Machine Learning Project 1

Franck Dessimoz, Paul Jeha, Pierre Latécoère
CS-433 Machine Learning, EPFL, Switzerland

***Abstract*—The Higgs Boson is a particle in the Standard Model for which the mechanism has been proposed in 1964 by Peter Higgs. It's existence has been proved in 2013 after several experiments done with the Large Hadron Collider at CERN.**

In this paper, we will describe how we applied our knowledge in machine learning to build a classifier, using original data from the CERN.

I. Functions

We first implemented functions that are relevant in the building of the classifier:

- Linear regression using gradient descent
- Linear regression using stochastic gradient descent
- Least squares regression
- Ridge regression
- Logistic regression using gradient descent
- Regularized logistic regression using gradient descent

II. Models

A. First Model

As preprocessing step, we standardized the training and test features matrices using the following formulas:

$$std(x_{tr}) = \frac{x_{tr} - mean(x_{tr})}{norm(x_{tr})} \quad std(x_{te}) = \frac{x_{tr} - mean(x_{tr})}{norm(x_{tr})}$$

This standardization step is followed by appending a column of ones in front of both the training and test matrices.

We chose our algorithm to be Ridge Regression: Mean Square Error loss function with regularization. This is the learning algorithm that we used in our four models, improving it each time.

In order to increase the representation power of our linear model, we augmented the input using polynomial basis expansion.

To find the best hyperparameter λ, we used 4-fold cross-validation.

The following three plots illustrate the evolution of train and test error as a function of lambda, for basis expansions of degree 1, 2 and 3 respectively:

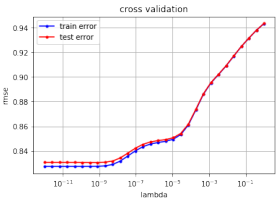

Fig. 1. Model 1, degree 1

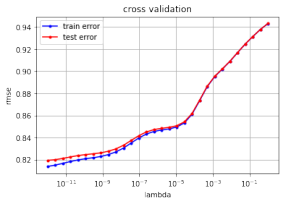

Fig. 2. Model 1, degree 2

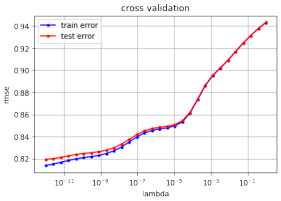

Fig. 3. Model 1, degree 3

We can see from these plots that the test error is smaller when increasing the polynomial basis degree from 1 to 2, but doesn't change when increasing the polynomial basis degree from 2 to 3. Therefore we chose this degree to be 2.

We can also see from these plots that the Mean Square Error is minimized when $\lambda = 10^{-12}$, therefore this is the lambda we chose.

B. Second Model

The improvement of this second model over the first one is that we take care of the columns containing -999 values.

These values correspond to missing entries in the dataset, i.e. N/A values. We defined a method that returns the indices of the columns with a number of -999 entries above a certain threshold (70% in our last version). We decided to remove these columns from both the train and test datasets.

After having removed these columns, there remained columns with undefined values. We thought of two ways to deal with these values:

1) replace them by the mean of the entries $\neq -999$
2) do a linear regression

We chose the second option and did a linear regression with least squares to estimate the values of the missing entries.

The rest of the methodology remains the same than in the first model: we still used Ridge Regression (MSE cost function with regularization), polynomial basis expansion and 4-fold cross-validation.

The following three plots illustrate the evolution of train and test error as a function of lambda, for basis expansions of degree 1, 2, 3 and 3 respectively:

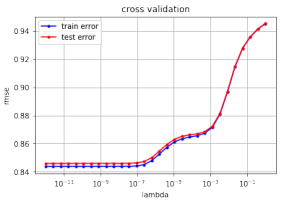

Fig. 4. Model 2, degree 1

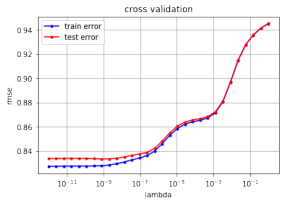

Fig. 5. Model 2, degree 2

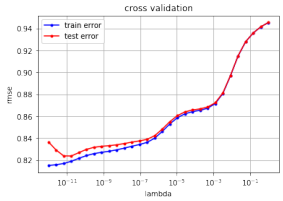

Fig. 6. Model 2, degree 3

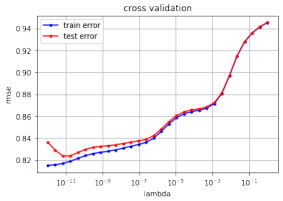

Fig. 7. Model 2, degree 4

We can see from these plots that the minimum Mean Square Error decreases when we increase the degree of the polynomial basis expansion from 1 to 2, and decreases again when we increase this degree from 2 to 3. But the minimum Mean Square Error doesn't change if we increase the degree of the polynomial basis expansion from 3 to 4. Therefore we chose the degree to be 3.

Looking at the third plot, the value of the hyperparameter lambda giving us the smallest Mean Square Error is $\lambda = 10^{-11}$, hence this is the lambda we chose.

After reflexion, we realized this model was not fully optimal since when deleting features with ratio of undefined values higher than 70%, we lose 30% of the information for each deleted features.

C. Third Model

This third model is based on the first model.

We realized that depending on the values of the `PRI_jet_num` feature, some columns contained only -999 entries. The `PRI_jet_num` column takes values in $\{0, 1, 2, 3\}$. We decided to divide the data set in three subsets: one for each value of the `PRI_jet_num` feature, knowing that values 2 and 3 of the `PRI_jet_num` feature lead to the same undefined columns.

We standardized separately the three subsets of features and appended a column of ones in front of each matrice.

We left the hyperparameters as in the first model: the degree of the polynomial expansion is 2 and the value of λ is still 10^{-12}.

D. Fourth Model

This is our most accurate model, obtained by combining the improvements from models 2 and 3.

We again separated the data set into three subsets depending on the value of the `PRI_jet_num` feature. For each of these three subsets, we were now able to delete the undefined features without any loss of information. We standardized the three subsets and appended the ones vectors in front of the three features matrices. In all three subsets, there were still undefined values which we predicted as in the second model, using least squares.

Using those three clean subsets, we found three models using ridge regression with the same parameters as in the second model: degree 3 and $\lambda = 10^{-11}$.

III. Results

Here is the accuracy of our four models, obtained after submitting them on Kaggle:

- First Model $(d = 2, \lambda = 10^{-12}) : 75.05\%$
- Second Model $(d = 3, \lambda = 10^{-11}) : 75.911\%$
- Third Model $(d = 2, \lambda = 10^{-12}) : 76.603\%$
- Fourth Model $(d = 3, \lambda = 10^{-11}) : 77.651\%$

IV. Discussion

At the end of this machine learning project, we brainstormed in order to see if any aspect of our models could have been improved.

First of all, when we separated the data set into three subsets in the models 3 and 4, we kept the same degree and lambda hyperpameters as the ones of models 1 and 2 respectively. Doing a cross-validation on each of the three subsets separately to find the optimal parameter for each of them would possibly have improved the accuracy of our results.

Secondly, we are not sure that the way we predicted the undefined values in the features matrices using least squares was the optimal solution.

Finally, we chose to use the ridge regression as the prediction function because the train predictions we were given in the dataset take values in $\{-1, 1\}$ instead of $\{0, 1\}$. As we are facing a classification problem, logistic regression would have been a more suitable technique. But we preferred to spend time on improving ridge regression as we are more experienced with this algorithm.

Taking into account that we used ridge regression which is not the best option for classification, we think that reaching an accuracy of 77.651% is quite good.