Graph Representation Learning Report

Zhi Ye zhi@food.ku.dk
Faculty of Science, University of Copenhagen, Denmark

I. Introduction

Autism Spectrum Disorder is a neural developmental disorder and becomes a global burden on global healthcare. Behavioral observations and self-reported symptoms are the most standard diagnostic criteria for Autism [1]. However, owing to overlapping symptoms with other mental disorders and non-psychiatric disorders, sometimes psychiatrists have difficulties distinguishing them only via questionnaires and observation. Magnetic resonance imaging (MRI) is a biomedical imaging technique to provide high-quality images of physiological activity inside the human body. The resting-state functional MRI time-series data records the blood-oxygen-level-dependent (BOLD) contrast when subjects are resting. The BOLD reflects many intrinsic brain activities because cerebral blood flow affects neuron activation: blood flows to the brain region which was functioning [2]. Resting-state functional MRI has three spatial dimensions and one time-series dimension. Owing to the characteristic that the brain regions have spontaneous fluctuations of the BOLD signal even without external stimulation, resting-state functional MRI was used to detect the brain's abnormal functional regions.

Previous neuroimaging studies illustrated abnormal activity patterns from resting-state functional MRI in Autism patients [3]. However, because of functional MRI's high-dimension and high-noise properties, it is difficult to find suitable statistical models and their analysis results are inconsistent. Meanwhile, prior studies demonstrated that functional connectivity network in functional MRI reveals latent correlations between distant regions, their BOLD signals were correlated across time courses [4]. Functional connectivity is a measurement of synchronized fluctuation in brain signals from functional MRI. The standard pipeline started with the parcellation of informative brain regions from resting-state functional MRI, then the brain signal at selected regions was extracted by averaging time-series data across voxels. To obtain the parcellation, a novel method, independent components analysis (ICA) [5] was employed to compute networks among informative brain regions, as well as reduced dimensions and de-noised at the same time. After that, correlations between different regions were calculated to obtain a functional connectivity matrix. The matrix indicated a brain network that each node was a region connected by edges, these edges were correlations between regions.

In our study, the functional connectivity matrix is precisely an adjacency matrix of a graph, that the graph consists of different brain regions as nodes and the correlation as edges. Now we convert the fMRI classification task to a graph-level prediction task and compute graph theory metrics degree centrality and efficiency as node features. Then we learn node embedding from the graph attention networks (GATs) and graph convolutional networks (GCNs), and pooled these embedding as features of the graph. After the feature extraction, we predict the autism status of subjects based on these features extracted by resting-state functional MRI and graph neural networks (GNN).

II. Methods

A. Graph Theory Metrics

We used two graph theory metrics, degree centrality and efficiency as our initial node features. The degree centrality of node i is the sum of edges' weight from one node:

$$s_i = \sum_{j}^{N} w_{ij} \tag{1}$$

where w is the weighted adjacency matrix, in which w_{ij} is non-zero if the node i is connected to node j, and the value represents the weight of the edge.

Efficiency is a measurement of how efficiently it exchanges information. The efficiency between two nodes i and j is $\frac{1}{d(i,j)}$, the $d(i,j)$ is the distance. The efficiency of node i is the sum of efficiency between all nodes connected with i :

$$e_i = \sum_{j \neq i}^{N-1} \frac{1}{d_{ij}} \tag{2}$$

B. Node Embedding and Graph Neural Networks (GNN)

In representation learning, feature engineering is not necessary since the aim of algorithms is to automatically learn the features. For graphs, our goal is to learn the task-independent node representation embedding efficiently, because it could encode the whole network information in a lower-dimension space and contain their spatial information at the same time. For example, the similarity between nodes is kept in the embedding space in the representation learning process. Then the low dimensional embedding vector is potentially used as a feature to represent nodes or graphs to solve classical machine learning problems, such as regression, classification, and clustering.

To achieve this goal, the node embedding method encodes nodes in latent space which can approximate their similarity in the graph. According to the framework, encoders map nodes from graph to embedding and decoders map embedding to the similarity score.

Encoder for node i maps it to a vector:

$$ENC(i) = z_i \tag{3}$$

A similarity function $similarity(i, j)$ is then used to evaluate the similarity between node i and j in the original graph. The decoder is based on node similarity. It specifies how good the mapping from embedding vectors to the original networks:

$$similarity(i, j) = z_j^T z_i \tag{4}$$

However, this approach Equation(3) is a shallow encoder that has some limitations such as ignoring the node features. With the development of deep neural networks, the graph neural networks (GNN) tackled this challenge and provide us with an efficient way to embed nodes, subgraphs, and even graphs. The GNN constructs a computation graph from the neighborhood around a node, it contains multiple layers of non-linear transformations based on graph structures. Then GNN is trained to propagate information across the graph to update the node features. This approach is not sensitive to node ordering and graph sizes because the node embedding is only computed from its local network neighborhoods. Every time the embedding is updated by aggregating messages from neighbors and averaging them. Hence, we used a Graph Convolutional Networks (GCN) [6] and initiate the node features with its graph theory metrics, degree centrality and efficiency:

$$h_i^0 = x_i = [s_i, e_i] \tag{5}$$

where the h_i^0 is the initial 0-th layer embedding is exactly the node feature. Then the embedding is updated in the neural network as:

$$h_i^{k+1} = \sigma(\sum_{j \in N(i)} w_k \frac{h_k}{|N(i)|}) \tag{6}$$

Another seminal graph neural network we used in this study is the Graph Attention Networks (GAT) [7]. The embedding way is modified as:

$$h_i^{k+1} = \sigma(\sum_{j \in N(i)} \alpha_{i,j} w_k \frac{h_k}{|N(i)|}) \tag{7}$$

where the $\alpha_{i,j}$ is the weighting factor of node i's message to node j. In GCN, nodes' neighbors are not treated equally important. The weight factor $\alpha_{i,j}$ focus more on essential part of training data.

After the node embedding, we use all node embedding in the graph and group mean pooling them to do the graph-level prediction, which is predicting the Autism status of each graph.

C. Baseline and Data Splits

We used a Support Vector Machine with a linear kernel as a baseline model [8]. It takes vectorized lower triangular part of the correlation matrix as input.

The MRI dataset was split into a training set, validation set, and test set, which contains 48, 16, and 16 MRI respectively. The training set was used to select the optimizing weights of our graph neural networks, the validation set was used to tune the hyperparameters and we evaluated our model on the testing set. We balanced the dataset in training, validation, and test sets so all of these sets are balanced between Autism and health control.

III. Data

The resting-state functional MRI data is from the Autism Brain Imaging Data Exchange (ABIDE) and was preprocessed by the Preprocessed Connectomes Project (PCP) [9]. The ABIDE consists of 539 subjects with Autism and 573 healthy controls. We only include subjects ages 21 to 33, because these subjects' physical conditions are at a good level in their life. Then we employed propensity score matching (PSM) to eliminate the bias from age and gender, and match groups to build a balanced dataset since age and gender are common confounding variables for mental disorders and brain MRI. After the cohort selection step, we include a total of 80 subjects in our cohort, 41 of them are autistic and the rest 39 are health control. The distribution of available subjects with phenotype information is shown in Table I:

Group	Age (mean ± std)	Female	Subjects
Control	26.18 ± 3.15	2	39
Autism	26.01 ± 3.76	1	41

Table I
AGE AND GENDER DISTRIBUTION OF AVAILABLE SUBJECTS.

All MRI data had the same dimension 61 × 73 × 61 × 195. We used ICA to extract 25 independent components. The atlas of ICA result is shown in Fig 1, then functional connectivity matrices were calculated between ICs. It is a symmetric matrix with the correlation between each IC as elements, so the diagonal of it is 1. The dimension of functional connectivity matrices for each sample was 195 × 25. The functional connectivity network is a complete graph since we computed the correlation between each pair of ICs, and the edge weight is possible to be negative because there are negative correlations between different ICs. To simplify the problem and obtain a sparse graph, we only include the edges with positive values to build our functional connectivity networks.

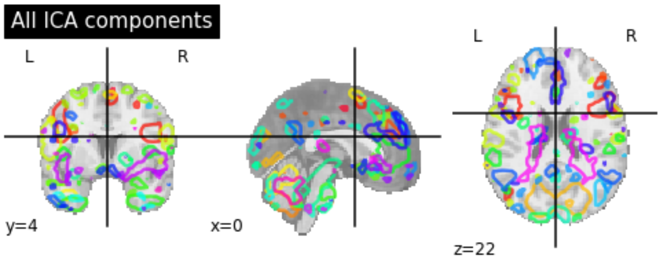

Figure 1. Group-ICA map with 25 components. Circles with the same color indicate brain regions that exhibit similar BOLD fluctuations over time.

Most previous neuroimaging studies flattened the half triangular part of the functional connectivity matrices and use the vectorized correlation matrix as features [10] [10]. The input feature's dimension was reshaped to the number of samples × 300. While it constructed a 1-D feature array that represent the network, it did not detect the latent variable of the network.

IV. Results

The MRI data downloading and processing procedure were implemented on the platform of Nilearn [11] (https://nilearn.github.io/), Networkx [12] (https://networkx.org/) and ScikitLearn [13] (https://scikit-learn.org/) in Python. The graph neural networks frame was implemented on the platform of PyTorch Geometric [14] (https://pytorch-geometric.readthedocs.io/) in Python. Part of the PyTorch Geometric code was referred from DTU 2022 Summer School 02901 Advanced Topics in Machine Learning: Graph Representation Learning (http://www2.compute.dtu.dk/courses/02901/).Our code is available at (https://github.com/zhiye9/GRL_project).

We run experiments with 5 models including a baseline linear SVM and 4 GNNs. Models are trained on the training set, parameters tuned on the validation set, and evaluated on the test set. The structure of GNN is refer to the GCN PyTorch framework from GRL lecture "GNN Introduction". We learn the node embedding by stacking GCNs or GANs together and use global mean pooling and linear transformation to do the classification. Network GAN6 and GAN3 indicate that there are 6 or 3 GAN modules imported from PyTorch Geometric packages stacked together to learn the node embedding. GCN6 and GCN3 simply replace GANs with GCNs with different amounts of modules. We set the learning rate to 0.0001 to avoid it converging too early and decrease the dropout rate to 0.15 since we have a small graph with only 25 nodes. We trained each model with 100 epochs owing to the limitation of computation resources. Binary Cross Entropy was used as a loss function and the model was evaluated on the test set with Area Under the ROC curve (AUROC) in Table II.

Models	AUROC (%)
GAN6	63.49
GAN3	58.73
GCN6	60.32
GCN3	53.12
Linear SVM	57.81

Table II
Model performance on test set with AUROC.

V. Discussion

Classification results from Table II illustrated clearly that in our study, with the increasing network's depth with the same module, the model's performance becomes better. A deeper GCN performs better than a shallow one and similar results are found in GAN. One of the reason could be with a deeper structure, more neighborhood information are aggregated in our network. Especially in our study, the correlation matrix of MRI data is not very sparse. Then it is also obvious that GANs have better performance than GCN. The most important reason is that the GCN did not use edge features as their inputs but GAN utilizes them [15]. Because we parcelled resting-state functional MRI into nodes by their correlation and regard the correlation as edges, it is a principal feature in our graphs. Treating edges as unweighted lost a lot of important information in the brain network.

The baseline model used vectorized correlation matrix as input and did not obtain information from graph structures. An SVM with a linear kernel was selected to find the hyperplane between Autism vectors and health control vectors and do the classification. It had a medium performance in all models and is even better than shallow GCNs. The linear SVM has low computation consumption and it is good at classifying 1-D vectorized data, and this is why it was popularly used in neuroimaging analysis. In summary, GNNs show great potential in graph structure data and neuroimaging are natural datasets with a network structure [16], our study implies that simple GNNs can beat traditional machine learning classifiers in mental disorder prediction tasks and it still can be improved to a large extent. A big obstacle in this study is that we only have limited samples and computing resources, and we would like to predict a complex mental disorder with high-dimension data. 80 subjects are too few for deep learning and the number of epochs during training still needs to be increased. In the future, more samples need to be included in the cohort and analyzed. With more computing resources, we could apply more complex networks with enough training epochs to improve our GNNs' performance.

References

[1] H. Hodges, C. Fealko, and N. Soares, “Autism spectrum disorder: definition, epidemiology, causes, and clinical evaluation,” *Translational pediatrics*, vol. 9, no. Suppl 1, p. S55, 2020.

[2] N. K. Logothetis, J. Pauls, M. Augath, T. Trinath, and A. Oeltermann, “Neurophysiological investigation of the basis of the fmri signal,” *nature*, vol. 412, no. 6843, pp. 150–157, 2001.

[3] W. K. Lau, M.-K. Leung, and B. W. Lau, “Resting-state abnormalities in autism spectrum disorders: a meta-analysis,” *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[4] A. Shmuel and D. A. Leopold, “Neuronal correlates of spontaneous fluctuations in fmri signals in monkey visual cortex: implications for functional connectivity at rest,” *Human brain mapping*, vol. 29, no. 7, pp. 751–761, 2008.

[5] G. Varoquaux, S. Sadaghiani, P. Pinel, A. Kleinschmidt, J.-B. Poline, and B. Thirion, “A group model for stable multi-subject ica on fmri datasets,” *Neuroimage*, vol. 51, no. 1, pp. 288–299, 2010.

[6] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.

[8] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[9] C. Craddock, Y. Benhajali, C. Chu, F. Chouinard, A. Evans, A. Jakab, B. S. Khundrakpam, J. D. Lewis, Q. Li, M. Milham *et al.*, “The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives,” *Frontiers in Neuroinformatics*, vol. 7, p. 27, 2013.

[10] G. Varoquaux and R. C. Craddock, “Learning and comparing functional connectomes across subjects,” *NeuroImage*, vol. 80, pp. 405–415, 2013.

[11] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi, A. Gramfort, B. Thirion, and G. Varoquaux, “Machine learning for neuroimaging with scikit-learn,” *Frontiers in neuroinformatics*, vol. 8, p. 14, 2014.

[12] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using networkx,” Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[15] L. Gong and Q. Cheng, “Exploiting edge features for graph neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9211–9219.

[16] X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang, D. Scheinost, L. Staib, P. Ventola, and J. Duncan, “Braingnn: Interpretable brain graph neural network for fmri analysis,” *bioRxiv*, 2020.