Machine Learning Project 1 report

Laurynas Lopata, Eloi Eynard, Aurélien Texier
EPFL, Switzerland

***Abstract*—Machine learning has been a field that has achieved astonishing results over the past two decades. A lot of it is based on deep learning and other advanced techniques. Yet even some elementary machine learning models can achieve excellent results. In this project, we evaluated multiple regression algorithms on a CERN dataset and analyzed how each one of them performs**

I. Introduction

In this project we are analyzing the dataset provided by CERN. In 2013 this dataset was used in the discovery of the Higgs boson, which was vital in understanding the world surrounding us. We emulate this discovery process by estimating the likelihood that a given event's signature is the result of a Higgs boson (signal) or some other process/particle (background). To do this we use four different methods, each of them is more complex

II. Models and Methods

The 4 different models we use are the following: Linear regression, Ridge regression, Logistic regression, Regularized logistic regression. We will describe each of them in more detail in this section.

A. Linear Regression

A linear regression problem can be solved in the following way: We need to find a linear function $f(x)$ for $x \in \mathbb{R}^d$ where d is the dimension of the data and $f(x)$ is defined in the following way: $f(x) = \sum_{i=1}^{d} w_i \cdot x_i$, such that the cost function $\mathscr{L}(w)$ would be minimized, we use the MSE loss function

To fit this function on the data we use the Least Square approach, which can be used in the following way to find the optimal w^*:

$$w^* = (X^T X)^{-1} X^T y$$

where X is the data matrix and y is the label vector

B. Ridge regression

In the ridge regression model we introduce an extra regularization term, which would prevent the model from becoming too complex and overfitting the training data. This new model can be desribed as minimzing a new loss function with respect to w $\mathscr{L}(w)_{ridge} = (\mathscr{L}(w) + \Omega(w))$. In our case we picked $\Omega(w) = \lambda \cdot ||w||^2$, where $||w||^2 = \sum_i w_i^2$ and $\mathscr{L}(w) = MSE$, to solve this optimization problem we used a closed-form solution, to find the optimal w^*

$$w^* = (X^T X + \lambda' I)^{-1} X^T y$$

where $\lambda' = \frac{\lambda}{2N}$, I is the identity matrix, X is the data matrix, y is the label vector

C. Logistic Regression

Logistic regression is a model that helps deal with nonuniform training data. In logistic regression we transform the output values of the prediction from an interval $[-\infty, \infty]$ to a new interval $[0, 1]$ which gives us a probability of the data point x having the label 0 or 1. In our implementation, we use the sigmoid function to do this mapping: $\delta(x) = \frac{e^x}{1+e^x}$

We define the loss function of this model in the following way:

$$\mathscr{L}(w) = \frac{1}{N} \sum_i ln(1 + exp(x_i^T w)) - y_i x_i w^T$$

Since this loss function does not have a closed-form solution to find the optimal w we use Gradient Decent for training, it is an iterative optimization algorithm with the update step defined as follows: $w^{t+1} = w^{(t)} - \gamma^{(t)} \nabla \mathscr{L}(w)$

D. Regularized logistic regression

The regularized logistic regression in extremely similar to ridge regression, just the loss function and the sigmoid are different. The new loss function with the added regularizer is defined as follows: $\mathscr{L}_{reg}(w) = \mathscr{L}(w) + \frac{\lambda}{2}||w||^2$

Gradient descent is used as the optimization algorithm, as described in subsection C

III. Results

We load the train and test data in a $.csv$ file. The data is composed of 30 different features and a binary label $y \in \{-1, 1\}$. We start by normalizing the data matrix X. Features that are measured at different scales do not contribute equally to the model fitting and the function that the model learns and might end up having a significant amount of bias. To deal with this potential problem feature-wise standardized with parameters ($\mu = 0, \sigma = 1$) is usually used prior to model fitting.

A. Cleaning the data

As we analyzed the train and test data, we observed that the feature PRI_jet_num impacted the distribution of the rest of the data, causing different values to be invalid (=-999). To mitigate these issues we did the following.

- We started by standardizing the whole dataset, leaving invalid values untouched, and not using them in the calculations of the mean and std.
- We then proceed by splitting the train dataset into 4, with respect to the PRI_jet_num, keeping the y values mapped to the same indices as the tx.

- For each of these jet subsets, we removed the features which contained exclusively invalid values. The number of features removed varied according to the the PRI_jet_num as shown below:

jet_num	Removed features
0	[4 5 6 12 23 24 25 26 27 28]
1	[4 5 6 12 26 27 28]
2	[]
3	[]

- We then set the remaining invalid values to the mean of the corresponding feature in the given jet set. Setting them to 0 gave approximately the same performance but slightly worse so we decided to keep them as the mean.

