Machine learning applied to the Higgs boson CERN database

Lucía Montero Sanchis, Nuno Mota Gonçalves, Matteo Yann Feo,
Department of Computer Science, EPFL Lausanne, Switzerland

***Abstract*—TODO Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum**

I. Introduction

The aim of this project is to classify the data in the CERN database in two different classes, for finding the Higgs boson. This task will be carried out applying the machine learning methods learned during the course, together with the improvements that we have considered the most appropriate to overcome the challenges of this assignment. Some of the challenges of this classification process include parsing the missing information in the database, properly dealing with a large number of features and samples, and choosing the learning method that is the most suitable.

The learning algorithms we focus on are least squares and logistic regression, being the latter the most adequate for classification. The possibility of including a regularization factor is considered in order to avoid overfitting. We expand the features by including a polynomial basis up to a certain degree, to achieve a higher accuracy in the classification. We also use Principal Component Analysis (PCA) to obtain a smaller number of linearly uncorrelated variables.

II. Models and methods

A. Initial data parsing

The Higgs boson database contains 250000 observations and 30 features or variables. These variables take values in different ranges. Certain features contain the value -999 for some of the observations, which we have considered to represent *missing information*.

Before applying a learning algorithm we verify that all the features are standardized so that they are considered equally to avoid a negative effect on the ability to learn. To standardize the features we substract the mean and divide by the standard deviation. For the variables with missing information, the mean and the standard deviation are computed only over the non-missing values. Afterwards the missing values are set to 0 so that they do not modify the scale of the variable. When standardizing the test data we used the mean and the standard deviation of the train data. This assures the same transformation to both datasets, although it should not make a big difference since the distribution of the training and test data is assumed to be the same.

B. Polynomial basis

To improve the accuracy in the classification we increase the amount of explanatory variables by building a polynomial basis. We expand the features including the powers of each of the original variables up to a certain chosen degree. Since the newly included variables may have a different standard deviation, we normalize them to assure that their standard deviation is 1. We also include an offset term.

C. Principal Component Analysis (PCA)

The total number of features increases considerably for polynomial basis with a large degree. In these cases we reduce the dimensionality d of the feature space using PCA.

As specified in [1], to do so we first find the covariance matrix of the features. We then compute the eigenvalues and eigenvectors. The eigenvectors form a basis for the data, and they can be sorted in order of *decreasing* eigenvalue. We can then select a subset of the first $L \leq d$ eigenvectors as basis vectors. This allows to compare the results achieved by building a polynomial base with a larger degree and then reducing dimensionality by applying PCA, with the results achieved for a polynomial basis of a lower degree.

D. Least squares and Ridge regression

We start classifying with least squares. Since it is possible to have overfitting, we consider the possibility of using ridge regression. However, the ridge regression method implemented penalizes the offset of the model, which might decrease the accuracy of the predictions.

E. Logistic regression and Regularized logistic regression

We focus on using this method, since it is more suitable for classifying. For the logistic regression with stochastic gradient descent algorithm we have first considered a constant learning rate λ, and then an adaptive one $\lambda^{(t)}$ as shown in 1, where t is the iteration.

$$\lambda^{(t)} = \eta \cdot t^{-\kappa} \tag{1}$$

Although by using an adaptive learning rate we increase the number of hyperparameters to adjust, the convergence improves. As for least squares, there is a possibility of

overfitting when considering a large degree for the polynomial basis. In this case we would use regularized logistic regression, which would result into a third hyperparameter to adjust. The regularization implementation in this case does not penalize the offset term of the model, solving the problem mentioned previously.

F. Cross validation and hyperparameters tuning

For the cross validation we have considered 80% of the data for training and 20% for test. Out of the *training* data, 80% is used for training and the remaning 20% for validation (i.e. for tuning the hyperparameters). To compare the classification errors for each model, we define *accuracy* as the ratio of correct predictions. We choose the parameters for the adaptive learning rate $\lambda^{(t)}$ and the regularization factor γ by using a random layout (random search).

III. Results

We first predict the labels for the test data using least squares, that we implement using the normal equations. After building the polynomial basis with degrees 2 to 8 for the training data, we apply least squares. Figure 1 shows the accuracy obtained for the training, validation and test data for each maximum degree.

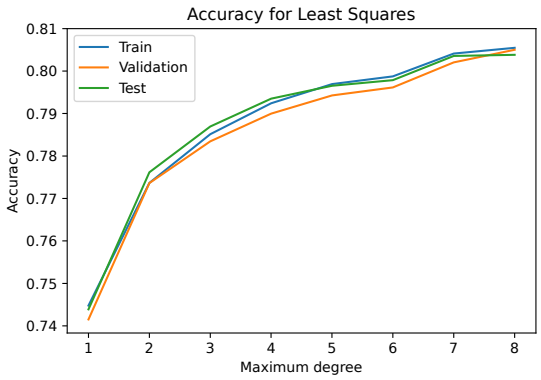

Figure 1. Accuracy achieved using Least squares

The accuracy increases for larger maximum degrees, and the increase in accuracy with respect to the degree gets lower as the degree gets larger. We also observe that there does not seem to be an overfitting, since the difference in precision between the training and the validation-test predictions do not seem large.

We then use the logistic regression Stochastic Gradient Descent (SGD) algorithm, in batches of 2000 and with 1500 iterations. As explained previously, we use an adaptive learning rate that is defined by two hyperparameters. In this case, the values are $\eta = 10^{-2}$ and $\kappa = 0.8$.

Figure 2 shows the accuracy obtained for the training, validation and test data for each maximum degree using logistic regression.

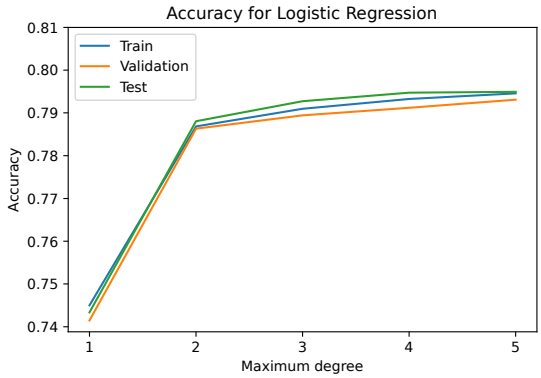

Figure 2. Accuracy achieved using Logistic regression

In this case we can again see that the accuracy increases with the degree of the polynomial, and that the model does not seem to overfit. Therefore, we do not include a regularization factor. As seen with least squares, the increment in accuracy is smaller for the largest degrees.

TODOPrincipal component analysis

IV. Discussion

In the results section we have shown that the accuracy obtained higher when considering a polynomial basis of a larger degree, since for larger degrees there are more features to consider when fitting the model. However, it has been shown as well that this difference in accuracy is larger when the degrees being compared are small. For instance, when using least squares it is not clear if it is worth it to consider a basis of degree 8 as compared to using a basis of degree 7. It is interesting as well to note that we did not find cases of overfitting when using either least squares or logistic regression.

We also found that although we expected the logistic regression algorithm to perform better than least squares, both achieve a similar accuracy for a same degree of polynomial. In fact, for some degrees least squares outperforms logistic regression. A possible reason for this is that we are obtaining the analytical solution for the least squares algorithm, whereas for logistic regression we have estimated it by choosing a series of parameters (learning rate, number of iterations, and size of the batch for SGD).

TODO: algo de PCA

V. Summary

TODO

References

[1] L. I. Smith, "A tutorial on principal components analysis," 2002. [Online]. Available: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf