CS-433 : Higgs Boson Recognition

Carruzzo Tristan - Dubien Victor - Preto Anne-Valérie

***Abstract*—In this project, the different linear regression methods, seen during the CS-433 course, were implemented for a classification problem ”The Higgs Boson Recognition”, using original data from CERN.[1] After pre-processing the data according to the important features, the 6 different linear regression methods were implemented. The focus was then shifted on Ridge Regression and Regularised Logistic Regression with gradient descent.**

I. Introduction

The Higgs Boson was discovered at the Large Hadron Collider at CERN.[2] This elementary particle explains why certain particles have a mass and why others don’t. The collision of two protons at high speed can rarely produce a Higgs boson. Physicist at CERN have developed a method to measure its decay signature. However, many decay signature are similar. Our goal is to predict whether the measurement is a result of a Higgs boson, a signal, or some other particle, a background.

Using the actual CERN data, we have applied different machine learning techniques to predict whether events were signals or backgrounds.

II. Models and Methods

A. Data Analysis

In order to implement a good classification, an analysis of the data is essential. The training data set contains 250’000 rows, with each row corresponding to a particle and 32 columns. The first 2 columns are the index and the labels, ’s’ for signal and ’b’ for background. To predict this, 30 columns are used in this machine learning process. They represent the different features.

The testing data set is similarly built, with almost 570’000 rows. The columns corresponding to the labels is also only filled with ’?’.

Of the 30 variables available for each event, 17 are primitive variables, 12 are algebraically calculated variables and 1 is estimated. They are all closely linked to variable number 22 ”PRI_jet_num” which takes values between 0 and 3. We decided to split our dataset in 4 separate subsets, one for each value of ”PRI_jet_num”, and to train a model for each subset.

We noticed that some values were fixed to ’-999’, those values are outside the normal range and cannot be computed. However, if we separate our dataset according to the value of the ”PRI_jet_num”, some variables can be removed since they are physically undefined. By doing this, the majority of the fixed variables at -999 were handled. Only in the first column, ”DER_mass_MMC” which estimates the mass m_H of the Higgs boson candidate, some values remain undefined. We would have to replace those values. Here I is a representation of the percentage of unclean values for this first feature.

	Subset 0	Subset 1	Subset 2	Subset 3
% of ’-999’ value	26.1	9.75	13.3	6.66

Table I
Percentage of unclean values

B. Mathematical Model

The labels column can be expressed as a binary column vector $y_{train} \in \mathbb{R}^N$ containing the values 1 for signal and -1 (or 0) for background. The features columns can be expressed as a matrix $X_{train} \in \mathbb{R}^{N\times D}$. Here D can vary from 30 if we decide to remove certain features or do feature expansion.

Our goal is to compute the most accurate weight vector $w \in \mathbb{R}^D$ that solves equation 1.

$$y_{train,i} = X_{train,i} \cdot w_i \tag{1}$$

$$i = 0, 1, 2, 3$$

Four different models, according to the value of the ”PRI_jet_ num” are implemented.

Different regression method were used to compute the weight vector w. In particular, we implemented : Linear regression using gradient descent - Linear regression using stochastic gradient descent - Least squares regression - Ridge regression - Logistic regression using gradient descent - Regularised logistic regression using gradient descent.

In order to quantify whether the model implemented was good, we measured its accuracy as well as its F1-score. These quantities are computed using equations 2 & 3.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{3}$$

with TP, TN, FP, FN corresponding to True Positive, True Negative, False Positive and False Negative.

Once the weight vector is obtained, we can multiply it with our matrix X_{test} to obtain a new vector y_{test}. Using the logistic function $\sigma(\eta) = \frac{e^\eta}{1+e^\eta}$ and setting the values to 1 if they are bigger than 0.5 and -1 otherwise, the vector y_{test} is converted to binary values.

