Machine Learning - Project 1 : Higgs Boson

ML_Budget : Aubet Louise, Cadillon Alexandre & Hoggett Emma
EPFL, Switzerland

***Abstract*—This project consists in finding the Higgs Boson through the use of machine learning techniques on CERN's experimental data. After data set pre-processing, different methods such as least squares, ridge regression and logistic regression are applied, to predict if a Higgs boson is created. The best results are obtained with the ridge regression : a categorical accuracy of 0.809.**

I. Introduction

The Standard Model predicts the existence of a particle that gives mass to all elementary particles: the Higgs boson. This particle has been observed at the CERN in 2013, validate further this theory. The Large Hadron Collider (LHC) speeds up protons and makes them collide in order to produce the Higgs boson. The purpose of this Machine Learning project [1] is to foresee whether or not a Higgs boson is present in a given experiment using the available measured information.

II. Data pre-processing

The raw data set used for training is made of data points $\{(\vec{x}_i, y_i)\}_{i=1}^N$ where $N = 250000$, $x_i \in \mathbb{R}^D$ and $D = 30$. Each dimension of $\vec{x}$ represents a physical quantity measured during the experiment [2]. The raw data set is already employed to make an initial predictive model, yet the results are improvable by properly cleaning the data set.

A. Number of jets

When having a first look at the data given in the file `train.csv`, it is easily noticeable that a lot of -999 values appear. This involves undefined quantities.
Another measure stands out in column 23, `PRI_jet_num`, which corresponds to the number of jets in the collision. This feature takes integer values between 0 and 3. The value takes by this feature directly determines the presence of undetermined values -999 for many of the features.
For instance, in column 5, `DER _deltaeta _jet` corresponding to the absolute value of the pseudorapidity separation between two jets is unspecified if `PRI_jet_num` is ≤ 1, considering there are not enough jets.

This led us to split our data set based on the `PRI_jet_num` value. Once the data are split, an elimination is proceeded the columns for which all values are set to -999.

B. Mass of particles

To increase the accuracy of the model, the column containing the estimated mass of the Higgs boson must be taken into account. Indeed, some of the data points have an indeterminate mass value, which lead to a split of the four sub data sets based on whether the mass is defined or not.

C. Standardisation

Every feature of the data set are then standardised so they have zero mean and unit variance. The mean and variance of the training data features are used to standardise the test set. This avoids operations with values of a different order of magnitude and therefore limits the rounding errors that emerge for finite precision.

D. Outliers

After the standardisation of the data, training data points containing outlying values are removed. A threshold of ± 3 is exerted which corresponds to a confidence interval of 99.7% since the data is standardised.

E. Extended features

To avoid under-fitting, a polynomial expansion for every feature is performed. The goal is to find the best degree d_i for the polynomial expansion of feature x_i. More precisely, for each feature x_i, a 4-fold cross validation is performed for d_i ranging between 1 and 4. The maximum degree is set to $d = 4$ as the computation time increases drastically with an additional degree. A trade-off has to be made to fit the data optimally: if a low feature expansion degree is chosen, there is a risk of under fitting, although if it is large, there is an over fitting's risk.

III. Implementation

The goal of the implementation part is to compute the most accurate model. Each model, with its optimal parameters, is tested so as the error minimisation when predicting a result.

A. Least Squares

Firstly, an execution of the best linear regression of the form $y_n = \vec{x_n}^T \vec{w}$ is achieved. The least squares implementation is a direct method, with one single matrix inversion, still it has a high computing time for problems with many dimensions. The Mean Square Error (MSE) loss function is performed: $\mathcal{L}(\vec{w}) = \frac{1}{N} \sum_{n=1}^{N} (y_n - \vec{x_n}^T \vec{w})^2$.

