proper use of a logistic regression technique would be a key improvement and could be done via a reduction of dimensions. A method such as Principal Component Analysis which consists of identifying the principal combinations of features via a basis change would have been helpful. Unimportant combinations of features could have been eliminated altogether and would have rendered the use of a gradient descent much more feasible. On the other hand, selecting a simpler model allowed us to fine-tune its parameters more efficiently with the limited computational resources that were at our disposal. The accuracy of our final submission is not optimal but our model performs remarkably well, when taking into consideration its simplicity and the lack of proper feature engineering.

V. Summary

Our machine learning problem consisted of the three following steps: feature engineering, model selection and hyper-parameter tuning. The feature engineering aspect was paramount in extracting only the most meaningful information from our data. This step is vital and lays the foundation for any efficient use of a data set. When it comes to model selection, a deep understanding of the intricacies of each regression is imperative in order to select the most suitable one. Finally, the hyper-parameter tuning involves iterating through a wide range of values. From these iterations, the best hyper-parameter is chosen in order to maximize the model's performance.

References

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, "Learning to discover: the higgs boson machine learning challenge," 2014.

[2] M. E. Khan, R. Urbanke, and M. Jaggi, "Regularization: Ridge regression and lasso," *Machine Learning Course CS-433*, 2015, [Lecture Notes].

[3] ——, "Logistic regression," *Machine Learning Course CS-433*, 2015, [Lecture Notes].