Project-I by Group 28: B. pseudomallei

Juraj Korcek & *Christian Tresch* & *Pierre Colombo*

November 6, 2016

Abstract

In this report we summarize our findings for the Project-I done for Pattern Classification and Machine Learning course taking place during winter semester 2016 at EPFL. The aim of this project is to classify a given data set from CERN regarding Higgs boson detection in collider experiments. The data consists of logical output variable $\mathbf{y}$ and 30 input (independent) variables $\mathbf{X}$ with about $N = 250000$ data samples. First, we have observed and analysed the set in order to get information about features that would help us classify the data the best. Our observations lead us to the first simple model. Aferwards, we have built complex model using the Penalized Logistic Regression with Gradient Descent algorithm and cross validation method to avoid overfitting. For simple model we have obtained 79.67% accuracy on half of the test data, while with more complex model we have achieved accuracy of 82.47%.

1 Introduction

In this paper we describe our thought process that lead us to the solution. The exploratory data analysis phase has been the most important phase of our project. Our objective has been to gather as much information as possible from the data set and thus we have spent a significant amount of time analysing its structure. The findings from this analysis oriented have driven our strategy.

As a result of the analysis we have constructed the first *simple model* consisting of only two features. Then our observations have lead us to define complex modell consisting of 6 different disjoint submodels. By disjoint we mean that each of the submodels work with it is own exclusive part of data set. These subsets of the dataset are mutually exclusive and collectively exhaustive (thus, these models do not need to vote to determine the output for given datapoint). This has been done to account for various configurations of missing values, as removing datapoints with missing values would result in huge information loss.

2 Choice of algorithm

We basically have implemented the following algorithms :

- linear regression with stochastic and batch gradient descent
- least squares
- ridge regression
- normal and penalized logistic regression

The linear regression, least squares and ridge regression algorithms are not suited for our problem because they are regression algorithms while our problem is of classification nature. Therefore, we have turned to logistic regression. We have decided to use penalized logistic regression in particular as it allows us to avvoid overfitting by proper use of the hyperparameter λ and also takes care of a potential ill-conditioning.

3 First model

3.1 Data visualization and cleaning

We have performed basic exploratory data analysis on our data and cleaned it. We have plotted a histogram (not included in the report) for each input variable. Additionally, we have plotted scatter plots of each input variable against each input variable given the output. Thanks to that, by plotting X(1) in function of X(2), we have found that data can be split well by these two variables (see Fig. 1).

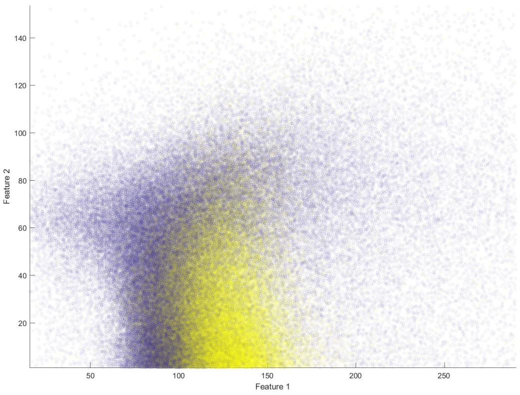

Fig. 1: Features 1 vs 2 (blue for background, yellow for Higgs)

3.2 First Model

BAsed on the above-mentioned finding we have built our first model: if the first feature of x is between 105 and 170 and the second feature is below 55 $allchosenbyvisualanalysis$ we predict $+1$ otherwise we predict -1 .
With this model we have achieved 79.67% of good predictions on the test set very fast. This is very simple model which has quite considerable performance.

4 Second model & Improvements

In this section we explain structure of our second, complex model using Penalized Logistic Regression.

4.1 Model definition

First we wanted to drop all collumns with missing or wrong value. However, that would not have been a good idea, because we would lose too much information. Thus, we have decided to keep all data by designing a model which includes all the information we have. After exploratory data analysis we came up with 6 disjoint models. Then we have noticed:

- that there are 7 columns (in yellow) which have -999 values $18 * 10^4$ times.
- that there are 4 columns (in blue) which have -999 values $10 * 10^4$ times. The set of points with -999 is a subset of those mentioned in the previous point.
- one column has categorical values 0,1,2,3. We have use ddummy coding to split this column in three columns, each one containg a logical value (if all collumns are 0 it means that this point belongs to the fourth category).
- the first column (in red) also contains -999 values, however, they are not subset of the others.

