Machine Learning Project I

Vincent Micheli, Ivan Schonenberger, Lionel Constantin
EPFL, Switzerland

I. Introduction

The objective of this project is to familiarize ourselves with Machine Learning concepts and methods. We first implement the different algorithms seen in class and then use and compare them on a real dataset.

II. The Data and Exploratory Data Analysis

A. The Data

For this project we are given a dataset produced by the ATLAS full detector simulator from CERN and published in Mars 2013. Each data point consists of 30 features and represent a signature event from which we want to infer whether or not a Higgs Boson was created. The data is split into a labeled training set of 568238 observations and an unlabeled testing set of 250000 observations. It should also be noted that a feature with a value of -999 represents a missing value.

B. Exploratory Data Analysis

We first investigate more closely the missing values and find that the presence or not of a value depends on the value of the PRI jet num feature in the same observation.

More specifically if $PRI.jet.num = 0$ then DER delta eta jet jet, DER mass jet jet, DER prodeta jet jet, DER lep eta centrality, PRI jet leading pt, PRI jet leading eta, PRI jet leading phi, PRI jet subleading pt, PRI jet subleading eta, PRI jet subleading phi and PRI jet all pt do not contain any value.

If $PRI.jet.num = 1$ then DER deltaeta jet jet, DER mass jet jet, DER prodeta jet jet, DER lep eta centrality, PRI jet subleading pt, PRI jet subleading eta, PRI jet subleading phi do not contain any value.

Finally if $PRI.jet.num = 2$ or $PRI.jet.num = 3$ all the features above have a corresponding value.

Moreover DER.mass.MMC might contain missing values independently of the value of PRI.jet.num.

We decide to set missing values at 0 instead of -999. We suppose this will reduce their impact on computations.

Next we look at the correlation matrix of the features and we notice that many covariates are correlated which may result in an ill-conditioned design matrix and cause numerical problems when running our algorithms, as well as increase the variance of our predictions. One solution to this might be to add a ridge parameter to the regression.

III. Models and Methods

In order to compare the models that we build, we resort to 80/20 cross validation with RMSE (root mean squared error). That is we split 80% of the labeled data to form our training set and 20% to form our local validation set. The points are randomly allocated to either one of the sets according to the proportion prescribed above. Then at each step of model selection and/or building, we train our model on the training set and use the parameters obtained to compute the RMSE on the local validation set. We then use the metric obtained to compare our models.

The first model we fit to our data is simple. We run a classical least squares regression on the training set.

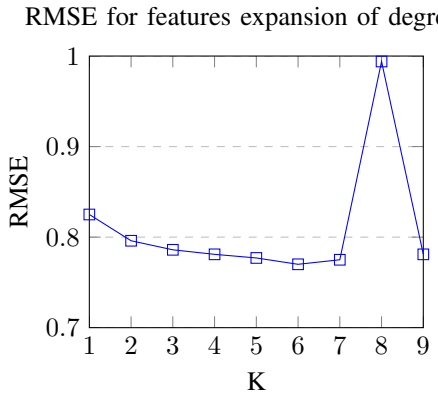
We then try to add new features to the data. Namely we try to expand each original covariate into a polynomial basis of degree K. To determine the optimal hyperparameter K we run a least squares regression for each possible K up to $K = 9$ and select the K for which the cross validation RMSE score is the lowest.

Finally, on top of feature expansion we try to address the multicollinearity in the data which was pointed out earlier. One natural way to approach this is ridge regression. There is yet another hyperparameter to select : λ. We employ a similar technique as for the polynomial bases: we fit a ridge regression with parameter λ for each possible λ from $\lambda = 1$ down to $\lambda = 10^{-15}$ (we select 50 points on a logarithmic grid) and select the λ for which the cross validation RMSE score is the lowest.

IV. Results

Our first model yielded a cross validated RMSE of 0.825.

Our second model yielded a polynomial basis expansion of degree $K = 6$ with a cross validated RMSE of 0.770. This represents a net gain from the original design matrix. We could explain this difference from the fact that the signal's nature depends on higher degrees of the features recorded.



Our third model yielded a ridge parameter $\lambda = 7.54\text{e}{-07}$ with a cross validated RMSE of 0.767. This represents a slight gain from ordinary least squares. Moreover, a small value of lambda suggests that very little regularization was necessary.

We fitted the test data set with our last model and obtained a score of 80% prediction accuracy on the Kaggle platform.

V. Discussion

One could argue that linear regression is not the best way to go about this classification problem since the metric we used, i.e RMSE, is usually not the best fit for this class of problem. However when fitting the data with a logistic regression model we did not achieve a better prediction accuracy.

Besides, more feature preprocessing could be done before attempting to fit any model. One could try to get rid of multicollinearity between covariates, drop irrelevant covariates and/or transform covariates to better align with the assumptions of the models he is employing.

As explained in section II-B, the data can be partitioned in subsets depending on the value of PRI.jet.num. We established a model that generated 4 different models depending on the value of this variable. Each model had the missing columns removed from its data. The predictions were calculated each one with its corresponding model. However partitioning the data did not significantly increase prediction accuracy.

VI. Conclusion

In this project, we implemented the requested Machine Learning algorithms and applied these methods to the Higgs Boson Challenge data set. We saw that cross validation, features expansion, regularization and hyperparameters tuning were paramount to improve our models. Our best model achieved a score of 80% prediction accuracy on the Kaggle platform.