Decoding Health: Predictive Modeling of Diseases through Brain Activity Data and Graph Structure Learning

Abduragim Shtanchaev [1]

Abstract

Graph Neural Networks (GNNs) have emerged as the de facto solution for predicting on graph-structured data. However, the assumption that the graph structure (i.e., the adjacency matrix) is given and true is often not valid in practice, as graph structures are often created by humans and may contain noise. This can potentially degrade the performance of GNNs. Moreover, there are real-world scenarios, such as physical particles interacting or fMRI data readings from the brain, where the underlying graph structure is latent and not predefined. In this work, we explore the prediction of graph structures from brain signals using GNNs, aiming to address the challenge of predicting graph structures from noisy or intermingled signals.

1. Introduction

It is a commonly held assumption within the field of deep learning that data is typically represented in the form of grids, such as images and word tokens. However, this perspective may not fully capture the complexity of objects in the real world, which can often be more accurately modeled using graphs. Graphs, which consist of nodes or vertices interconnected by edges, provide a versatile and intuitive representation that can better capture the intricate relationships and interactions among elements in complex systems.

In fact, graphs can be observed at various levels of organization in the living world, ranging from the fundamental building blocks of organisms, such as organic molecules, which can be effectively represented as graphs with atoms as nodes connected by chemical bonds as edges, to the intricate organization of neurons in the human brain that underlies our cognitive processes. Thus, considering the ubiquity of graphs in natural systems, they offer a powerful framework for representing and analyzing complex data in deep learning tasks, beyond the limitations of grid-based representations commonly employed in traditional deep learning approaches.

Graph **N**eural **N**etworks(Defferrard et al., 2016)(**GNN**s) are a canonical and extensively utilized method for modeling data that are inherently structured as graphs. For instance, GNNs have demonstrated remarkable performance in recommendation systems(Wu et al., 2022), enabling personalized recommendations in various domains such as e-commerce, social media, and online advertising. Additionally, GNNs have been employed in optimizing routing and transportation systems (Derrow-Pinion et al., 2021), where the inherent graph structure of the road network or transportation network can be leveraged to find optimal routes efficiently. Furthermore, GNNs have been utilized in drug discovery (Cheung & Moura, 2020), where the graph representation of molecular structures allows for effective prediction of their properties and activities. Notably, GNNs have also been employed in the cutting-edge domain of chip design for AI (Mirhoseini et al., 2021), where they have been shown to surpass human-designed chips in terms of performance and efficiency, highlighting the significant potential of GNNs in advancing various practical applications.

GNNs are a widely used technique for learning from graph-structured data, where the graphs are assumed to be static and known upfront. However, this assumption of a fixed input structure is often solely based on the intuition and expertise of the machine learning practitioner, which may not necessarily be optimal for the specific task at hand. This places a burden on the practitioner to determine the most appropriate graph structure to use.

Consider, for example, the case of physical particles interacting with each other, where the forces between particles can only be inferred from their motion. In such cases, particles and forces can be represented as nodes and edges in a graph, but it is not straightforward to determine the appropriate assignment of edges in the graph. Ideally, the adjacency matrix A, which represents the edges in the graph, should be inferred from the data itself, a process commonly referred to as *"latent graph inference"* (Brugere, 2015). Once the

[*]Equal contribution [1]Computational and Data Science and Engineering, Skoltech, Moscow, Russia. Correspondence to: Abduragim Shtanchaev <abduragim.shtanchaev@skoltech.ru>.

Proceedings of the 40^{th} International Conference on Machine Learning, PMLR 139, 2023.

adjacency matrix A is inferred, it can be used as input to the GNN for further processing.

The human brain can be scanned using various medical imaging techniques (see Figure 1), including MRI, EGG, and PET. Among these, MRI data, particularly functional MRI (fMRI) and Diffusion Tensor Imaging (DTI), are widely used for brain analysis research. fMRI captures correlations between time-series signals of brain regions, while DTI models the physical connectivity between gray matter regions in structural brain networks (Neurosurg, 2016).