We get three submodels as shown in the figure 2 below. Each of these submodels is split into two submodels according to the value of first input variable X(1). If X(1) is different from -999 we have model iA; else we have model iB. This way we obtain final 6 submodels.
NB: These six models are disjoint, a data point belongs exclusively to one of these six.

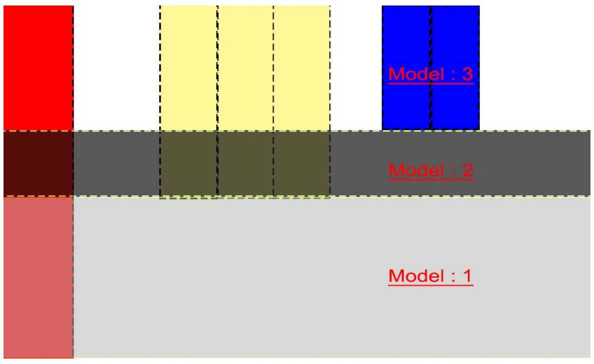

Fig. 2: Data Observations & Models Definition

4.2 Feature transformations

We have tried to train models on transformations by polynomial basis of degree 1 and 2. Because of computation time we have not been able to run it with degree 3.
For the polynomial basis transformation we also added the cross terms.
We normalized all the data except for the categorical ones.

4.3 Penalized Logistic Regression

Logistic Regression is a classification algorithm. It uses the maximum likelihood in order to find the best model which will fit the given data set correctly.

Different approaches are available to perform Logistic Regression: The Gradient Descent and the Newton Method. However, in the frame of this project, we have used only the Gradient Descent algorithm as the Newton's method is not suitable for huge datasets due to exponential computational complexity.

One parameter is extremely important in Gradient Descent. It is the descent rate *stepsize* γ. It determines how fast our gradient converges and, therefore, the quality of our predictions. We had to tweak it to get relevant results. We have found that stepsize decrease of gamma / sqrt(i + 1) where i is the number of iteration helps the GD converage relatively fast.
In order to solve the model selection problem we have used the cross-validation. After some trial and error we have come up with following parameters::

- We have tried to run the test with 4 to 10 K-folds and the result has remained the same up to a little variance. That is why have sticked with the number of K-folds being 4.
- The maximal number of iterations is 1000.
- The feature transformation will be polynomial of degree 2.

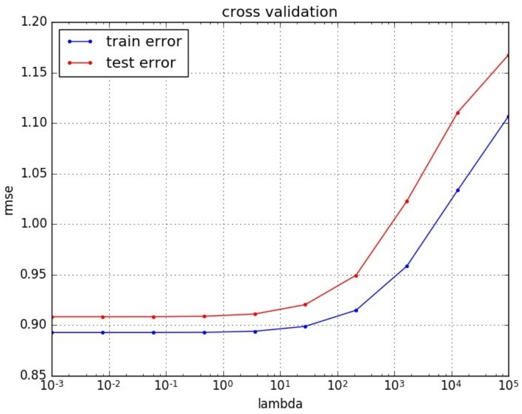

Fig. 3: Cross validation results evaluating model 2

For the feasibility of computation we have chosen the degree of 2. For this degree the best lambda obtained is 0.

4.4 Results on the training set

We finally gather our results in a table :

Model Name	Accuracy (%)
Model 1	82.92%
Model 2	93.85%
Model 3	78.46%
Model 4	92.60%
Model 5	80.61%
Model 6	95.08%
Total model	82.75%

4.5 Results on the test set

Finally by predicting the data set with our algorithm consisting of the 6 submodels we get a result of : 82.47%.

5 Summary

The Penalized Logistic Regression using Gradient Descent is an algorithm which performs very well for given problem. Our models are able to converge and give us a solution with an acceptable accuracy.