CS-433 Machine Learning: Project 1

Olle Ottander, Paolo Celada and Gustav Karlbom
Department of Computer Science, EPFL Lausanne, Switzerland

***Abstract*—A critical part in the ATLAS experiment is being able to distinguish between a *tau tau decay of a Higgs boson* versus *background*, using data detected after a head-on collision between 2 protons. The projected consisted of applying machine learning methods to a set of original and already classified decay signatures in order to predict unseen ones. After an initial cleaning and preprocessing step, multiple ML methods have been tested on training data and the relative test error measured locally using cross-validation. The best model's prediction were then submitted to an online platform which calculated both accuracy and F1 score.**

I. Introduction

After the discovery in 2012 of the Higgs boson, physicists at CERN started working on how to measure its characteristics, and determine if it could fit the current model of nature. Through the ATLAS experiment multiple small signals have been detected, showing that a Higgs boson can decay into two tau particles producing a so called *decay signature* [1]. Since these signatures are always buried in background noise this paper describes several Machine Learning methods used to create a model which can classify them. In Section II the step performed in order to create the best model are explained in detail. This consisted of exploratory data analysis, data cleaning, preprocessing, model optimization, and cross validation. This is followed by a results section where obtained values from different models are presented.

II. Models and methods

A. Initial ML Methods Implementation

Firstly, the Machine learning methods seen so far during lab sessions were implemented. These are linear regression using normal equations, gradient descent and stochastic gradient descent, ridge regression, logistic regression and regularized logistic regression. From data shown in Table I, it can be noticed that in general all methods perform similarly, with the best one being Least Squares with a prediction accuracy of 0.7468. It is important to notice how, without any data cleaning and preprocessing, linear methods overall perform better compared to classification methods.

TABLE I
Accuracy and hyper-parameters for initial methods

Method	γ	λ	**Degree**	**Iterations**	**Accuracy**
Least Squares	-	-	-	-	0.7468
Gradient Descent	0.1	-	-	100	0.7416
Stoc. Gradient Descent	0.01	-	-	100	0.6797
Ridge Regression	-	0.01	7	100	0.7433
Logistic Regression	0.1	-	-	200	0.7387
Reg. Logistic Regression	0.1	0.1	-	200	0.7277

B. Data Cleaning and Preprocessing

After a first implementation, multiple measures were taken regarding the training dataset. Plotting the distributions of each feature separately, together with a quick look to the original challenge's documentation[2], enabled detecting inconsistencies or properties that could be exploited to improve the models:

- only feature *PRI Jet Number* is categorical, the rest are numerical;
- invalid input data that have been set to -999 have a direct relation with the categorical feature (except feature *DER Mass MMC*);
- different features present right-skewness, with the presence of outliers.

Starting from the first property, the training data was split into 4 different subgroups, based on the value assumed by *PRI Jet Number* feature (i.e. 0, 1, 2, 3). The subgroups were unbalanced at first, with a significantly lower number of entries in subgroup 3. Both subgroup 0 and subgroup 1 presented invalid values for multiple features, while neither subgroup 2 nor 3 presented any. Hence, subgroup 2 and 3 were considered together as a joined group, achieving a more balanced division (as show in Table II). Consequently, features which contained only -999 for each subgroups were removed. The feature *PRI Jet Number* was also removed since it was not necessary anymore.

To handle invalid values in *DER Mass MMC* feature, which as written before does not have any relationship with the categorical feature, it is opted for a remove of the whole decay signature when dealing with the training set, and for a substitution with the mean for the testing set.

TABLE II
Features with -999 per subgroup

Subgroup 0	Subgroup 1	Subgroup 2
DER deltaeta jet jet	*DER deltaeta jet jet*	None
DER mass jet jet	*DER mass jet jet*	
DER prodeta jet jet	*DER prodeta jet jet*	
DER lep eta centrality	*DER lep eta centrality*	
PRI jet leading pt T	*PRI jet subleading pt*	
PRI jet leading eta	*PRI jet subleading eta*	
PRI jet leading phi	*PRI jet subleading phi*	
PRI jet subleading pt		
PRI jet subleading eta		
PRI jet subleading phi		

C. Feature Transformation

For right-skewed distributions a logarithmic transformation was applied to make the distribution more similar to a normal

distribution and thereby easier to handle for optimization methods. A function to remove outliers was also applied, using 1% and 99% percentile (this function is only called when training the model, not testing it).