Numerous techniques have been proposed to evaluate brain connectivity between pairs of nodes, with one of the most commonly employed methods in the field of neuroimaging involving pairwise correlations between time-series data obtained from Regions of Interest (ROIs). Other approaches encompass partial correlations, mutual information, coherence, and Granger causality. Following the selection of a Functional Connectivity (FC) measure, the strength of connectivity between each pair of ROIs can be assessed.

Nonetheless, it is worth noting that such simplistic heuristic correlation measures may not always accurately capture the underlying graph structure of brain activity, potentially leading to reduced performance of GNNs trained on such graphs.

In this study, we aim to investigate the efficacy of such heuristic measures utilized for constructing brain ROIs connectivity graph, and how they impact the final performance. Subsequently, we propose a data-driven method for learning graph connectivity without relying on handcrafted proximity measures, and compare it with standard approaches.

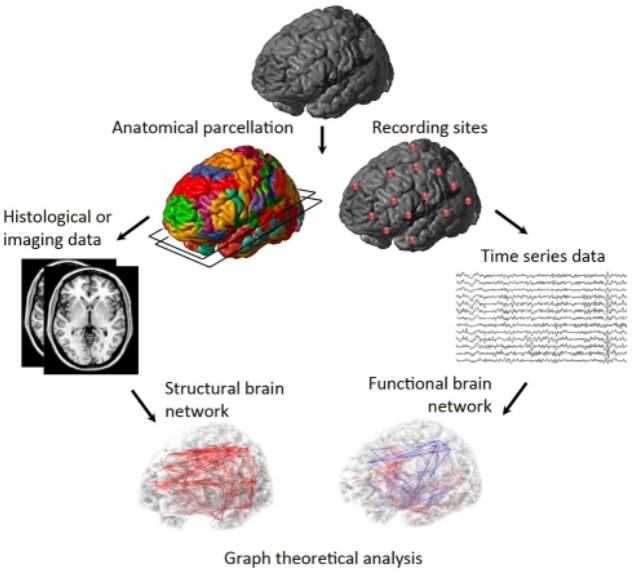

Figure 1. Graph analysis to brain networks. Structural (including either gray or white matter measurements using histological or imaging data) or functional data (including resting-state fMRI, fMRI, EEG, or MEG data) is the starting point. Nodes are defined (e.g., anatomically defined regions of histological, MRI or diffusion tensor imaging data in structural networks or EEG electrodes or MEG sensors in functional networks) and an association between nodes is established (coherence, connection probability, or correlations in cortical thickness). The pairwise association between nodes is then computed, and usually thresholded to create a binary (adjacency) matrix. A brain network is then constructed from nodes (brain regions) and edges (pairwise associations that were larger than the chosen threshold)(Elgoyhen et al., 2012)

2. Related Work

There exists a notable body of literature that endeavors to address the challenge of link adjacency problems through various strategies. For instance,(Kipf et al., 2018) propose a method that leverages variational inference to model interacting systems . In this approach, the encoder predicts the probability distribution for edge existence between nodes, and utilizes the Concrete distribution for sampling to enable backpropagation. Meanwhile, the decoder is tasked with predicting the future continuation of the interacting system's trajectory. (Fatemi et al., 2021) propose a solution that employs classification supervision for learning graph structure, along with a self-supervised branch of the network to mitigate the issue of supervision starvation. Counter to the rest of the work (Liu et al., 2022) suggests completely self-supervised approach to graph structure learning. It relays on mutual information maximization objective between learner and anchor (teacher) views of a network.

In order to measure functional connectivity(graph connectivity), it is common practice to preprocess fMRI time series data, which may involve performing detrending, demeaning, and whitening of the fMRI BOLD time series data at each voxel.(Wang, 2016). After such reprocessing one of the simplest and most frequently used methods in the neuroimaging community is via pairwise correlations between BOLD time courses from two ROIs. Other methods include partial correlations (Wang, 2016), mutual information, coherence, Granger causality (S. M. Smith & Woolrich, 2011).

