Machine Learning Project I Report

Joaquim Campos
joaquim.campos@epfl.ch

Simão Sarmento
simao.moraessarmento@epfl.ch

Rodrigo Bernardo
rodrigo.moreirabernardo@epfl.ch

***Abstract*—The problem that was given to us was that of using binary classification techniques used in class in order to analyze a real-world dataset. That dataset was from CERN particle accelerator experiments regarding the Higgs boson.**

To that end we: performed exploratory data analysis, data cleaning, feature processing and feature selection; used and compared concepts learnt in class, like gradient descent, stochastic gradient descent, least squares, ridge regression, cross-validation and logistic regression; slightly modified the method of augmented feature vectors, in order to support better our feature processing/selection; analyzed our models, tested them using cross-validation, and compared them.

I. Introduction

Given the dataset from CERN, we should estimate the probability that a certain event was either the result of the Higgs boson or of noise.

For this project we applied some of the methods from the toolbox that we implemented for step 2 of the project. After analyzing and preprocessing the dataset, we derived a model based on ridge regression and augmented feature vectors. We describe it in Section II.

We find that our model is a better fit for this dataset than the models we derived for the other methods we tried. We report our experiments in Section III and our results in Section IV.

II. Models and Methods

We started by directly applying every method from step 2 of the project to the dataset, apart from logistic regression, namely: gradient descent, stochastic gradient descent, least squares and ridge regression with normal equations. However, our exploratory data analysis revealed some structure of the data that we could exploit. We describe the data analysis in Subsections II-A and II-B. The derived model to exploit that structure is explained in Subsection II-C.

A. Data Analysis

In order to get find some intrinsic structure of the dataset that we could exploit, we did some exploratory data analysis.

Some data points had meaningless values (equal to -999 in the dataset; we will call them MVs from now on, for short). By inspection, we found out that some features had an unusually high ammount of MVs. We grouped the features by their percentage of MVs. We obtained three non-overlapping groups with large quantities of MVs: group 1 (one feature), with around 15% of MVs, group 2 (eleven features), with around 40%, and group 3 (three features), with around 70%. No other features had MVs.

Also by inspection, we found out that the column with name "PRI_jet_num" categorized the data points in four non-overlapping sets (sets 0, 1, 2 and 3). These had direct correspondence to the percentage of MVs for the features in the data points of these categories. In particular, set 0 encompassed all MVs from group 2. Moreover, it encompassed the same ammount of MVs from group 3. Set 1 encompassed all other MVs from group 2 that were not encompassed by set 0, i.e., around 30% of the MVs of group 2. Sets 2 and 3 did not have any MV from groups 2 and 3. Finally, group 1 had MVs scattered around every set.

We also categorized our features according to their "importance". To that end, we used mutual information as a measure of importance. Intuitively, this choice is justified by the fact that mutual information is a measure of the mutual dependence of two random variables and so, if a label is highly dependent of some feature, then that feature should be "important".

B. Data Cleaning

Ordering the features by importance, we verified that the features from groups 2 and 3 were not important enough, given the ammount of missing information. With that, we decided to remove these features from sets 0 and 1.

On the other hand, the feature from group 1 and the label random variable had high mutual information, so it was not removed from any set. We replaced the MVs with the mean of that feature over all other data points.

For training our model, we removed the outliers. For each feature we considered to be an outlier every data point whose value for that feature differed from the mean by more than three times the variance.

Finally, as a matter of convenience, we decided to merge sets 2 and 3, as they both only had MVs on the feature from group 1.

C. Derived Model

We soon realized that it do not make sense to use gradient descent or stochastic gradient descent, over least squares or ridge regression. In particular, as we used the technique of augmented feature vectors for our model, we built it on top of ridge regression, to try to avoid overfitting.

Given a set and the ordering of the features by importance for that set, we built an augmented feature vector by constructing a polynomial of arbitrary degree d. The procedure was the following. Suppose the features are numbered, by order of importance, from 1 to the number of features (lets call it m) we still have in this set, i.e., that were not removed in the data cleaning process. For the n best features we constructed a feature vector whose components are the monomials of the multinomials of degrees 1 to d, $\sum_{i=1}^{n} x_i, (\sum_{i=1}^{n} x_i)^2 \ldots (\sum_{i=1}^{n} x_i)^d$. For the other $m - n$ features, we simply added a polynomial basis of degree d.

For example, suppose $n = 2$, $m = 3$ and $d = 3$. Thus, our n most important features would be features x_1 and x_2. Then our extended feature vector would be

$$(1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_3, x_3^2, x_3^3).$$

III. Experiments

Our hyperparameters are n, the number of "best" features, d the degree of the polynomial, and λ, the regularization factor. To tune them we tried values of n from 1 to 10, values of d from 1 to 7, and 25 values of lambda from 10^-5 to 1 in a logarithmic scale of base 10 (by using numpy's command logspace(-5, 0, 25)).

We trained and evaluated the model for each of the three sets described in Sections II-A II-B using 4-fold cross-validation. So, in reality, we trained three different models, one for each set.

IV. Results

For sets 0, 1 and 2 we report accuracy values of 85%, 82% and 84%, respectively. The number of "best" features was 9 for every set.