B. Gradient descent

The gradient descent is an iterative method. It deals with the loss function gradient's $\nabla \mathcal{L}(\vec{w})$ with the purpose to make a step in the steepest descent direction's. A different version, the stochastic gradient descent is implemented too. In this method, the total gradient $\nabla \mathcal{L}(\vec{w})$ is replaced by the partial gradient $\nabla_n \mathcal{L}(\vec{w})$ of a random point in every iteration of the algorithm, such that $\nabla L(\vec{w}) = \frac{1}{N} \sum_{i=1}^{N} \nabla_n L(\vec{w})$. To apply this procedure a convergence criterion is needed, to bypass an explosion of the loss value.

C. Ridge regression

The ridge regression is a regularised least-squares method. $\mathcal{L}_{MSE}(\vec{w}) + \lambda||\vec{w}||^2$ has to be minimised, for finding an optimal $\vec{w}$ and λ. The second term favours simpler models with less features and therefore prevents over fitting the training data set.

D. Logistic regression

The Higgs boson problem has a binary output. The logistic regression is more adequate for this kind of issues rather than a linear regression that assumes that the output is continuous. Standardisation of the x_i is required. The logistic function is used to transform the known regression into a probability. This implementation uses the iterative gradient descent method. Finally, the regularised logistic regression adds a regularisation term with λ to the loss function of the logistic regression.

E. Optimisation

The function `split _data` separates the training set into two subsets, one for training (80%) and the other one for testing (20%). This allows an accurate estimation of predictions. For submissions, the model is trained on the whole data set.

The local cross-validation function randomly partitions the training set into four equal subsets. Every set is used for testing while the three others are used for training. An average of the test errors is then computed. This method is used in order to find an optimal value for λ and for the degrees in the polynomial feature expansion. The goal is to spot issues such as over-fitting or selection bias.

IV. Results

A. Least-squares

The first method applied is least-squares with no data pre-analysis. The obtained success rate is 0.745. By dividing the data into eight subsets depending on the number of jets and the mass and without normalisation, the rate increased to 0.750. When the features are standardised and extended to their optimal degrees, the result reaches 0.779. As predicted, standardisation and extended feature significantly improve the model.

B. Ridge Regression

Then, the ridge regression is implemented. Without any data pre-processing, the grade is only 0.774 whereas with all the pre-processing explained and $\lambda = 0.12$, the result is 0.806. This shows again the importance of cleaning the data.

C. Logistic regression

Finally, the logistic regression with data pre-processing and extended feature gives 0.747. The regularised logistic regression does not improve much the results as the grade is 0.748. However, adding the cross validation of lambdas and removing the outliers allow the result to go up to 0.76. The λ in the regularised logistic regression improve the results because large model weight are penalised.

V. Improvements

Here's some improvements that will refine the project's results.

A. Standardisation

The features standardisation is done by column, which assumes that each column is independent. It might not be the case and leads to a loss of information. The solution is to normalise the data all at once.

B. Feature expansion

A way to improve the feature expansion would be by crossing the features with each other, for example by creating a new feature $x_1 \cdot x_2$. This has a valuable impact on the results, if some features are related. A downside is that it makes the computational cost grow significantly, especially if we want to cross more than two features. Optimisation for this kind of feature expansion is more difficult than for the polynomial one.

C. Cross validation

During this project, a 4-fold cross validation is implemented. Nevertheless, it is common in Machine Learning to have a 10-fold cross validation to reduce variance. This increasing in the number of partitions increase the results, though it will take a longer time to compute.

D. Parameter λ of the Ridge Regression

In the ridge regression, the λ used is a global λ and isn't determined by cross-validation. Having a different λ by cross-validation for each subset should increase the score.

VI. Conclusion

The ridge regression gives better results than least-squares and logistic regression. This can be explained by the problem structure and the construction of the data pre-processing. This project is first and foremost an introduction to Machine Learning use in Physics problems. Moreover, it shows how important data pre-processing is essential before training a model.

The model suggested in this report can only be taken as a first approximation of the problem, since the best percentage of the prediction error is around 20%.

References

Sources verified on October 28, 2019:

[1] Martin Jaggi & Rudiger Urbanke, *Class Project 1*, EPFL

[2] C. Adam-Bourdario and al., *Learning to discover: the Higgs boson machine learning challenge*, July 2014