# Brain Graph Super-Resolution with Regression

İsmet Atabay
*Istanbul Technical University*
*Computer Engineering Department*
Istanbul, Turkey
atabayi18@itu.edu.tr

Muhammed Yavuz Köseoğlu
*Istanbul Technical University*
*Computer Engineering Department*
Istanbul, Turkey
koseoglumu18@itu.edu.tr

Emre Taşar
*Istanbul Technical University*
*Computer Engineering Department*
Istanbul, Turkey
tasare18@itu.edu.tr

***Abstract*— This paper aims to achieve high resolution output matrix using low resolution feature matrix using ml models. For this, vectors were obtained by using off-diagonal upper triangular part of low- and high-resolution matrices. These include performing PCA on them and training them with Bayesian Ridge Regression.**

***Machine Learning Models, Up Sampling, Multioutput Regressor***

## I. INTRODUCTION

The brain might be thought of as an interconnected system consists of linked nodes which are responsible from specific physiological tasks in a living body. In this interconnected system, brain regions communicate with each other by conveying neural signals and the connectivity between these regions is essential to perceive and clarify the neurological system of a human being. Disorders that huge number of people have been suffering can be caused by the any brain connectivity impairment. Therefore, extracting and processing brain connectome is crucial to better understand and treat these disorders. Nevertheless, processing this information is a challenging task because of limited data and expensive accessibility to it. For this reason, we as a team with the name 150180021_150180030_150180054 proposed and implemented an approach to brain graph super resolution technique which is the process of obtaining a high-resolution brain graph from a given low-resolution brain graph. In our approach, we achieved to predict a connectivity matrix with the size $\mathbf{X}_{\boldsymbol{HR}} \in \mathbb{R}_{268\times268}$ by using a given connectivity matrix with the size $\mathbf{X}_{\boldsymbol{LR}} \in \mathbb{R}_{160\times160}$ for each 189 sample. Proposed approach by our is evaluated with mean squared error, and our result is outperformed any other approach by ranking 1st in Kaggle class competition with the 0.02298 score.

## II. DATASET

In the dataset, low resolution brain connectivity matrix is given as input $\mathbf{X}_{\boldsymbol{HR}} \in \mathbb{R}^{268\times268}$. The data in the matrix shows the connectivity strength on that point. There is a high resolution matrix as output in the data set $\mathbf{X}^{\mathrm{HR}} \in \mathbb{R}$ 268 ×268. This data has been vectorized as an off-diagonal upper triangular part of $\mathbf{X}^{\mathrm{LR}}$ and $\mathbf{X}^{\mathrm{HR}}$ as feature vectors $\mathbf{X}^{\boldsymbol{LR}} \in \mathbb{R}^{1\times12720}$ and $\mathbf{X}^{\boldsymbol{HR}} \in \mathbb{R}^{1\times35778}$. As a task, the following mapping process is expected to be performed.

$$f: \mathbb{R}^{12720} \rightarrow \mathbb{R}^{35778}$$

$$f(\boldsymbol{x}^{\boldsymbol{LR}}) = \hat{\mathbf{X}}^{\boldsymbol{HR}} \approx \boldsymbol{x}^{\boldsymbol{HR}}$$

When the input vector is considered in the data set, it is seen that the size of the vector is too much for training the model. (1x12720) If an attempt is made to train the model with an input of this size, it causes a huge waste of time and memory. For this, the feature vector dimension has been reduced as much as possible by using the Principal Component Analysis (PCA) method. With this method, it is aimed to reduce the feature by keeping the variance[1] in the data set at the highest level. Since the number of reduced components cannot be more than the number of samples or features, the number of components parameter is set to 180. Before the PCA method, a score function was written and tested with the GenericUnivariateSelect method, which takes the average in the train set. As a result, it was decided not to use this method since there was a decrease in the MSE score measured on Kaggle.

## III. METHOD

In order to obtain the desired High resolution output in the trained model, regression was applied for each target vector in the $\mathbf{X}^{\mathrm{HR}}$ matrix on the data set. For this process, the "MultiOutputRegressor" method of the scikit-learn library was used.

Bayesian Ridge Regression, which is a probabilistic linear model, was used for the regression model. The reason for using Bayesian Ridge Regression, which is a model similar to the classical ridge model, is that it can adapt to the data set and allows the use of regularization parameters. In addition, although it is a model that can work slowly in terms of time, this model has been used because we do not have a time problem on Kaggle. Before the data was inserted into the model, the outliers were determined using the LocalOutlierFactor method. The number of features has been reduced to 180 with the help of the data PCA processed through this process. The model is trained with the new $\mathbf{X}^{\mathrm{LR}}$ matrix vectors obtained from here. Pipeline is indicated in Figure1.

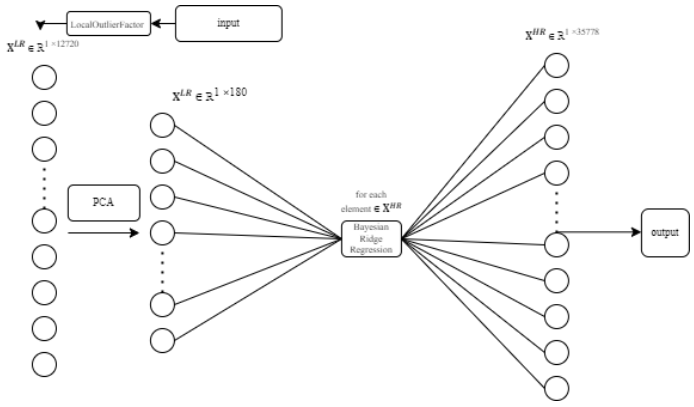


Fig. 1. Learning Pipeline

In the parameters, the parameters assigned by scikit learn by default (i.e. n_iter = 300) are used. Afterwards, trainings were made by changing the iteration number and regularization term in these parameters, but it was seen that the parameters that gave the best results in the Mean Squared Error (MSE) values obtained were the default parameters.

## IV. Results and Conclusion

To conclude, different approaches can be efficient for brain graph super resolution technique. Although vary regression models such as Adaptive Boosting Regression(AdaBoost), Lasso, Support Vector Regression(SVR), KNearest Neighbors Regression(KNR) were trained, minimum MSE and Mean Distance Error(MAE) were obtained by using Bayesian Ridge Regression with. Error values are given in the table below.

| **Models** | **Errors** | |
|---|---|---|
| | ***Mean Squared Error*** | ***Mean Absolute Error*** |
| Bayesian Ridge | 0.024 | 0.012 |
| AdaBoost | 0.027 | 0.013 |
| SVR | 0.029 | 0.013 |
| Lasso | 0.031 | 0.017 |
| KNR | 0.034 | 0.021 |

In order to reduce processing load and shorten the training time PCA is used. Under favour of PCA's ability of creating new features that quantitively fewer than original features but have same functionallity features, training can be performed without any loss of accuracy.

Cross validation divides dataset into desired number of folds. 5 fold cross validation was used in this project. Dataset divided into 4 training fold and 1 validation fold. MSE values are calculated in each of which combinations. Hence trained model become more generalizable and stable.

## References

[1] Mishra, Sidharth & Sarkar, Uttam & Taraphder, Subhash & Datta, Sanjoy & Swain, Devi & Saikhom, Reshma & Panda, Sasmita & Laishram, Menalsh. (2017). Principal Component Analysis. International Journal of Livestock Research. 1. 10.5455/ijlr.20170415115235. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.