BLG 454E Learning From Data Fall 2021 Predicting High Resolution Brain Graph from Low Resolution Brain Graph

Oben ÖZGÜR
ITU Computer Engineering
150190719
ozguro18@itu.edu.tr

Ömer Faruk TOPAL
ITU Computer Engineering
150170029
topalo17@itu.edu.tr

Muzaffer AYDIN
ITU Computer Engineering
150170086
aydinmu17@itu.edu.tr

İhsan Mert ŞAHİN
ITU Computer Engineering
150170108
sahinih17@itu.edu.tr

Hakan TOKER
ITU Computer Engineering
150170726
tokerh@itu.edu.tr

I. Introduction

The main goal of the project is to predict high resolution brain graph from low resolution brain graph using brain connectivity matrices given in vectorized form. For this purpose, we were given low resolution train data, high resolution train data and low resolution test data. The given encodings were actually a symmetrical matrix, which shows strength of the connectivity between two different areas in brain. We've had to train on given low resolution and high resolution datasets, then test it locally with the given low resolution test file. After getting good results, we have committed them on Kaggle. Our team name on Kaggle is 150(190719_170029_170086_170108_170726). At the competition closure, we've held the 7th rank (including teacher's submission) with the score of 0.02316.

II. Datasets

A. Variance Threshold

For our training set, we have first examined our data in terms of the variances of the features and their distribution. In each of our training sets we have set up a variance threshold of 0. Therefore, features that have variance of 0 are dropped if there is any since they are unlearnable features with constant values over all samples. We have also examined larger threshold values but we observed that the performance drops across all folds. So, we have just used the basic convention of 0 threshold.

B. Local Outlier Factor

In the next phase of our data preprocessing, we have examined the outlier detection methods both on sci-kit and the last year's projects. We have decided to try "Isolation Forest" and "Local Outlier Factor" methods since it is stated that they perform well in high-dimensional data in sci-kit's documentation on outlier detection [1]. We have tried both methods on our train data and we have decided to use "Local Outlier Factor" since it produced more robust results and requires less hyperparameter tuning. For the hyperparameters, we have used the default setting given in sci-kit.

III. Methods

A. The Pipeline

Our learning pipeline consists of two main phases. First phase is to make data cleaner, more learnable and generalizable by preprocessing with the mentioned methods and second step is to regression phase to make the correct predictions. We have examined the dimensionality reduction techniques such as PCA but our performance dropped significantly across the folds since the maximum number of extracted features is lower bounded by "$min(\#samples, \#features)$" in PCA. Therefore, we have directly regressed our data upon preprocessing.

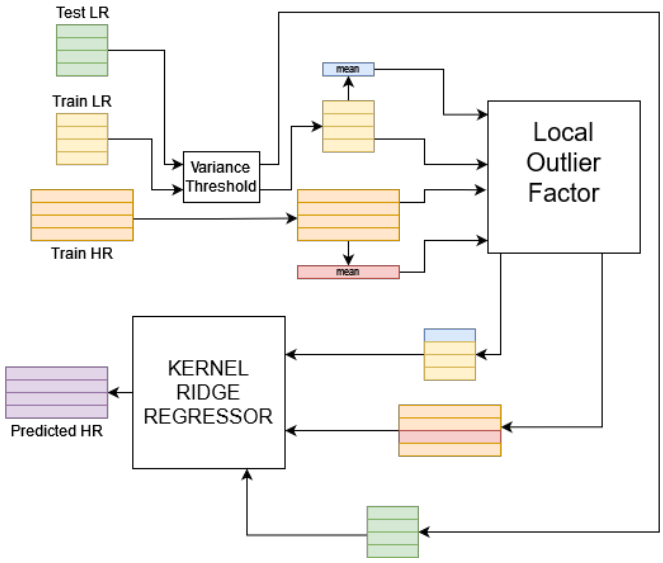

Fig. 1. The Learning Pipeline

B. Components

In the first phase of the pipeline which is preprocessing, variance threshold is applied to train data with given input value. In our case we have set it to 0 by convention but it can be tuned. Moreover, we have used "Local Outlier Factor" method to detect outliers across the training sets since it is a reliable method with high dimensional data as mentioned in the "DATASETS" section. When an outlier is detected, it is replaced by the mean of the set since the number is samples is already quite small compared to the dimension of the data.

As the regressor on regression phase, we have selected the "Kernel Ridge Regressor". Firstly, we have examined the projects and the regressors of the last year and we have realized that most of the top teams have used feature focused learning. We have first tried learning all the features separately by looking at our data and training different models for each feature but the big problem was the high dimensionality of the data both for input and the output. Therefore, we have focused on the regressors that support multi-output by its nature and defined our problem as a "Multioutput Regression Problem".

After defining our problem, we have examined many different regressors that support multioutput and searched about their advantages and disadvantages. Firstly, we have just set up a linear regression model as baseline and observed the metrics. Then, we have examined different possible regressors with different hyperparameters using 5-Fold cross validation. While searching for Ridge regression more in detail, we have seen that Ridge regression is suitable for datasets with multicollinearity and cases where samples are less than the features. Since that given features are actually representing a graph network of a brain with Pearson correlation coefficients, we thought the features should have some collinearity and tried the "Kernel Ridge Regressor" since it also allows different kernels to be used [2] along with the tuning of λ, the regularization term form $l2$ regularization.

Finally, we have tried different kernels and regularization strength by tuning λ using 5-Fold CV and trial submissions on Kaggle. We got the best results in terms of MSE by tuning $\lambda = 230$ and using the default $linear$ kernel of the "Kernel Ridge Regressor" method. Therefore, we have finalized our learning pipeline with those tunings.

IV. Results and Conclusions

The 5-Fold cross validation scores are given in Table 1 for the given set of 189 samples by our learning pipeline with metrics of mean squared error, mean absolute error and Pearson correlation coefficient between the low resolution graph and high resolution graph. Mean squared error and mean absolute errors are just different distances between the ground truth and the predicted values of our model divided by the number of elements. Only difference is mean squared error is squared as the name implies. Since the Kaggle competition is scored on mean squared error, we should expect a similar score if the model trains well. Pearson correlation coefficient defines the strength of the relationship between our predicted result and the ground truth result which is another way of measuring the strength of a model. Since the Pearson is between $0.5 - 1$ across all folds, we can conclude that there is a strong relationship and our model predicts strongly related high resolution graphs.

TABLE I
5-Fold CV Results

Fold Numbers	**Metrics**		
	MSE	***MAE***	***Pearson Correlation Coefficient***
Fold1	0.02321	0.1219	0.7052
Fold2	0.02557	0.1272	0.7294
Fold3	0.02272	0.1208	0.7200
Fold4	0.02390	0.1239	0.7360
Fold5	0.02382	0.1231	0.7197

Metrics over 5-fold CV on given train data with 189 samples with $\lambda = 250$

Our kaggle score is 0.02316 which is the mean squared error and we are ranked 7th while writing this report as team 150(190719_170029_170086_170108_170726).

References

[1] "2.7.novelty and outlier detection," scikit.[Online]. Available:https://scikit-learn.org/stable/modules/outlier_detection.html. [Accessed: 23-Jan-2022].

[2] "1.3.kernel ridge regression," scikit.[Online].Available: https://scikit-learn.org/stable/modules/kernel_ridge.html. [Accessed: 23-Jan-2022].