Machine Learning Project 1
CS-433

GUSSET Frédérick, SCANZI Jonathan and ZBINDEN Boris

I. Introduction

The aim of the project is to explore different methods to classify a dataset, in this case the Higgs Boson. The data is presented as a large csv file that needs to be parsed, then used to train our model, and finally classify a test set. Here we are trying to determine which method of classification, as well as data cleaning, gives the best results, since the right answers are known in advance and we are trying to get as many right classifications as possible.

II. Methods

A. Problem and Dataset

A brief study of the data set gives plenty information about the given problem. It is a binary classification problem as the output can have only two discrete values, described by 30 distinct features.

A naive approach is tested first in order to get a basis for comparison when using different methods, more complex models and advanced cleaning of the dataset. A simple way to achieve that is to use least squares to see the weight of all features. A comparison of the method is shown in the section III.

The most important part in all machine learning problem is to understand the given data and apply a correct data cleaning. The procedure used in this project is explained below.

B. Data Cleaning and Features Analysis

In order to get better results, we had to clean the data. The process involved loading the csv file containing the datasets separately in a Python script, then do two operations on both datasets (train and test): elimination of problematic values, and normalization.

By manual inspection of the files, it became clear that -999 (off values) was a magic number used to indicate a missing, or maybe erroneous measurement. Indeed, this number is present far too frequently, and usually out of place (most numbers are much closer to 0, and have a fractional part). We thus elected to replace these values with the mean of every column (i.e. every measurement family), such that it does not affect the learning and testing algorithms.

We also decided to normalize every columns, to have mean 0 and standard deviation 1. This way, every measurement feature has the same weight in our evaluation, which is reasonable, since we do not have any prior knowledge as to which measurement is the most, and least relevant to be able to classify the Higgs boson.

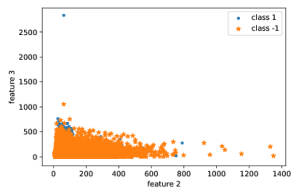

Figure 1. uncorrected distribution

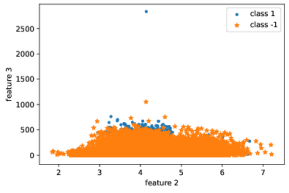

Figure 2. logarithmic transformation applied

The first tests were done with this configuration and additional treatment were applied later. The distribution of the data along most of the features revealed to be skewed. Thus, a logarithmic transformation is applied in a tentative to correct the distribution, as seen in figure 2. Finally, the outliers are removed and treated the same way as the off values.

All this preprocessing is applied the same way but separately for the train set and the test set, and a model has been tested with and without these operations to see the potential improvement.

Other possibilities that are tested in parallel is that the off values hold valuable information for the classification and should be taken into account as a new potential feature, or that features can influence each other.

C. Model Testing

The goal now is to find the best method to classify correctly the data. Several aspects such as the speed or the precision can be taken into account. But the best model is the one with a minimum loss that is still accurate, or general enough to predict correctly unseen data.

As our problem is a binary classification problem (two discrete outputs), several methods can be applied. A first one is linear regression, either by gradient descent or using least squares, even adding a regularization term to avoid complex models. Another one is binary classification using logistic regression. But as the data is clearly not linearly separable, simple method of classification will not give the best results. Thus the logistic regression, even regularized, does not give the best results.

D. Model Training

After all different methods are taken into consideration and tested to see which one gives the most satisfactory results, the goal is to train the chosen best model to fit our data. Thus, the hyper-parameters must be carefully chosen to ensure the best fit of the general data and avoid any over-fitting in the training set.

To do so, cross-validation is used, splitting the given train set in order to test our model, which is necessary to observe if the resulting model is correct enough or over-fitting. The main purpose of cross-validation is to use the whole dataset alternatively as train and test set. So when tuning the hyper parameters and observing which one gives the best output, we're more confident in our results than if we used a simple train-test validation. We used a 8-fold cross-validation, hence divided the dataset in 8 and at a time used one of this 8 parts as training data and the others as testing data. More fold cross-validation were considered as computationally too heavy, not worth for a too small improvement in the result.

III. Results

In order to compare all methods afterwards, a basic initial model is applied to the data. In this case, least squares is used to determined the weight of features and a basis loss. Cross-validation is used to determine the loss of a untrained test set, and verify the accuracy of the model, for this method and the next ones.

A. Difference between models

All six methods are measured with optimized hyper-parameters in order to compare them more efficiently. Furthermore, the data was neither standardize nor cleaned at this state. The results can be seen in figure 3.

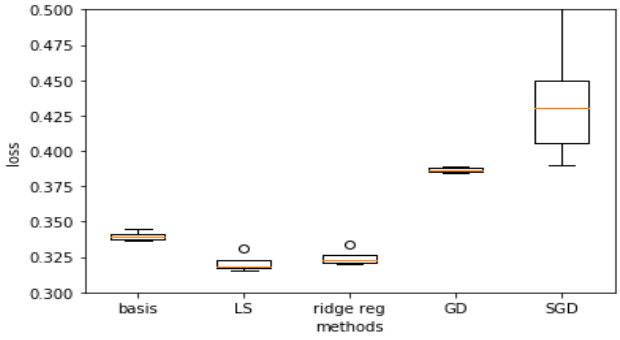

Figure 3. efficiency of methods

Gradient descent is not able to give better result and logistic regression, even regularized, could not achieve less than 0.5 loss. It is clear that the least squares method is the more efficient, and using the regularized method will bring better result.

B. Influence of data cleaning

Standardization and process described in section II-B is added to confirm the benefit provided. In the figure 4 these iterations are shown, proving the assumptions were correct.

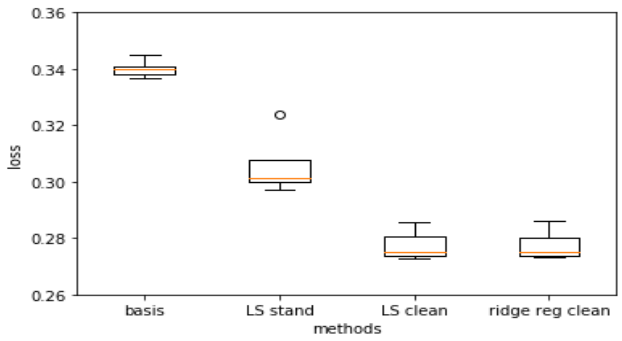

Figure 4. efficiency of data cleaning

C. Hyper-parameters and Cross-validation

To find the best hyper-parameters for the ridge regression (the degree of the model d and the weight of the regularization λ, a first cross-validation is run to find the degree of the linear model, using $\lambda = 0$. Then d is chosen as the element giving the lowest loss in the test set and an acceptable variance. The process is repeated for λ. Results can be seen in figure 5, where the loss of the testing set tends to increase when the model overfits.

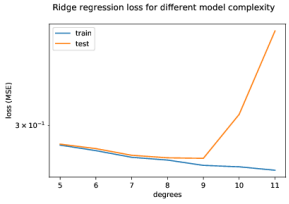

Figure 5. loss obtained through cross-validation for different degrees

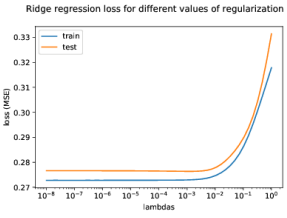

Figure 6. loss obtained through cross-validation for different λ

One can observe that the variance tends to increase with the complexity of the model. Before over-fitting, the variance grows up to 1.5 % of the mean, which is deemed acceptable.

D. Other

Finally, the addition of a new feature representing the strange values was tested using the same method as earlier. The cross-validation tests gave great results as the test loss was lower than before. But a final submission with another testing set proved that this feature increase the risk of over-fitting as the percentage of correct prediction was extremely low comparing to its expectation.

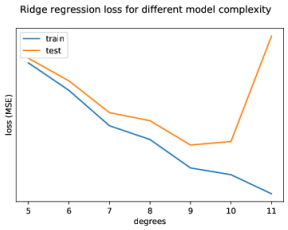

Figure 7. cross-validation over complexity of model

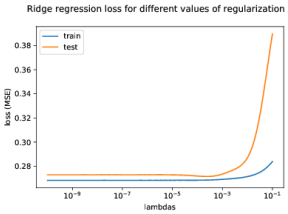

Figure 8. cross-validation over λ

In another hand, adding new features for the cross interaction of feature, $x_i \cdot x_j$ gave results with a greater loss and was quickly discarded.

IV. Discussion

Even though our problem is a classification one, the simple classification method gave not satisfying result in comparison of regression with regularization. It proves that the data is more complex and clearly not linearly separable. More advanced non-linear classifier must be used to improve further the results.

Proceeding by iteration to find the best hyper-parameters may not be the best choice as one can forget other important aspects. In this case a grid search over the space of hyper-parameters is more precise, but leading to high computation time. The first solution is a trade-off between the execution time and the accuracy.

In addition, all tests could have been run with different seeds, observing the variance. Only the most precise and accurate could show the best hyper-parameters. Indeed, choosing small variance results is better because a bigger variance may lead to models that tends to predicts in more hazardous way.