Machine Learning Project 1 (CS-433)

Nicola Ischia
Computational Science and Engineering
nicola.ischia@epfl.ch

Marion Perier
Life Sciences Engineering
marion.perier@epfl.ch

Leonardo Pollina
Life Sciences Engineering
leonardo.pollina@epfl.ch

I. Introduction

The first project of this course was based on the Higgs-Boson Machine Learning challenge. This challenge started in 2014 as an open data analysis competition on Kaggle.
The dataset comes from the CERN particle accelerator and it was used to catch the decay signature of the Higgs boson. The goal of the challenge was to identify signals corresponding to Higgs bosons decays and those corresponding to other particles or background events.
It results in a binary classification problem for which a various range of machine learning functions can be implemented to predict the class of each signal.

II. Data Analysis

The dataset was composed of $250'000$ samples, each sample corresponding to a decay signal characterized by 30 different features. Among them we can list: masses, momentum, energies and the number of jets. This last feature, the 22nd one, is an integer with value of 0, 1, 2 or larger. When the dataset is split in 3 subsets based on the 22nd feature value, some columns become constants. Thereby some features bring informations only for certain number of jets.
Some features were found to be correlated and the highest percentages of correlation between features are highlighted with yellow colors in Figure 1 (left). High correlations were also assessed by Principal Component Analysis (PCA), this is shown in Figure 1 (right).
Finally, a significant amount of -999 values are found in the dataset. This number indicates that the corresponding feature was impossible to obtain nor compute.

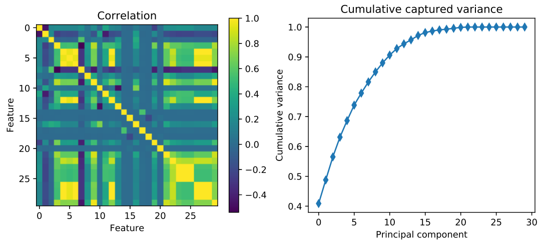

Figure 1. Correlation among features (left). Cumulative captured variance of principal components (right).

III. Methods

A. Preprocessing

The first issue to tackle was to replace - 999 values, called invalid values. Different approaches were implemented, such as replacing them either by 0 or by the mean of the valid values. In a more drastic way, we also tried to remove the columns (i.e. features) in which the percentage of invalid values was higher than an arbitrary threshold (in our case: 60%).
In order to obtain more consistent training dataset, our original dataset was divided in three subsets using the value of the 22nd feature (the number of jets). Notice that data was simply split in 0 jet, 1 jet and >1 jets, assuming the absence of discriminant features between 2 and more.
Since the range of values present in each feature differed a lot, standardization was applied in order to allow a correct comparison among features.
In addition, in the case of the dataset split in the jet-depending manner, a dimensionality reduction was performed. It deletes constant columns (i.e. columns with standard deviation = 0) and highly correlated (HC) columns. Two features were considered HC if their Pearson correlation was higher than 0.8.

B. Models

Least Squares Regression model was the first one we implemented. This model is based on the minimization of the MSE loss function. Both iterative and direct approaches were used. Note that the direct approach is the one using normal equations. For the iterative approach, Gradient Descent (GD) and Stochastic Gradient Descent (SGD) algorithms were applied. A polynomial expansion of features was performed to overcome the limits of linear regression. However, the use of an expanded training set for the Least Squares method could produce overfitting.

Ridge Regression was thus implemented based on Least Squares Regression. This model reduces the risk of overfitting by adding a regularization term to the MSE loss function. The regularization term is composed of the $||.||_2$ norm of the weights multiplied by a scalar, called lambda. Since the performance of such regression technique depends a lot on the degree of the polynomial expansion and on the lambda term, a 4-folds cross validation was applied. It allows the testing of severals models with different combinations of hyperparameters. Hyperparameters giving the best result were selected, see figure 2.

Because the expected output is binary, Logistic Regression was also implemented. It is based on the sigmoid function that remaps the predictions between 0 and 1. Convergence of the predictions was made possible with the choice of an iterative approach like GD.
To ensure stability, Regularized Logistic Regression was implemented. Again, the $||.||_2$ norm was chosen as a regularization term.

