Machine Learning - Class Project 1

Team #5
"Only the journey is written, not the destination"
Varun Batta, Jan Bednarik, and Sahand Kashani

I. Introduction

Discovering the Higgs boson, while a revolutionary finding in the world of physics, involved hundreds of thousands of data points and a lot of data analysis before any conclusive findings. For all this, machine learning was a tool that was thoroughly used to help classify which data points refer to a Higgs boson signal, and which are just background noise from the other particles.

When taking on this challenge ourselves, we considered a couple of possible approaches. However, after initial analysis, we discarded the regression approaches, namely least squares and ridge regression, and we decided to use classification approach instead since the given task is in nature binary classification. This report explains our approach towards this challenge, and how different ideas in machine learning were applied to minimize classification errors.

II. Approaches and Analysis

To begin this challenge, we were given a data set containing $N = 250000$ data points as training data that had already been classified. Each of these data points had a $D = 30$ dimensions, 1 of which was categorical with 4 categories. We were also given a test data set of $N = 568238$ data points, simply to check how successful our classification algorithm was.

After analysis of the task, we felt that least squares, ridge and lasso regression were all more focused on fitting the model to go through each point of data. We are not looking for a function to fit the data, but instead a function to classify and split the data appropriately. To this end, we decided to use logistic regression.

There are two other factors to consider: the cost function and how much regularization to use. While we are unable to show you the necessary data, due to timing constraints, our findings were that for the cost function, both Mean Absolute Error (MAE) and Mean Square Error (MSE) do not fit the requirement of classification, so we ended up using logistic error. We feel that regularization is important to consider, but since the L1 models were failing to successfully zero out any of the weights, we decided to work with L2.

III. Feature Engineering

A. Pre-analysis

In order to help compare the models and approaches we established a baseline to be the unconstrained logistic regression operating on the standardized version of the original features. Each feature vector $\vec{f}$ was standardize so as to have zero mean and unit variance: $\vec{f} = \frac{\vec{f} - \mu_{\vec{f}}}{\sigma_{\vec{f}}}$. This baseline yields classification error $E_{cls} = 0.7499$.

B. Outlier Elimination

Brief exploratory data analysis exhibits significant number of outlier values $outlier = -999$ in certain dimensions of the original datasets (see Figure 1). The value $outlier$ was chosen arbitrarily and it represents meaningless values [1]. In order to keep all the dimensions we performed the outlier removal separately for each dimension $d_i \in D_{train\_test}$:

$$d_{ij} = \begin{cases} d_{ij}, & \text{if } d_{ij} \neq outlier \\ \mu_{d'_i}, & otherwise \end{cases}, j = 1..N_{data},$$

where D_{train_test} is the joint train and test dataset, d_{ij} is the j-th data sample in i-th dimension, $\mu_{d'_i}$ is the mean of all the inliers in dimension d_i and N_{data} is the number of values in each dimension.

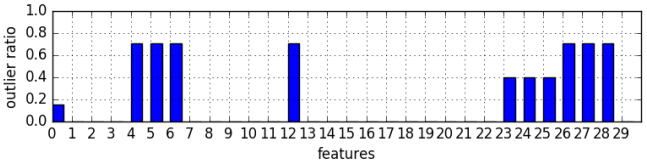

Figure 1. Outlier ($outlier = -999$) occurrence ratio for each dimension of the joint train and test dataset.

C. Decomposing Categorical Attributes

The dataset contains one categorical attribute (labeled as `PRI_jet_num`) where the categories are distinguished by numerical integer values [1]. However, the quantities used do not carry any useful information per se. Therefore we used a better approach to expose the categorical attributes to a linear classifier [2] — the one-hot coding. The categorical feature f' is substituted by new features f_i:

$$f_i = \begin{cases} 1, & \text{if } f' = C_i \\ 0, & otherwise \end{cases}, \forall f' \in D_f, i = 1..N_c,$$

where D_f is the vector of quantities corresponding to the categorical feature of the original dataset, N_c is the number of categories and C_i is the i-th category.

D. Polynomial Expansion

Since, to this point, the *background* and *signal* classes cannot be separated perfectly, we believe that in higher dimensional feature space the better decision boundary might exist. Therefore we perform polynomial expansion of all the features yielding a new set of features $\{f_i f_j | f_i, f_j \in F\}$, where F is the original set of features extended by one-hot coding of categorical column (see Section III-C). Polynomial expansion of features results in a new set of 558 features.

IV. Dealing with Overfitting

To reduce overfitting we perform 3-fold cross-validation on our logistic regression by dividing our training set into 2 parts: 66 % of the data is used for training, and the other 33 % is used for validating the model parameters obtained after training. We settled on this ratio as a trade-off between execution time and the resulting training and validation errors obtained. To ensure the ordering of samples within the dataset do not influence the training and validation errors, we shuffle all samples prior to dividing the dataset into the 3 regions.

Unfortunately, logistic regression suffers from a non-ending optimization process if the underlying data is linearly separable. However, since we do not know if linear separability is present in advance, we decided to use L2 regularization and added a penalty to our optimization cost function. The issue then becomes to find the best regularization parameter λ to avoid underfitting the test data. To find the best λ, we performed a grid search with exponentially-spaced values until we found the range of values for which the validation set's error dropped, then continued with a fine-grained linearly-spaced search. Figure 2 shows the result of the grid search for λ and the associated training and validation misclassification ratios.

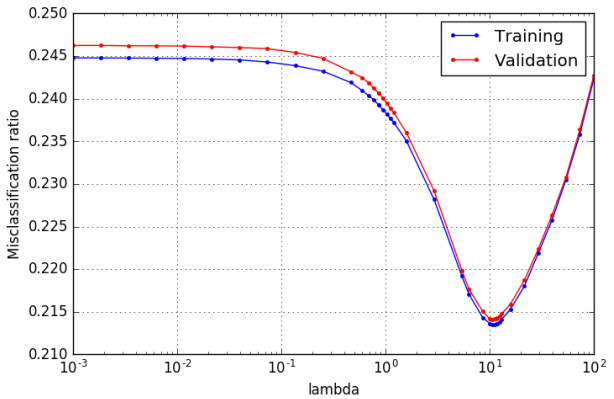

Figure 2. Misclassification ratio as a function of λ after cross-validation.

V. Results

Given the results obtained from cross-validation, the lowest classification error should be obtained when fixing the regularization parameter $\lambda = 10.6$. However, after running unconstrained logistic regression as a sanity check, we found out that the learner exhibits the lowest validation error for $\lambda = 0.0$.

Therefore, we selected the final model to be the unconstrained logistic regression and it was trained on the whole training dataset transformed using the steps presented in Section III. The step size was empirically selected to be $\gamma = 0.08$ as an optimal trade-off between learning speed and magnitude of prospective oscillations of loss over the course of training. The full-batch gradient descent was selected as the optimizer performing $N = 7000$ steps.

Using such a setting results in smoothly decreasing loss function (see Figure 3). The resulting classification error obtained while testing on training dataset $E_{cls_{train}} = 0.1822$ while the classification error obtained on the test dataset[1] $E_{cls_{test}} = 0.1831$. Negligible difference between $E_{cls_{train}}$ and $E_{cls_{test}}$ proves that the model is not significantly overfitting even when regularization is not used.

Note that the best result we reached on Kaggle platform (yielding $E_{cls_{test}} = 0.1785$) was obtained using ridge regression, however, we decided to use logistic regression for this task, as was explained in Section II.

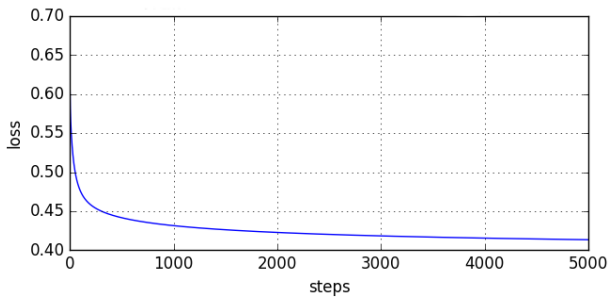

Figure 3. Training loss as a function of training steps of the gradient descent optimizer computing unconstrained logistic regression (first 5000 steps).

VI. Summary

The system for binary classification of events representing a decay of Higgs boson and background events was described in this report. To obtain the final result it was necessary to appropriately transform the input features and to search for the best hyper-parameters using cross-validation. Out of multiple models the unconstrained logistic regression yielded the lowest classification error.

[1]The classification error for test dataset were obtained after the predictions were uploaded to the Kaggle platform.

REFERENCES

[1] C. Adam-Bourdariosa, G. Cowanb, C. Germain, I. Guyond, B. Kegl, and D. Rousseaua, "Learning to discover: the higgs boson machine learning challenge," Jul. 2014. [Online]. Available: https://higgsml.lal.in2p3.fr/files/2014/04/documentation_v1.8.pdf

[2] J. Brownlee, "Discover Feature Engineering, How to Engineer Features and How to Get Good at It," Sep. 2014. [Online]. Available: goo.gl/Ld8dk4

[3] P. Domingos, "A Few Useful Things to Know About Machine Learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012. [Online]. Available: http://doi.acm.org/10.1145/2347736.2347755