3. Method

Formally, in the task of brain analysis, the input is a brain network dataset $\mathcal{D} = \{\mathcal{G}_n, \mathcal{Y}_n\}_{n=1}^{N}$ that consists of N subjects. $\mathcal{G}_n = \{\mathcal{V}_n, \mathcal{E}_n\}$ represents brain network of each subject n and $\mathcal{Y}_n$ is ground truth label, for example such as neural disease. Brain network $\mathcal{G}_n$ for every subject involves same size set of M nodes defined by the ROIs on a particular brain parcellation, $\forall_n, \mathcal{V}_n = \mathcal{V} = \{v_i\}_{i=1}^{M}$, $v_i \in \mathbb{R}^d$ column vector representing brain activity intensity where d

time span.

As it was mentioned earlier, normally edge connections $\mathcal{E}_n$ for brain network $\mathcal{G}_n$ of each subject are estimated using signal correlation measures ROIs of a brain and are different for each subject. $\mathcal{E}_n$ is generally represented by an adjacency matrix $\mathbf{W}_n \in \mathbb{R}^{M\times M}$ describing strengh of connection of each node - ROI. In this work we will examine two scenarios:

1. $\mathcal{E}_n$ given (calculated using procedure described in section 2). Here the task is to predict target variable $\mathcal{Y}_n$ for each subject by training GNN
2. $\mathcal{E}_n$ is *not* given. The task is to estimate graph structure $\mathcal{S}_n$ such that the estimated $\mathcal{S}_n$ will be more optimal for training GNN to predict $\mathcal{Y}_n$.

3.1. Predicting $\mathcal{Y}_n$ with a given graph structure $\mathcal{E}_n$

Generally $\mathcal{E}_n$ is a real valued matrix of $\mathbb{R}^{M\times M}$. In order to create edge between nodes the matrix needs to be binarized by the threshold τ - which is a hyper-parameter.

$$\mathcal{E}_n^{\text{binary}}(i,j) = \begin{cases} 1 & \text{if } \mathcal{E}_n(i,j) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

After binarization procedure dataset $\mathcal{D} = \{\mathcal{E}_n^{\text{binary}}, \mathcal{V}_n\}_{n=1}^N$ is used to train graph neural network $\mathcal{F}_\theta(\mathcal{E}_n^{\text{binary}}, \mathcal{V}_n) = \hat{\mathcal{Y}}_n$ by optimizing objective function $\mathcal{L}(\mathcal{Y}_n, \hat{\mathcal{Y}}_n)$

$$\arg\min_{\tau,\theta} \mathcal{L}(\mathcal{Y}_n, \hat{\mathcal{Y}_n}) = \arg\min_{\tau,\theta} \left[-\frac{1}{N} \sum_{n=1}^{N} \mathcal{Y}_n \cdot \log \hat{\mathcal{Y}}_n \right] \tag{2}$$

It should be noted that since binarization procedure is not differentiable τ is selected iteratively in the optimization process.

3.2. Estimating graph structure $\mathcal{S}_n$

As in this setting graph structure is not provided and thus must be inferred using only brain activity information $\mathcal{V}_n$ stored at each node $v_i, v_i \in \mathcal{V}_n = \{v_i\}_{n=1}^M$. In this work we employ self-supervised graph structure prediction method suggested by SUBLIME (Liu et al., 2022). Over all the pipeline suggested in SUBLIME is following:

