Machine Learning Project1 Report

Bohan Wang
321293
bohan.wang@epfl.ch

Ke Wang
326760
k.wang@epfl.ch

Siran Li
321825
siran.li@epfl.ch

***Abstract*—In the first project, we apply six machine learning algorithms into a real-world problem, 'Higgs boson classification'. We implement different feature augmentations, cross-validation for model selection and evaluate the performances of six models. To this end, the best performed model is 83.6% on test set using ridge regression.**

I. Introduction

The Higgs boson is an elementary particle in the Standard Model of physics which explains why other particles have mass. Based on a vector of features representing the decay signature of a collision event, we will use six machine learning methods to classify whether it is a Higgs boson, including linear regression using gradient descent, linear regression using stochastic gradient descent, least squares regression, ridge regression, logistic regression and regularized logistic regression.

II. Feature Engineering

Numerous methods have been implemented for missing values, data processing, and feature augmentation.

A. Missing values

There are a lot of -999 values in the dataset. Some features such as x_4 and x_5 contain above 70% missing values. These missing values greatly impair the performances of models. We replace them with the median value of the other valid values (without the -999) considering the mean is sensitive to the outliers.

If we check only the data points with missing values, we can discover that most data points with missing value belong to -1 class. The ratios between positive samples (y = 1) and negative samples (y = -1) of the data with missing values in x_0, x_{23}, x_{24} and x_{25}, are significantly less than the overall ratio .

Therefore, we consider that whether containing missing values in these four features is related to class label. We create the indicator feature for each of them: 1 indicates the sample has a missing value, and 0 if it does not.

B. Feature Importance Study

We use three different methods to study the importance of each feature.

1) **correlation with y**
   We computed the Pearson correlation coefficient of each feature to y, and discovered three most important features: x_1, x_{11}, x_0, and the least important features are: x_{14}, x_{17}, x_{18}
2) **data distribution of opposite class labels**
   The data distribution of each feature is plotted separately for opposite labels, from which we discovered that x_1 has a clearly distinct data distribution for opposite class labels, suggesting its importance.
3) **correlation between features**
   The mutual correlation of each feature pairs are calculated, from which we discovered that features x_9, x_{21}, x_{23}, x_{29} are much correlated to each other, potentially undermining their importance.

Now we have specified the most important features x_1, x_{11}, x_0, and potentially unimportant features x_4, x_5, x_6, x_{12}, x_{26}, x_{27}, x_{28} (for too many missing data), x_{14}, x_{17}, x_{18} (for limited correlation with y), and x_9, x_{21}, x_{23}, x_{29} (for high mutual correlation). However, simulation results suggest that deleting these features provide very limited improvement to classification accuracy (even negative affects), so we decided to keep all the features. But this study helps us successfully recognize the most important features.

Besides, we have discovered that, despite relatively low correlation to y, feature x_{22} (PRI_jet_num) plays an very important role. The data with different jet-num follow very distinctive distributions. To visualize the distinction, we use Principle Component Analysis (PCA) to reduce the features dimensions into 2-D, and plot the points of different PRI_jet_num in different colors in Figure 1.

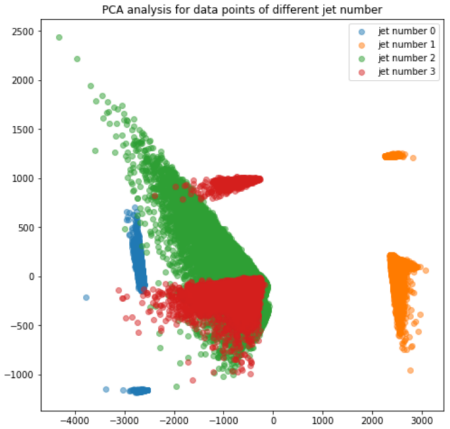

Fig. 1: PCA components of samples with different jet-num

C. Feature Augmentation

The integer powers of features and cross-products with other features can provide more information to the classification

apart from the original features. Therefore, We explore four types of feature expansion strategies

1) polynomial expansion (k∈{2,3, ..., 9})
   $[x_1, x_2\ , \ldots] \rightarrow \left[x_1^k, x_2^k, \ldots\right]$
2) 2nd-degree cross product of each two features,
   $[x_1, x_2, x_3\ , \ldots] \rightarrow [x_1x_2, x_2x_3, x_1x_3, \ldots]$
3) 3rd-degree cross product.
   We select the top 3 important features [x_1, x_{11}, x_0], and the feature x_{22} (*PRI_jet_num*) , and add the 3rd-degree cross product of these four important features with other features [x_2, x_3, x_4, ...]. $[x_0, x_1, x_2, ...] \rightarrow [x_1^2x_2, x_1x_2^2, x_1^2x_3, x_1x_3^2, ...]$
4) Negative integer power (-k) (k∈{1,2,...,5}).
   A small constant value δ is added to avoid numerical problems,
   $[x_1, x_2, \ldots] \rightarrow \left[\frac{1}{(x_1^k+\delta)}, \frac{1}{(x_2^k+\delta)}, \ldots\right]$
5) Expand over jet-num

   As data of different jet-num (x_{22}) follow different distributions, we propose a method to train a separate weight for data with different jet-num. The procedure is sketched as follows (j represents the jet-num, d represents the rest features):

$$\begin{bmatrix} d_0 & j_0 \\ d_1 & j_1 \\ d_2 & j_2 \\ d_3 & j_3 \end{bmatrix} \Rightarrow \begin{bmatrix} d_0 & j_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_1 & j_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_2 & j_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_3 & j_3 \end{bmatrix} \tag{1}$$

   Using this expansion, in practical it will be like we trained four different models for each jet-num.

III. Results

In this part, we report the performances of our models.

A. Ablation study

We first implement feature augmentation to improve the non-linearity of our classifier and learn more complicated decision bounds. We conduct an ablation study to evaluate the contributions of above proposed feature augmentations. We use ridge regression here considering that ridge regression can compute the optimal parameters fast with less risk for overfitting, and does not suffer from the non-invertibility of Gram matrix. We demonstrate the results in Table I. Note that one-hot encoding columns of 'PRI_jet_number' does not improve the performance significantly, so we did not use one-hot encoding later on.

B. Cross validation

After feature augmentation, We implemented 5-fold cross validation to find the best hyper-parameters such as polynomial degree (d), negative integer power ($-k$) and regularization parameter (λ). For SGD, we decay the learning rate by 0.1 once the iteration reaches half of the max iterations. This can help SGD converge better when the weight parameter ($\mathbf{w} \in R^D$) is close to the optimal solution. Finally, we obtain the best hyper-parameter configuration of the 6 methods (as shown in Table II).

Model	Train acc	Val acc
Base +cm	0.745	0.745
Base +cm + in	0.745	0.746
Base +cm + in + one_hot	0.746	0.747
Base +cm + in + pf	0.828	0.825
Base +cm + in + pf + scs	0.834	0.829
Base +cm + in + pf + scs + ccs	0.837	0.831
Base +cm + in + pf + scs + ccs + np	0.834	0.831
Base +cm + in + pf + scs + ccs + np + fe	**0.840**	**0.833**

TABLE I: Train and validation accuracy of ablation study. The symbol, 'cm' refers to replacing missing values with the median. 'in' refers to adding indicator features of the missing values. 'one_hot' refers to adding the one hot encoding feature. 'pf' refers to adding 9 order degree polynomial features. 'scs' refers to adding quadratic cross-product synthetic features. 'ccs' refers to adding cubic cross-product synthetic features. 'np' refers to adding negative integer power of meaningful features. 'fe' refers to feature expansion over jet-num.

Methods	max_iters	γ	λ	d	$-k$
Least Squares_GD	2000	0.001	/	8	-5
Least Squares_SGD	8000	0.001	/	8	-5
Least Squares	/	/	/	8	-5
Ridge Regression	/	/	0.0001	10	-8
logistic Regression	8000	0.001	/	8	-5
Regularized Logistic	8000	0.001	0.001	10	-8

TABLE II: Best hyper-parameters of each model

IV. Discussion

In our result, all models have close performance. We choose ridge as our final classifier as it is fast to train and avoids overfitting. It finally achieves 83.6% accuracy on learning board with the test set. We find that the promising result will be achieved, if we do some meaningful feature augmentations. This is probably caused by two reasons. First, the decision bound for the dataset is highly non-linear and complicated. In the future, we can implement support vector machine (SVM) with gaussian kernel or neural networks to learn even more complexed classification bound. Second, only 30 features in the dataset (much less than the size of samples, 250000) can easily result in under-parameterization. We can implement K-means clustering or Density-based clustering (DBSCAN) to create new useful features.

V. Conclusion

In this project, we evaluated different models for Higgs boson classification and saw that missing value processing and feature augmentation can improve the performance of classifiers. The ridge classifier with feature processing techniques achieved 83.6% accuracy on competition platform. In the future, we can do more meaningful feature augmentations and implement more complex machine learning methods to achieve better classification accuracy.