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 precision 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. Otherwise this could have a negative effect on the ability to learn, especially with methods such as ridge regression. To standardize the features we have substracted the mean and then divided by the standard deviation. For the variables with missing information, both the mean and the standard deviation have been computed considering only the non-missing values. Afterwards the missing values have been set to 0 so that they do not affect the mean of the variable nor modify its scale.

When standardizing the test data we have used the mean and the standard deviation obtained for the training data. This makes sure that both datasets are transformed in the same way. However, this 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 precision in the classification we have increased the amount of explanatory variables by building a polynomial basis. We have expanded the features by 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 have normalized them afterwards to make sure that their standard deviation is 1. We also include an offset term, because the variable that we want to predict might have an offset.

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 start by finding 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 by classifying using 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 precision 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% has been used for training and the remaning 20% for validation (i.e. for choosing the values of the hyperparameters).

We have chosen the parameters for the adaptive learning rate $\lambda^{(t)}$ and the regularization factor γ by using a random layout (random search), instead of grid search.

III. Results

TODO Least squares (y Ridge regression??) - valores de training/val/test para least squares-¿ hay overfitting? si lo hay, mirar ridge regression y sacar la grafica para distintos factores de regularizacion (TODO? ridge regression que no penalice offset).

Logistic regression. We used batch size of 2000, limitado por un numero de iteraciones de ...?.ogistic regression with stochastic gradient descent. Para degrees de 1 a 4, y con o sin los 50% mas relevantes de PCA. Decir lambdas utilizadas (los dos valores, tabla?). Comparar errores de training, test y validation para las 8 opciones. Con eso, decidir si usamos regularized logistic o no. (Regularized logistic regression? decir lambda y factor de regularizacion usado, y en que rango se han buscado.)

IV. Discussion

As explained previously, this offset is required due to the characteristics of the variable that we want to predict. Therefore, we need to bear in mind that it is possible that the precision obtained using ridge regression is lower than using least squares. (ridge vs ls) TODO Adaptive lambda Hay o no hay overfitting (en least squares y en logistic regression) Es mejor usar degrees mas bajos y no aplicar PCA. Mirar para cada degree.

V. Summary

TODO Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam at magna sodales, posuere enim id, vulputate mi. Praesent tristique augue nec augue mattis, eu rhoncus eros rhoncus. Donec pellentesque, lectus a lacinia condimentum, turpis mi consectetur sem, nec scelerisque lorem ex non tortor. Vestibulum ut dictum orci, sit amet auctor nisl. Aenean ullamcorper nulla eu velit sodales, iaculis suscipit ligula pulvinar. Quisque varius auctor tellus, vel commodo tellus volutpat non. Praesent augue neque, ultricies at nulla vitae, pellentesque rutrum velit. Sed fermentum arcu lorem, eu pulvinar diam pellentesque eu.

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