1. Predict adjacency $\hat{\mathcal{S}}$ matrix using structure learner $\hat{\mathcal{S}} = p_w(X)$, where $X \in \mathbb{R}^{M\times d}$ and d is a feature vetctor dimensionality
2. Process $\hat{\mathcal{S}}$ using post-processor $q(\hat{\mathcal{S}}) = \mathcal{S}_l-$ a Learner view
3. Bootstrap $\mathcal{A}_a$ from $\mathcal{S}$, $\mathcal{A}_a \leftarrow \tau\mathcal{A}_a + (1-\tau)\mathcal{S}_l$. $\mathcal{A}_a = I$ is initialized as an identity matrix in the beginning of training
4. Augment on both views $\mathcal{A}_a$ and $\mathcal{S}_l$ and data matrix X.

$$\widetilde{X} = \mathcal{T}_{aug}(X)$$
$$\widetilde{\mathcal{A}_a} = \mathcal{T}_{aug}(\mathcal{A}_a)$$
$$\widetilde{\mathcal{S}_l} = \mathcal{T}_{aug}(\mathcal{S}_l)$$

5. Apply GNN encoder f_θ followed by a project g_ϕ on both views to get final embedding of each node.

$$\mathcal{Z}_l = g_\phi(f_\theta(\widetilde{X}, \widetilde{\mathcal{S}_l}))$$
$$\mathcal{Z}_a = g_\phi(f_\theta(\widetilde{X}, \widetilde{\mathcal{A}_a}))$$

6. Calculate contrastive loss between graph strucure in Anchor view - $\mathcal{A}_a$ and Learner view - $\mathcal{S}_l$

$$\mathcal{L} = \frac{1}{2n} \sum [l(\mathcal{Z}_{l,i}, \mathcal{Z}_{a,i}) + l(\mathcal{Z}_{a,i}, \mathcal{Z}_{l,i})]$$

$$l(\mathcal{Z}_{l,i}, \mathcal{Z}_{a,i}) = \log \frac{e^{sim(\mathcal{Z}_{l,i}, \mathcal{Z}_{a,i})/\ t}}{\sum_{k=1}^{n} e^{sim(\mathcal{Z}_{l,i}, \mathcal{Z}_{a,k})/\ t}}$$

However, SUBLIME is only capable of prediction structure for a single graph and therefore is not suitable for out task at hand in its original form. In this this work we examine:

1. Possibility of extending SUBLIME to work on multiple graphs simultaneously
2. Compare graph structure predicted by SUBLIME with methods that employ heuristics to find such structure

Namely, to make SUBLIME work on multiple graphs simultaneously, we experiment with randomly fetching a feature matrix X of subject n during training.

4. Experiments

4.1. Data

In this work fMRI brain readings from Cobre dataset are used for trains models and evaluating our models. The dataset contains brain scans of 163 patients. Patient in this dataset are either healthy, have schizophrenic disorder or are inclined towards schizophrenia. The task it to classify if patient is health, schizophrenic or inclined to schizophrenia.

Brian reading that are used in this study are already prepossessed and ready for use. For each patient we have 116 ROIs (Regeon of Interest). Each **ROI**s contains 157 samples of data recorded over time. These **ROI**s readings are converted

to Adjacency matrix for each patient using method mentioned in section 2 and therefore can be used to train GNNs. Produced Adjacency matrices are real- valued, hence need to be binarised. We thresholding technique with threshold τ as explained in section 3.1

4.2. Settings

We run experiments on single NVIDIA GeForce GTX 1080 Ti with 11 Gigabytes of memory. For training we set following parameters learning rate = 0.003, batch size = 20. We also found that setting drop out to 0.5 improves metrics. Because dataset we are using is small we uses cross-validation techniques to validate our model more fairly. We set number of folds to 7.

4.3. Results for Predefined Structure $\mathcal{E}_n$

Following experiments are designed to illustrate how parameter τ which is used to threshold Adjacency matrix $\mathbf{W}_n \in \mathbb{R}^{M \times M}$ for binarization affects the final performance of a model. Figure 2 shows accuracy metric result of a trained GNN model using cross validation on 7 folds. It can be seen that different values of τ result in different model performance metrics.

