CS323 Machine Learning Project 1 - Higgs Boson

Xiaoqi Ma, Diyuan Wu, Shuo Wen
Department of Computer Science, EPFL Lausanne, Switzerland

***Abstract*—For complex datasets with high dimension, machine learning is an efficient tool to interpret, analyze and extract core information. Combined with the data from scientific problems and machine learning algorithms, we can predict scientific experiment outcomes. In this project, we use binary classification models to predict the occurrences of the Higgs Boson with efficient accuracy.**

I. Introduction

The Higgs boson was discovered at CERN in 2013 by collision experiments. From the datasets provided by CERN, we need to predict whether the collision events were the signal of the Higgs boson or background noise according to a vector of features which represent the decay signature of collision events. To achieve this purpose, we implement six machine learning algorithms to complete this binary classification task.

II. Models and Methods

A. Data Preparation

There are three steps in data preparation: first the dataset is diveded into 8 groups; then the unusable data and meaningless features are removed; finally the remaining features are scaled.

For the train set and test set provided by CERN, there are 30 features to represent the the signature of the events. By observing the distributions of these features, we firstly classified the events into 4 categories according to the PRI jet number with only four types of value in 0,1,2,3. Then, we find the feature: DER mass MMC, which estimates the mass of the Higgs boson, contain a lot of value '-999' in the data due to the topology of event(others are positive numbers), so we split each type of data again. Finally, we split the train/test dataset into 8 groups.

For the 8 datasets, we remove the meaningless features based on the following rules:

(a) Invalid features. We observe that some special features are all invalid values (-999). For example, the feature DER_mass_jet are all invalid for set which jet number equals zero. Therefore, we delete such invalid columns.
(b) Correlated features. For example, in set with jet=0, the Der_pt_h is totally equal to Der_pt_tot, so we deleted one of them.
(c) Unchanged feature. For example, the PRI_jet_all_pt is all equal to 0 in jet 0, so we delete that column.

After removing some meaningless features, we used the mean of each feature to replace the special invalid value (-999) and then standardized each column.

Because the linear models are highly prone to overfitting, we decide to add a polynomial basis to extend the feature vectors. Here we use least suqare model to find the best degree for each of the 8 datasets, and the result is shown in Table I.

Data Groups	Degree	Prediction(%)
jet=0,mass>0	4	79.906
jet=1,mass>0	6	77.348
jet=2,mass>0	6	80.840
jet=3,mass>0	7	79.925
jet=0,mass<0	5	94.867
jet=1,mass<0	5	91.827
jet=2,mass<0	2	90.210
jet=3,mass<0	1	92.955

Table I
Optimal Degree for different groups of data

B. Learning Algorithms and Parameter Selection

1) Learning Algorithms: In this project, we implement 6 different learning algorithms: Linear regression using gradient descent (least_squares_GD); Linear regression using stochastic gradient descent (least_squares_SGD); Least squares regression using normal equations (least_squares); Ridge regression using normal equations (ridge_regression); Logistic regression using gradient descent (logistic_regression); Regularized logistic regression using gradient descent (reg_logistic_regression).

2) Parameter Selection: In general, there are three parameters that needs to be selected: the learning rate γ, the maximal iteration numbers max_iters, the regularization coefficient of ridge regression and regularized logistic regression λ.

The method for selecting parameters is as following: first, we choose γ and max_iters by iteration, that is, we fix γ, and then keep increasing the max_iters by a fixed number, until the increased accuracy is less then a fixed small number ϵ; then we fix the max_iters chosen in last step, and keep decreasing γ by a fixed small number until the increased accuracy is less then ϵ; after that we turn back to fix γ and adpat max_iters. After choosing the best γ, and max_iters, we choose λ with a start of a small λ, and then keep increasing λ by $\Delta\lambda$ until the increased accuracy is less then ϵ. The result is shown in Table II.

Methods	λ	γ	max_iters	Prediction(%)
least_squares_GD	/	10^{-4}	6000	66.535
least_squares_SGD	/	10^{-4}	6000	70.966
least_squares	/	10^{-4}	6000	81.489
ridge_regression	10^{-5}	10^{-4}	6000	81.488
logistic_regression	/	10^{-4}	6000	73.914
reg_logistic_regression	10^{-5}	10^{-4}	6000	73.900

Table II
OPTIMAL PARAMETERS FOR SIX LEARNING ALGORITHMS

The idea behind this method is that: in general, for convex function, if the γ is very small and the iterations is big enough, the gradient descent algorithm can converge to the optimum with arbitrary small loss.

C. Cross Validation

As mentioned in section II-A and II-B, we use cross validation to choose the parameters γ, λ, and max_iters of the learning algorithms and the degrees in feature scaling.

In cross validation, we first randomly sort the index and divide the index into K groups averagely. Then we divide the data into K groups according to the index. After that we train and test K times, and for each time we choose $(K-1)$ groups for training and 1 group for testing. Finally, we get the generation error and its variance which show the performance of ML model.

Here is a experiment about how the prediction and its variance change with group number K. In this experiment we use least square and the result is shown in Figure 1.

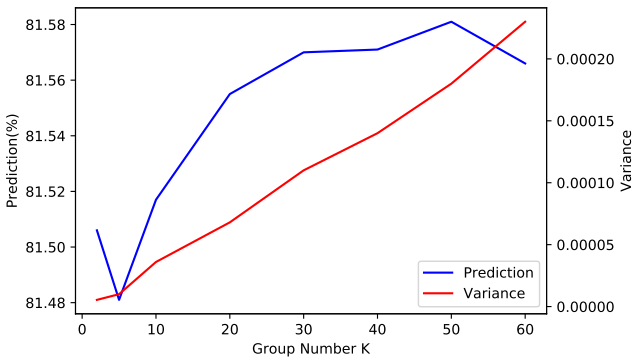

Figure 1. Prediction and Variance vs Group-Number K

As shown in Figure 1, when K is too small, the dataset is not fully used for training, so although the variance is relatively small, the generation error will be large; when K is too large, the generation error is small, but the variance will be large since the small amount of test data. According to the former discussion, we choose K equals to 20 in this task.

D. Training and Test

In this project, we first do the data preparation; then we choose the parameters of the models with the help of cross validation; finally we implement the learning algorithm on the train set.

In data preparation, we first divide the data into 8 groups and then remove the unusable data and meaningless features as described in section II-A. Finally we scale the features by the degrees shown in Table I and standardization the data.

There are four parameters we need to choose: degree of feature scaling, regularization coefficient λ, learning rate γ, and number of iterations max_iters. To evaluate the models, we use cross validation and choose K equals to 20. We using least square to choose the degree and the other three parameters by the method described in section II-B,

As shown in Table II, regulation logistic regression is the optimal learning algorithm for this task, so it is implemented on the train set and the parameters are shown in Table II.

III. RESULTS AND DISCUSSION

With the help of cross validation, the Least Squares is finally selected to be the optimal method. With the parameters in Table I and II, this method achieved an overall accuracy of 81.5%.

In data preparation, data grouping, features removing, and feature scaling are used. As shown in Table III, these three steps do improve the performance of our model.

	Group Data	Remove Feature	Scale Feature	Prediction(%)
1	-	-	-	74.443
2	✓	-	-	76.491
3	✓	✓	-	76.513
4	✓	✓	✓	81.492

Table III
PERFORMANCE OF DATA PREPARATION

For choosing the the degrees of feature scaling, the result is shown in Table I. When we try larger degrees, the predict accuracy decrease sharply due to overfitting.

As for choosing the hyperparameters of the learning models, as long as the γ is small, after enough times of iterations, the gradient descent algorithm can converge to the optimal position for convex function. Finally we choose the parameter shown in Table II.

IV. CONCLUSION

In this project, we processing the given dataset, build a polynomial based non-linear model, and tried six different learning algorithms to solve a binary classification problem. Considering the running time and accuracy, we choose to use ridge regression to obtain the final model. During the project, we also analysis the effect of different data-processing method and different parameters of learning algorithms to predict accuracy, and select the best data-processing method and learning parameters.