CS-433 Project 1: Report

Una Pale, Philippe Tueckmantel, Pablo Maceira

I. Problem Statement

THIS first project consisted of a classification task, applied to a data set originally published by CERN. The goal was to classify a number of samples, described by 30 features, into two groups: signal and background. Provided features included, in many cases, missing values, which is one why preprocessing of the data was necessary. Additionally, we explored other characteristics of the data, such as skewness of the distribution, existence of structure, etc. Initially applied methods (i.e. least squares, gradient descent, etc.), well suited for linear regression, yielded poor results. We applied different classification techniques and tried multiple combinations of parameters before choosing the method that generated our final submission. Classification performance was assessed using various error metrics and also through the Kaggle platform.

II. Methods

A. Data Exploration and Feature Preprocessing

We first applied different visualization tools in order to acquire some notion of how we could process the data. Our data exploration (see "Data exploration final version" Jupyter Notebook) led us to choose a procedure consisting of an initial replacement of all missing values (i.e. -999) by "NaN", in turn replaced by the mean of each feature. We checked what the difference would be if we replaced missing values with the mean of each class. Indeed, classification accuracy increased during cross validation, but as we could not give the same treatment to the test data set, we applied the mean of all features, irrespective of class.

We noticed that for some features, the percentage of missing values was very high (70%), which is why later on we tried removing these features altogether to see if we could improve the results of classification. Removing the features did not help. Nevertheless, looking at the proportion of missing features for each class separately made us notice that there were consistently more occurrences of missing data in Background samples than in Signal samples, which lead us to add extra features containing a mask of where missing values had been present.

After removing outlier values (i.e. larger or smaller than $\mu \pm 3\sigma$), we looked at the general distribution of the data for each feature. As some of the features presented a skewed distribution, we tried to transform data (all or only data depicting at least moderate skewness). Cubic transformation was performed for skewness to the left and log transformation for skewness to the right [1]). It turned out that it is better to modify only data with significant skewness, but in general that is better not to change distribution at all. .

In addition, we performed polynomial expansion for individual features up to degree 10, and added interaction terms between features up to degree two. Further, we added extra features in the form of sines and cosines.

The features *PRI tau eta*, *PRI lep eta* and *PRI jet leading eta* are projections along the plane perpendicular to the detector's long axis. The pseudo-rapidity η is used, which is related to θ by $\eta = -ln(tan(\theta/2))$. We tried computing the reverse transformation $\theta = 2arctan(-\eta))$ in order to assess whether the θ coordinate can give additional information. This also improved accuracy slightly.

After standardization, we applied bucketing to test the possibility of data not depicting a linear distribution in the high dimensional space, segmenting the data into several buckets and adding features accordingly (see Figure 3). In the end, 10 buckets were chosen. Increasing number of buckets also increases number of features fast so the computation is much longer. Bucketing where each bucket represents equal data values ranges or equal number of data-points per bucket were tested, the latter resulting in better accuracy.

B. Label Prediction

After preprocessing, we applied different classification techniques and tried to assess their performance using 80% of the training set to train and 20% to test (1X cross-validation). This approach was used due to the high computation times inherent to cross validation with more folds. We tested multiple combinations of preprocessing steps described in the previous subsection, and chose the steps that lead to highest classification accuracy in our test subsample. 4-fold cross-validation was used, varying γ and λ for multiple sets of preprocessing steps and parameters and we compared the training and test scores to estimate what preprocessing steps worked better; further, it provided some notion of how well our model would generalize to unseen data (i.e. the test set).

To classify the data, we started with least squares, as it was easy to implement and quick to run, but results were not satisfactory, besides the fact of the data matrix becoming singular as the number of features was increased. In contrast, ridge regression gave good results during cross-validation, but the resulting model did not generalize to the test set on Kaggle.

Independent reading and material discussed on the lectures made clear that we needed a different cost function, as the mean-square error and mean-absolute error could yield inadequate classification results depending on the distribution on the data points in the multidimensional space. The logistic function was introduced with this purpose. We used regularized logistic regression with an L2-regularizer. The optimal learning rate γ and regularization parameter λ were both determined iteratively, observing accuracy on training and test sets (see Figure 1). In order to have γ and λ not dependent on the number of observation we used score that is averaged.

Logistic regression and ridge regression gave us the almost equally good results, so we tested it with various parameters to achieve an optimal setup (see Performance_RidgeLigistic_DiffParameters.png for details) . Note: Labels where changed from -1,1 to 0,1, since this is one of the assumptions of the formula for logistic regression that we used.

As we used over 1000 features, we tried to see whether any improvement could be made by selecting some of the features only. This would also simplify the model and reduce computation time. We applied Recursive Feature Elimination [2] to gradually remove features with lowest weights, assuming they explained the data to a lower extent. The effect of removing these features is shown in Figure 2. Even though there seems to be an increase in classification accuracy for the test set, improvements are very small, but reducing computational cost could make this approach attractive in future endeavors. Additionally, we explored the use of PCA to see whether the exclusive inclusion of features with most variance explained could give better results. A 99% threshold for cumulative variance was set, which removed almost half of the features, but in the end accuracy didn't improve.

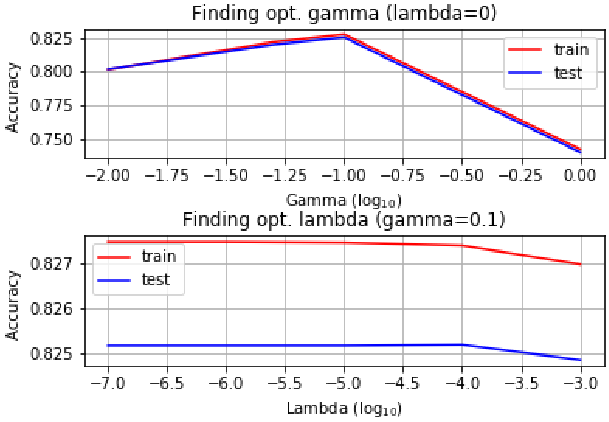

Fig. 1. Optimizing γ and λ parameters. Ultimately, learning rate was fixed to γ=0.1 and λ=0.0001 even though from the plot it is visible that overfitting still hasn't happened because smaller λ were giving better results.

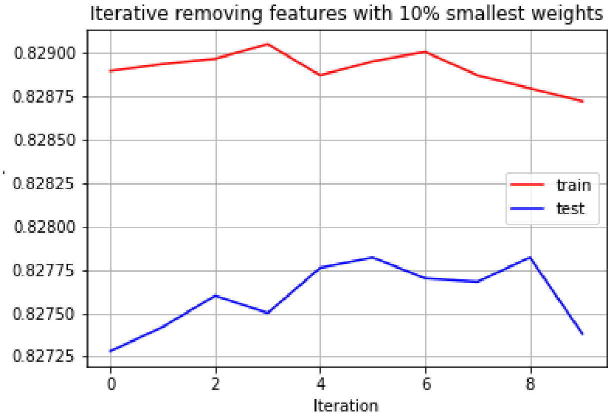

Fig. 2. Iterative selection of features. At every iteration, 10% of the weights with smallest values were removed

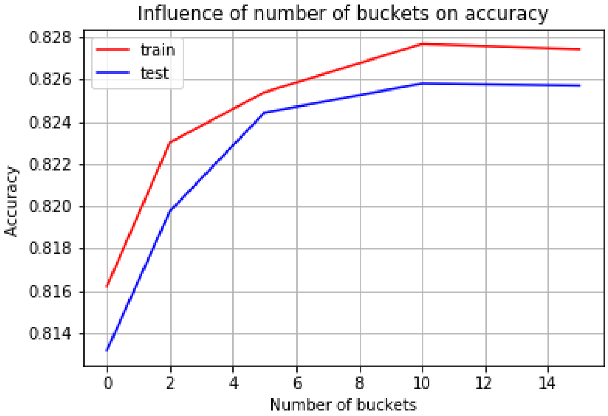

Fig. 3. Influence of number of buckets on accuracy. Number of buckets were increased up to 30 (only 14 shown, as saturation occurred after 10 buckets)

III. Results and Discussion

Highest classification accuracy (83.35%) was attained with a learning rate $\gamma = 0.1$ and $\lambda = 0.0001$ using logistic regression and in total around 1200 features. Including so many features raised concern due to the risk of overfitting, but as varying the value of λ seemed not to affect the accuracy too much and as removing features with lowest weights did not help, we decided to keep them. With more time, and, especially for future endeavors, it would be interesting to explore more popular dimensionality-reduction methods. Furthermore, documentation on the Higgs-boson problem suggested the use of a different cost function, which could be interesting to explore as well. More advanced methods for classification, such as support vector machines (SVM) could yield better results than logistic regression, which could be key to improve classification accuracy achieved so far.

References

[1] V. SaiGayatri. Data transformation: Skewness, normalization and much more. https://medium.com/@TheDataGyan/day-8-data-transformation-skewness-normalization-and-much-more-4c144d370e55.

[2] M. Pal, G.M. Foody. Feature selection for classification of hyperspectral data by SVM. *IEEE Trans. Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2297-2307, May. 2010.