B. The four machine learning regression methods and their results

We implement the four basic machine learning methods and try to get the best accuracy for each of them. We split the data set into a training set and testing set respectively 80% and 20%. For methods which contain hyper-parameter like λ, γ and the degree m for the polynomial expansion we have to find the best value possible. That's why we implement cross-validation with a grid search method. To have a reasonable time we use k-fold = 5.

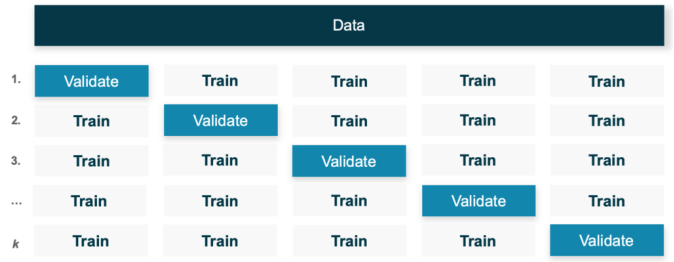
Model	Train accuracy	Test accuracy
Linear Regression (GD)	71%	70.3%
Ridge Regression	80.9%	80.6%
Logistic Regression	78.8%	77.4%
Reg Logistic Regression	79.3%	79.1%

C. Our best model: Ridge Regression

A model which gives us a much better accuracy is the Ridge Regression model. In this model, there is a hyper-parameter λ that we have to optimize to get the best accuracy possible. Firstly we clean the data as describe in III.A. So split the data in 4 jet set according to their feature $[0, 1, 2, 3]$. Then we deal with the -999 value. We replace them by the mean of the feature among all the sample. Once the cleaning data done, we expand our model to a non-linear model we modify our training set using feature expansion method. We augment the data in the following way

$$[x_{1,n}, ..., x_{d,n}] \; to \; [x_{1,n}, x_{1,n}^2, ..., x_{1,n}^m, ..., x_{d,n}, x_{d,n}^2, ..., x_{d,n}^m]$$

where m is the degree of the expansion.
We have to optimize λ and m. This is why we implement the k-fold cross-validation to improve our method. K-fold cross-validation is one ways to improve the holdout method. This method guarantees that the score of our model does not depend on the way we picked the train and test set. The data set is divided into k number of subsets and the holdout method is repeated k number of times.



The larger k is, the longer it will take to compute the algorithm. We take $k = 5$. The idea is to train our ridge regression method with a set of different λ and different degrees m to find the best $w*$ possible for each jet set. That means we have possibly four different λ and degree m to minimize the error for each jet set. We set the lambda interval to be $\lambda \in [1e^{-4}, 1]$ with 20 equally distributed lambda value and $m \in [2, 7]$. The test and train RMSE is calculated for each λ and each degree in order to find the best hyperparameter.

Jet set number	best λ	best degree m
0	0.0001	3
1	1	5
2	1	3
3	0.72789538	4

The final test accuracy sent to the competition arena AIcrowd Higgs boson challenge give us 80.6 % of accuracy

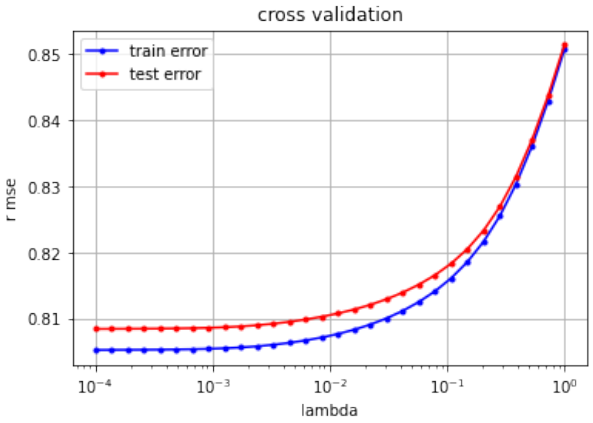


Fig. 1. Jet number 0 : Cross validation through the lambdas for the best degree $m = 3$.

IV. Conclusion and Discussions

The improvement provided by the processing of the data is significant in a limited way. In our coding venture, we noticed that some parameters (lambda, k_fold) could be tuned more finely but at an expense in computation time and coding complexity. The choice of model however, provided the most potential for improvement. Our final result is satisfying given the tools and time at our disposition, but could be improved even further with more thorough data cleaning, specific algorithms and time.