Event Classification Using Various Machine Learning Regression Techniques

Jean Gillain
sciper no. 331411

David Desboeufs
sciper no. 287441

Mathieu Caboche
sciper no. 282182

***Abstract*—This project aims at classifying events recorded at CERN. This is achieved by implementing various machine learning regression techniques and eventually using them to create a good model. The project also required us to process the given data in order to achieve the best possible results.**

I. Introduction

In this report we will give a brief but detailed explanation of the thought and development process we went through in order to train a model enabling classification of a physical event. The assigned goal was to create a model capable of classifying data records from CERN as occurrences or non-occurrences of the Higgs boson.

Section II describes the given data set in more detail. Over the course of this project we used several different regression techniques learned during the course. In order to find the best possible model we then used one of the implemented regressions with pre-processed data. An important part of creating a good model is to have a good training data set. That is why data analysis and cleaning was a major part of this project as well. The methods used to analyze and clean our data sets, as well as the steps we went through to find the best regression method are described in section III. Finally, we will compare the models obtained by various regression techniques to our best performing implementation in section IV.

II. The Higgs boson Data Set

The given data set for model training consists of some 250000 records which classify an event as a Higgs boson occurrence or not. The events are recordings of 30 features. The testing data set is identical except that classification is left up to our model.

Let us describe the training data in more detail so as to highlight the importance of analyzing and subsequently cleaning the latter. A look at the records quickly reveals that the value -999 appears very frequently and for a number of different features and records. This value could thus be interpreted as an error of measurement. Furthermore we can notice that some features take on only very few different values while others almost never repeat the same value twice. This raises the question of feature importance, i.e. which features impact the classification a lot and which do not (or even lead to misclassification).

Finally we noticed that the $22nd$ feature only takes on 4 different values. One could interpret this feature as an indicator of which machine was used to obtain the results for a given measurement. Section III describes how we decided to clean the data set with respect to the above mentioned concerns.

III. Models and Methods

As specified by the project guidelines we implemented the required regression techniques. We will not go into the details of their implementation. Rather we will describe how we tried to obtain a best performing model and with which solution we eventually did get the best results. We shall also cover how we cleansed the data sets and for what reasons.

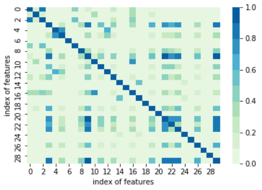

Figure 1: Correlation matrix of all features

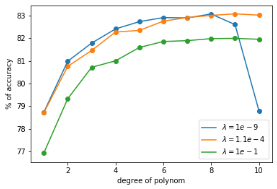

Figure 2: Accuracies obtained for different polynomial extension degrees and λ values

A. *Finding the Best Regression Technique*

The regression which we decided to use in order to get the best model is the ridge regression. This decision is based on the fact that ridge regression performed best among all of our implemented regressions, as shown in table I.

To further improve on the performance of ridge regression we cleaned the input data and used feature extension. Section III-B describes the steps we took to clean the data. Let us describe a few feature extensions that we implemented and used.

Our initial intuition was that features with large variance should impact the resulting model more than features with low variance. Thus we decided to use augmented feature vectors on the input data according to feature variance. We first sort the features in increasing order of their variance. The ordered features are then subdivided into equally sized groups. The number (and size) of groups is a parameter we are free to adjust.

Finally we extend the feature space by building polynomials of higher exponent for groups of higher variance. The result are augmented feature vectors with say 5 different polynomials (if we have 5 groups).

As it turns out, the previously described implementation did not provide an actual improvement over classical ridge regression on the raw (non cleaned) data. That is why we implemented a different feature extension. We first apply the inverse logarithm function (1) and then a fixed polynomial extension of degree 9.

$$f(x) = \log \frac{1}{1+x} \tag{1}$$

Although this feature extension gave us our best possible results when cross-validating, we were unable to apply the same feature extension on the test set, yielding this implementation unusable for the purposes of this project.

In fact, the nature of the inverse logarithm function forces us to discard features that contain values which make the inverse logarithm tend to infinity. Since the features removed on the train set and the test set may differ, we cannot use the inverse logarithm function.

In the end, using ridge regression with polynomial feature extension of degree 8 (for all features) proved to yield the best valid results. In order to find the best parameters (λ and degree) for our ridge regression, we simply used a grid search algorithm on possible parameter values. Figure 2 shows why we determined 8 to be the best polynomial degree to use.

B. Data Processing

Let us briefly go over some data cleaning that did not prove effective in improving model performance.

- Remove features that contain over a given percentage of error values (value -999).
- After applying the previous point, remove lines (measurements) that still contain error values.
- Remove features that contain few different values (frequency analysis).
- Standardize all features. This step is the last step of the data cleaning.

Here are the data cleaning steps that did prove useful and which we actually use in our best performance regression.

- Replace all error values by the mean of the feature in which they appear. We do not consider error values for mean calculation.
- Compute the correlation matrix between all features and keep only one feature in cases where multiple features are strongly correlated (more than 90% in our case). The correlation matrix is shown in figure 1.
- Split the data into 4 parts according to the 4 different values taken on by the $22nd$ feature, PRI_jet_num. Compute the weights per data part and use the corresponding weights when classifying a measurement (discriminate using the $22nd$ feature).

C. Model Validation

To validate our obtained models we used cross-validation using the given training data set. We have to apply all feature extensions and data cleaning steps on the testing data part, just as we do on the training data part.

We used 10-fold cross-validation to assert the accuracy of all of our implementations.

IV. Results

In this section we analyze our best performing regression (respectively the model obtained by the latter) to the models obtained using our implementations of other regression techniques.

To compare the performance of the models resulting from our various regression implementations we can simply compare the model accuracy. I.e. we check what percentage of a known classified set is correctly classified by our model, as described in section III-C. The model accuracies obtained by our various regressions are listed in table I.

The regression technique denoted as *Our Best Regression* consists of all useful data cleaning and processing steps listed in section III-B, as well as polynomial extension of degree 8 on ridge regression.

Note that all regressions are run on the raw, uncleaned and unprocessed data except for *Our Best Regression*. The only exception being the fact that we use polynomial feature extension where specified.

Regression Technique	**Accuracy (Avg.) - STD**
Least Squares GD (and SGD)	68.5% - 0.20%
Least Squares	74.4% - 0.25%
Ridge Regression	74.4% - 0.24%
Logistic Regression	65.7% - 0.29%
Newton Logistic Regression	65.7% - 0.29%
Our Best Regression	81.3% - 0.82%

Table I: Accuracy of Various Models Obtained by Our Regressions

It is worth mentioning that we obtained a cross-validation accuracy mean of about 83% using the inverse logarithm and polynomial with degree 9 feature extension on ridge regression with the aforementioned useful data processing (cf. section III-B). As mentioned in section III-A this result is however irrelevant for this project.

V. Conclusion

This project allowed us to experiment with multiple regression techniques and we learned the importance of good data processing for machine learning problems. During project development we went through many trials in order to build the best possible model. We concluded that proper data analysis and processing played a major role and greatly helped us to find better results.