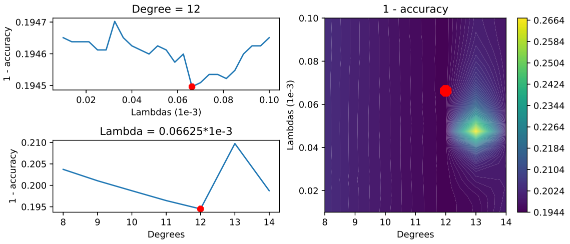

Figure 2. Best combination of degree and lambda based on accuracy for jet = 1 (Ridge Regression)

A 4-folds cross validation with GD was used on Logistic and Regularized Logistic Regressions, allowing the selection of the best polynomial expansion degree and the best lambda.

Since it was difficult to reach the convergence of the Logistic Regressor, the Newton′s method classifier was tested. Newton′s method is supposed to show a better convergence rate than Logistic Regression even if it has a higher computational cost. To balance the computational requirements, the training of this model was realized with a train/validation ratio of 75/25%, instead of a 4-fold cross validation.

C. Model Selection

Selection of the best combination of hyperparameters can rely on different loss measures (MSE, MAE, RMSE) or accuracy. The accuracy is defined as the number of good predictions over the total number of samples in the validation (or test) set. Notice that in regression problems we generally aim to minimize loss, while in classification ones the aim is to maximize the accuracy.

IV. Results

Table I indicates performances of all the tested models in terms of accuracy. The best accuracy was obtained with Ridge Regression model.
For each model, the dataset was preprocessed as following: conversion of -999 values into 0, division in three datasets based on the number of jets, standardization, deletion of constant columns and deletion of highly correlated columns.
For all the methods, polynomial feature augmentation has been used (without considering cross features).
Optimal degrees and penalization coefficients (λ) found through Grid Search for the Ridge Regression are shown in table II.

Regression Method	Kaggle score (%)
Least Squares	82,792
Ridge Regression	82,833
Logistic Regression	71,282
Reg. Logistic Regression (GD)	72,290
Reg. Logistic Regression (Newton)	82,262

Table I
Scores of different prediction methods.

Parameter	0 Jets	1 Jet	> 1 Jets
Lambda	$7.08625 \cdot 10^{-5}$	$6.625 \cdot 10^{-5}$	$8.425 \cdot 10^{-4}$
Degree	12	12	14

Table II
Best combination of hyperparameters per #jet, using Ridge.

V. Discussion

Some of the preprocessing steps have been essential to reach a good result. For instance, the values -999 are clearly out of range, therefore replacing them with 0 leads to an improvement of the result. Another step that has a very good impact on the final accuracy was to to split the data in three sets according to the number of jets then train the three models separately.
To save computational time, one could be interested in reducing the features dimensionality. For this competition this is not strictly necessary, considering that there is a much higher number of samples with respect to the number of features, even after feature augmentation.
Feature reduction techniques have been tried through correlation analysis and PCA. The latter was not a satisfactory strategy, since the results obtained on the same model with PCA are worse than the results obtained with all the features. On the other hand, removing highly correlated columns allows us to reduce dimensionality keeping the overall accuracy quite stable. For instance, in the case of Ridge Regression, the test set accuracy only loses roughly 0.15%.
Finally, a model comparison can be performed thanks to the results in table I. It is clear that best models are obtained via regularization, which prevents overfitting.
It can also be noticed that the best performing regression model is not a proper classifier. This might be explained by several reasons. At first, the tuning of the hyperparameters is easier with Ridge Regression. The weights are simply obtained by solving a linear system, and therefore it is faster than an iterative method. Moreover, it is not necessary to deal with the iteration step size. Therefore, there is one hyperparameter less to tune, leaving more time to refine the research of the remaining parameters.
From table I, it can also be seen that using the same logistic function, a better result is obtained with Newton w.r.t the standard Gradient Gescent. This is due to the slow convergence-rate of the Gradient Descent for the Logistic function, applied on this dataset. It prevents the model from reaching the real minimum of the loss function in an affordable time. Despite this improvement, the results are still worse than the ones obtained with Ridge Regression, since it was too demanding to properly tune the hyperparameters of the Logistic Regression using Newton′s method.

VI. Conclusion

Taking into account scores obtained and computational costs of methods such as Logistic Regression and Newton, our best model for this project is Ridge Regression. Our final score in the Kaggle competition was 82,833%.

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,” July 2014.