To improve accuracy of linear methods, polynomial feature transforms were performed on all features up to degree 7 for least squares and ridge regression. Consequently a slight increase in prediction accuracy was recorded, at the cost of adding a large number of new input features. The function used is the following one:

$$f(X, deg) \rightarrow (1|X^d), \forall d \in [1, deg)$$

The degrees used for the last submission were lower for classification methods (2) and higher for linear methods (7), which avoids any over-fitting problems. For future project implementations, to boost even more accuracy and exploit features non-linearity, feature crosses can be used (a form of feature combination). Lastly, all features present at this points were standardized to get 0 mean and 1 standard deviation.

D. Hyper-parameters Selection

A key part in method selection is choosing the best hyper-parameters to have better performance while reducing the error as much as possible. For all methods except least squares, the focus was on evaluating the loss with respect to different γ and λ values. By brute forcing a number of different hyper-parameters, and training each model on 2 subsets of the training set, it was possible to pick the ones which gave the smallest loss with respect to the train and test error. The subset considered in this case was built by splitting into training and test subsets given a *seed* and a *split-ratio*, 56 and 0.5 respectively. It is worth noticing that this procedure could have been performed also exploiting cross validation, and its ability to compute the best hyper-parameter for an average of all subgroups combinations.

E. Cross Validation

Lastly, cross-validation was used to determine which were the best preprocessing steps and which were the best optimization methods. To settle this all methods were tested using K-fold cross-validation on the train data. Choosing 4 folds was an optimal trade-off between computational speed and result reliability. The generated results showed how the models performed on average for all combinations of train and test sets. This step was used to improve the reliability comparing the accuracy of the preprocessed and non-preprocessed data.

III. Results

Comparing all methods and hyper-parameters together, using K-fold cross-validation to quantify train and test errors, the conclusion was made that the logistic regression performed better on average for all subgroups. The regularized logistic regression also performed on a comparable level. The final submission on the online platform consisted of using logistic regression for subgroup 0, 1 and 2, using degree 2 polynomial expansion and data cleaning and preprocessing. Hyper-parameters chosen and final accuracy and F1 score are shown below in Table III.

TABLE III
FINAL ACCURACY AND HYPER-PARAMETERS

Method (Subgroup)	γ	λ	**Iterations**	**Final Accuracy**	**F1**
Logistic Regression (0)	0.3	-	500		
Logistic Regression (1)	0.3	-	500	0.812	0.729
Logistic Regression (2)	0.3	-	500		

IV. Discussion

Creating an ML model is always a trade off between computational speed and results, with a particular focus on hyper-parameters selection. The model could have been further optimized to obtain an ever better result with more measures taken on the training dataset. However, the improvements made were considered good enough within the scope of the project.

A final note can be made on the choice to divide the data into subgroups, which has both pros and cons. As explained in the method section(II), it made the model more efficient as data was more balanced fully exploiting the dependency on the characteristic feature. However, dividing the dataset into three subgroups causes the model to have less data points to train on, compared to if it could be trained on the entire dataset.

V. Summary

The model which gave the highest accuracy was the logistic regression with an accuracy of 0.812. This was obtained using $\gamma = 0.3$ and 500 iterations.

The main measures taken to make the model perform better with respect to the data have been to divide the data into subgroups based on feature *PRI Jet Number*, to transform features and optimize hyper-parameters for each ML method.

References

[1] CERN, "Higgs boson machine learning challenge," 2014. [Online]. Available: https://www.kaggle.com/c/higgs-boson/overview

[2] C. Adam Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, "Learning to discover: the higgs boson machine learning challenge," 2014. [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf