Similarity Learning with Higher-Order Graph Convolutions for Brain Network Analysis
Guixiang Ma¹, Nesreen K. Ahmed², Ted Willke², Dipanjan Sengupta², Michael W. Cole³, Nicholas B. Turk-Browne⁴, Philip S. Yu¹

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read
Motivation

- learn a similarity between brain networks
- Uses GCN in a Siamese framework
- incorporating higher-order proximity via random walks in graph convolutional networks
- incorporates community structure in brain nets
- 4 real world brain datasets with respect to brain health status and cognitive abilities
Method: Notations

- A multi-subject fMRI data set $\mathcal{G} = \{G_1, G_2, \cdots, G_N\}$
- Where $G_i = (V_i, E_i, A_i)$ is the fMRI brain network of subject $i$,
- $V_i$ is the set of vertices in $G_i$,
- $E_i \subset V_i \times V_i$ is the set of edges in $G_i$,
- $A_i \in \mathbb{R}^{m \times m}$ is the affinity matrix of $G_i$
A whole-brain fMRI image consists of a sequence of 3D brain image scans, where each volume consists of hundreds of thousands of voxels.

To convert the original fMRI images to region-by-region brain networks, extract a sequence of responses from each of the regions of interest (ROI), where each ROI represents a brain region.

compute the region-to-region brain activity correlations.

only keep the positive correlations as the links among the brain regions.

The final constructed network is a graph where the nodes/vertices represent brain regions and the edges are the region-to-region correlations.
Higher-order Graph Convolutional Networks.

- GCNs use spectral filterings: localized to within $K$ neighbor nodes

\[ y = g_\theta \ast x = g_\theta(L)x = g_\theta(U \Lambda U^T)x = U g_\theta(\Lambda) U^T x \]  

(1)

\[ g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\hat{\Lambda}) \]  

(2)

filtering operation can be written as

\[ y = g_\theta(L)x = \sum_{k=0}^{K-1} \theta_k T_k(\hat{L})x, \]

where $T_k(\hat{L}) \in \mathbb{R}^{n \times n}$ is the Chebyshev polynomial of order $k$. The $j^{th}$ output feature map of sample $s$ is then given by

\[ y_{s,j} = \sum_{i=1}^{F_{in}} g_{\theta_{i,j}}(L)x_{s,i} \in \mathbb{R}^m \]  

(3)

build the GCN by stacking multiple convolutional layers with a non-linearity activation (ReLU) following each layer.
The framework

\[ L^{\text{hinge}} = \frac{1}{N_p} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \max(0, 1 - Y_{ij}s_{ij}), \quad (4) \]

where \( N \) is the total number of subjects in the training set, and \( N_p = \frac{N(N - 1)}{2} \) is the total number of pairs from the training set.
### Algorithm 1 Higher-order Siamese GCN

**Input:** $\mathcal{G} = G_1, G_2, \ldots, G_n$ (training graph samples); $\mathbf{y}$ (class labels); random walk parameters: $\gamma$ (number of walks), $l$ (walk length), $w$ (window size)

1. Obtain the mean $k$-nn graph $\bar{G}(V, E, \bar{A})$
2. Initialize a frequency matrix $\mathbf{F} \in \mathbb{R}^{m \times m}$ with 0s
3. **for** $i = 0$ to $\gamma$ **do**
4. \hspace{1em} $V' = \text{Shuffle}(V)$
5. \hspace{1em} **for each** $v_i \in V'$ **do**
6. \hspace{2em} $W_{v_i} = \text{RandomWalk}(\bar{G}, v_i, l)$
7. \hspace{2em} Update $\mathbf{F}$
8. \hspace{1em} **end for**
9. **end for**
10. Obtain a $k$-nn graph $G'$ based on $\mathbf{F}$
11. Merge the edges of $G'$ into $\bar{G}$
12. Obtain the updated adjacency matrix $\mathbf{A}$
13. Prepare pairs of training samples from $\mathcal{G}$
14. Initialize the parameters $\Theta$ of GCNs in Siamese network
15. **while** not converge **do**
16. \hspace{1em} Perform spectral filterings according to Equation (3)
17. \hspace{1em} Compute the similarity estimate $s_{i,j}$ for the input pair $(G_i, G_j)$
18. \hspace{1em} Compute the loss $L^{hinge}$ according to Equation (4)
19. \hspace{1em} Apply stochastic gradient descent with ADAM optimizer to update $\Theta$
20. **end while**
Experiments: Data

- Autism Brain imaging Data Exchange (ABIDE)
- Human Connectome Project (HCP): Does not have class labels of cognitive traits, use three key cognitive features from the participants’ behavioral data to apply K-means clustering with the three features to cluster the subjects into 2 groups
- Bipolar: fMRI data of 52 bipolar I subjects who are in euthymia and 45 healthy controls with matched demographic characteristics
- Human Immunodeficiency Virus Infection (HIV): resting-state fMRI data of 77 subjects, 56 of which are early HIV patients and the other 21 subjects are seronegative controls.
Results

Table: AUC Scores of Pair Classification (mean ± std).

<table>
<thead>
<tr>
<th>Methods</th>
<th>ABIDE</th>
<th>HCP</th>
<th>HIV</th>
<th>Bipolar</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>0.51 ± 0.01</td>
<td>0.52 ± 0.01</td>
<td>0.54 ± 0.07</td>
<td>0.52 ± 0.01</td>
</tr>
<tr>
<td>SE</td>
<td>0.55 ± 0.02</td>
<td>0.54 ± 0.01</td>
<td>0.57 ± 0.02</td>
<td>0.55 ± 0.01</td>
</tr>
<tr>
<td>S-GCN</td>
<td>0.78 ± 0.29</td>
<td>0.81 ± 0.36</td>
<td>0.61 ± 0.25</td>
<td>0.74 ± 0.19</td>
</tr>
<tr>
<td>HS-GCN</td>
<td>0.96 ± 0.02</td>
<td>0.98 ± 0.03</td>
<td>0.77 ± 0.20</td>
<td>0.94 ± 0.07</td>
</tr>
</tbody>
</table>

Figure: AUC Scores of Pair Classification.
Figure: Visualization of the community structure captured by HS-GCN in healthy and bipolar disease networks. Notably this figure highlights the reduced functional connectivity as shown by decreased clustering in the bipolar network.
Figure: Pair classification AUC of S-GCN and HS-GCN with different values for K

(a) ABIDE

(b) Bipolar
Figure: Pair classification AUC of S-GCN and HS-GCN with different numbers of GC layers

(a) ABIDE
(b) Bipolar
Figure: Subject classification accuracy on ABIDE and Bipolar with two different loss functions: We apply the weighted k-nearest neighbour (kNN) $L^{\text{convar}} = \max(0, \delta^{2+} - a) + \max(0, \delta^{2-} - a) + \max(0, m - (\sigma^+ - \sigma^-))$, (5)
Parameter Analysis

(a) ABIDE

(b) Bipolar

(c) HIV

(d) HCP

Figure: AUC scores for Pair classification with different values for Random Walk parameters.

Similarity Learning with Higher-Order Graph Convolution

Presenter: Arshdeep Sekhon

https://qdata.github.io/deep2Read
**Conclusion**

- for learning similarity among fMRI brain networks using higher-order GCNs as the twin
- working well for relatively small datasets
- Graphs structurally different per sample but uses the same modified $A$ for all samples
- limited to *fMRI* or known graph structures