C. Data processing

To begin testing our different models, we cleaned the data sets. We decided to remove the features containing '-999' when they made no sense. This operation gave use only 18 features to work with when $PRI\_jet\_num = 0$. Those 18 features were normalised and each model was tested.
The first improvement we decided to implement was K-fold cross-validation. With this, we would obtain K different weight vector. To compute our final vector w, we decided to average the weights whose accuracy was above a threshold value.
Another improvement was to replace the true undefined '-999' in the "DER_mass_MMC" feature. A simple method used, was to replace them with the mean value of the mass m_H in each subset.

III. Results

We started our analysis by calculating a first accuracy for each of the above mentioned methods. Here is a summary table of the obtained accuracies II

	LGD	LSGD	LS	RR	LRGD	RLRGD
Accuracy	0.701	0.665	0.702	0.704	0.744	0.752

Table II
First accuracy computed for each method

Based on these results, we decided to improve both Ridge Regression and Regularized logistic regression using GD.

A. Improvement of Ridge Regression

In order to improve this model, we decided to do feature expansion. For each subset we have defined a parameter p. This parameter is the power up to which we have raised each of the features, thus creating $(p-1) \cdot D$ new features. These parameters p were chosen through trial and error for each subset.
The second parameter on which we worked is the hyper-parameter λ. Through a grid search, we searched for the optimal parameter λ for each subset. Once it was found, we could compute the mean accuracy for each subset and finally compute a weighted mean for the whole data-set. Here III is a summary of the parameters obtained to improve our Ridge Regression.

	Subset 0	Subset 1	Subset 2	Subset 3
p	4	7	4	5
λ	1e-3	1e-16	1e-10	1e-8
Accuracy	0.760	0.726	0.806	0.710

Table III
Summary of parameters for Ridge Regression

The final accuracy of the model can now be computed through a weighted mean and the final value obtained is 0.754. We have improved the model from an accuracy of 0.704 to an accuracy of 0.754.

B. Improvement of Regularised Logistic Regression with gradient descent

The preprocessing was done in the same way as the previous parts.

The main idea behind this improvement is to use a more appropriate loss function : the regularised binary cross entropy loss 4, which is equal to 0 if the prediction is correct, and 1 otherwise.

$$\mathcal{L}_{BCE} = \frac{-1}{n}\sum_{i=1}^{n}(y_i \cdot \log \hat{y_i} + (1-y_i)\cdot \log(1-\hat{y_i})) + \lambda\|w\|_2^2 \quad (4)$$

Similarly to the improvement on ridge regression, hyper parameters were found through grid search. Moreover, a scheduler was implemented on the learning rate, in order to have better convergence : if the loss does not change more than 10^{-4} in absolute value in the next iteration, then the learning rate is multiplied by the "decay".
With these parameters, the best accuracy obtained was 0.797 (Submission 204789). Because of some randomness in the cross-validation, it is not guaranteed that another submission would give the exact same results.

	Subset 0	Subset 1	Subset 2	Subset 3
p	5	9	9	9
λ	1e-3	1e-4	1e-4	1e-4
Initial γ	1e-2	0.8	0.55	0.8
Decay for γ	0.995	0.9999	0.999	0.9999
Accuracy	0.829	0.776	0.782	0.778

Table IV
Summary of parameters for Regularised Logistic Regression

IV. Discussion

From the results seen above, some amelioration is still possible. The strength of our approach is to have separated our data set according to the variable "PRI_jet_num". This allowed us to create four different models.
Data processing is very important in this project. The feature expansion allowed us to significantly improve our accuracy. The grid search on hyperparameters such as γ,λ and p is also a key feature of our model.
To improve our model, a better NaN handling could have been done. The NaNs found in the first column could have been replaced by the k-Nearest Neighbor label. A search on outliers could also have been implemented in order to replace them accordingly.

V. Summary

In addition to the six basic methods, we tried to develop and improve ridge regression and regularised logistic regression with gradient descent, which gave the best results. Some further improvements could have been made, as mentioned above in the Discussion IV.

REFERENCES

[1] AIcrowd — EPFL Machine Learning Higgs — Challenges. [Online]. Available: https://www.aicrowd.com/challenges/epfl-machine-learning-higgs

[2] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, “The Higgs boson machine learning challenge,” p. 37.