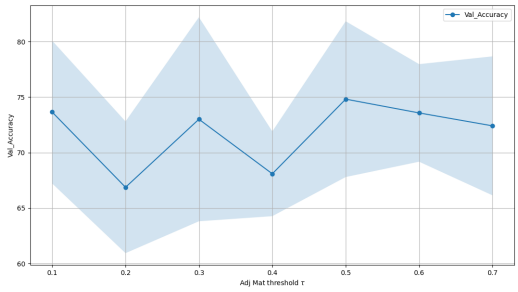

Figure 2. Accuracy results for cross validation on 7 folds for GNN model. Dark blue line is the mean of accuracy results folds and light blue region shows uncertainty expressed by standard deviation. Y axis shows threshold values used for binarizing Adjacency matrix

Figure 3 shows F1 score metrics of the same GNN and Logistic Regression model for comparison. It also should be noted that uncertainty of model is huge.

5. Analysis

6. Conclusion

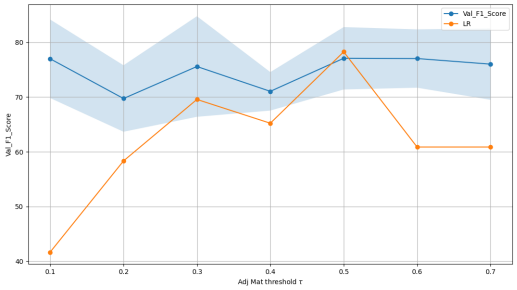

Figure 3. F1 score (weighted) results on cross validation using 7 folds. Blue line depicts mean results of GNN model and orange line corresponds to Logistic Regression. Light blue region shows uncertainty. Y axis shows threshold values used for binarizing Adjacency matrix

References

Brugere, I. Latent graph inference and validation. *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1494–1495, 2015.

Cheung, M. and Moura, J. M. F. Graph neural networks for covid-19 drug discovery. *2020 IEEE International Conference on Big Data (Big Data)*, pp. 5646–5648, 2020.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.

Derrow-Pinion, A., She, J., Wong, D., Lange, O., Hester, T., Perez, L., Nunkesser, M., Lee, S., Guo, X., Wiltshire, B., Battaglia, P. W., Gupta, V., Li, A., Xu, Z., Gonzalez, A. S., Li, Y., and Velickovic, P. Eta prediction with graph neural networks in google maps. oct 2021. doi: 10.1145/3459637.3481916.

Elgoyhen, A., Langguth, B., Vanneste, S., and Ridder, D. Tinnitus: Network path physiology-network pharmacology. *Frontiers in systems neuroscience*, 6:1, 01 2012. doi: 10.3389/fnsys.2012.00001.

Fatemi, B., Asri, L. E., and Kazemi, S. M. Slaps: Self-supervision improves structure learning for graph neural networks, 2021.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems, 2018.

Liu, Y., Zheng, Y., Zhang, D., Chen, H., Peng, H., and Pan, S. Towards unsupervised deep graph structure learning, 2022.

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E. M., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., Pak, J., Tong, A., Srinivasa, K., Hang, W., Tuncer, E., Le, Q. V., Laudon, J., Ho, R., Carpenter, R., and Dean, J. A graph placement methodology for fast chip design. *Nature*, 594:207 – 212, 2021.

Neurosurg, J. Functional mri resting state fmri and dti for predicting verbal fluency outcome following resective surgery for temporal lobe epilepsy. *Some Journal*, pp. 929–937, 2016.

S. M. Smith, K. L. Miller, G. S.-K. M. W. C. F. B. T. E. N. J. D. R. and Woolrich, M. W. Network modelling methods for fmri. *NeuroImage*, 54:875–891, 2011.

Wang, Y., K. J. K. P. B. . G. An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation. *Frontiers in neuroscience*, 10, 2016.

Wu, S., Sun, F., Zhang, W., Xie, X., and Cui, B. Graph neural networks in recommender systems: A survey, 2022.