Higgs Boson Binary Classification

Diego Dorn Sydney Hauke Ana-Maria Indreias

EPFL

diego.dorn@epfl.ch sydney.hauke@epfl.ch ana-maria.indreias@epfl.ch

1. Introduction

The aim of this project is to perform binary classification on the CERN Higgs Boson data set, in order to detect whether a Higgs boson was present during a given event. We begin with an exploratory data analysis phase, during which we discover important characteristics of the given features. Using these findings, we learn how to best pre-process our data set, and also develop a binary classification model.

2. Methodology

2.1 Data pre-processing

We explore the data using box plots and histograms of each feature. Doing this, we notice the presence of outliers - including some 120 standard deviations away from the mean! As seen in class, certain loss functions, such as the Mean Squared Error (MSE), do not perform well with outliers, so we are interested in removing these samples. To do so in a systematic way, we define a sample's **"weirdness score"** as the sum of its deviations across features. We decide to remove outliers, which we define as samples with a weirdness score greater than 60, thus removing 1825 samples out of 200 000.

We then notice strong correlation ($> 99.9\%$) among three binary feature groups. However, we notice that they correspond in fact to features for which values are undefined and were set to -999 [1]. We do not know how to process them meaningfully except by using different models for each value of `PRI_jet_num`, as linear regressions do not handle case disjunction well.

After visualizing the data, we also notice that some features are angles so we add two features for each with their sine and cosine as they could add some explanatory power. Similarly, some distributions are on an exponential scale, so we add a feature with their logarithms (to discover that their logarithm follows a gaussian).

We continue expanding features by experimenting with the polynomial basis function seen in class, adding the product of every pair of features that might make sense to multiply together (that is, not angles for instance). We also add the powers (up to degree 5) of each feature.

To further improve our predictions, we add a constant feature of value 1. Surprisingly, this standalone modification yields a 75% accuracy with the least squares.

In total, we crafted **150 new features** from existing ones.

2.2 Ridge regression

A ridge regression model is based on the least squares solution. Its hyperparameter, λ, is used to penalize bigger weights, i.e. of higher l_2-norm. We also use λ to obtain numerically stable matrices.

We choose ridge regression because it is fast and offers a closed-form solution. Furthermore, we are in a situation where we have less degrees of freedom (about 200) compared to the number of data points (about $1\,000\times$ more). In such context, there is no worry for overfitting, but underfitting is a much bigger problem (and is the reason why we perform feature expansion). Indeed, we get very similar accuracy on the test and training set, about 80%.

2.3 Stochastic gradient descent

A stochastic gradient descent model iteratively updates the weights vector. At each step, it computes a partial gradient, and hence is more computationally expensive compared to ridge regression. Nevertheless, it offers more flexibility for binary classification, as it allows clipping the predictions to [-1, 1] at each training iteration. This help to get one more percent of accuracy, which is not much compared to the higher computational costs. We still use this algorithm as our final model as it achieves the best accuracy.

2.4 Results

For all considered models, we transform predictions as follows: positive (> 0) predictions are mapped to 1, while the rest are mapped to -1. To measure their performance locally, we average the accuracies for ten 80-20 train-test splits. The results are summarized below.

Model	Accuracy
Background (always guess -1)	0.6567 ± 0.0023
Ridge Regression ($\lambda = 10^{-6}$), without feature expansion	0.7180 ± 0.0020
Ridge Regression ($\lambda = 10^{-6}$)	0.8000 ± 0.0013
Modified MSE SGD	0.8078 ± 0.0012

2.5 Techniques that were left out

We were keen to use **(regularized) logistic regression**, because it is a binary classification model ”out-of-the-box”. We experimented with 4-fold cross validation on the values of the regularization strength, but did not reach an accuracy as high as with simpler models. We believe this is the case because we did not choose a good gradient descent step size. Due to lack of time, we did not perform extensive cross validation on both hyperparameters.

Another technique that we do not employ is **handling the undefined values**. One can remove -999 values completely - however, 72% samples and 36.6% features have undefined values, so removing either would sharply decrease the data set size. One can also replace -999 values with zeroes, or the mean of valid values within each feature. We do not think replacement is a good technique for this data set, as -999 values are deliberately out of the normal range of the concerned features[1], and this information could be useful to our classifier.

3. Conclusions and future work

We develop a model which reaches 0.8078 ± 0.0012 accuracy, as evaluated using ten 80-20 train-test splits. For data pre-processing, we remove outliers and make use of feature expansion techniques. Overall, we notice big improvements to regression methods when using ”textbook” polynomial expansion, as well as pairwise products or trigonometric identities. Moreover, by understanding the various features' distributions, we are able to fine-tune this expansion by including logarithms of features $\{2, 9, 10, 13, 16, 19, 21\}$[1] and binary features based on the values of PRI_jet_num. For this reason, we believe that **exploratory data analysis is crucial in the development of our model.**

Going forward, two interesting research directions would be, respectively: using other basis functions for feature expansion, such as the truncated Fourier series [2], and **combining four models**, each corresponding to a value of the integer feature PRI_jet_num. As detailed in the data set's documentation, certain features are undefined if PRI_jet_num $= 0$ or ≤ 1 [1]. One could pass new samples to the corresponding model out of the four. We did implement a simple version of this concept, but ultimately decided against using it for our final model.

[1] Feature indexing starts at 0

References

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, D. Rousseau. Learning to discover: the Higgs boson machine learning challenge. 2014.

[2] R. P. Adams. Features and Basis Functions. 2018.