MGNN: Graph Neural Networks Inspired by Distance Geometry Problem

Presentation

Guanyu Cui, Zhewei Wei*

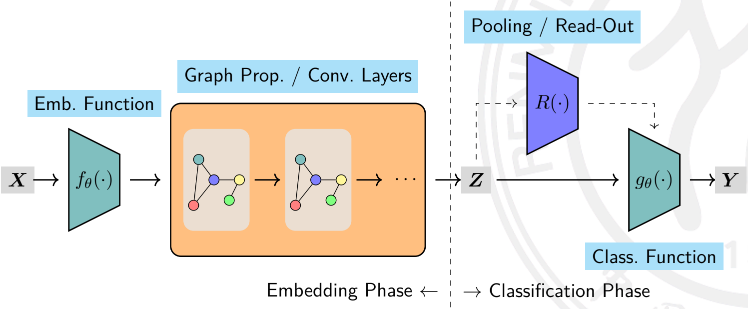
Gaoling School of Artificial Intelligence
Renmin University of China

Graph Neural Networks

- Graph Neural Networks (GNNs) have become a central topic in graph learning;
- They have found diverse applications in
  - physics simulation,
  - traffic forecasting,
  - recommendation systems, and more...

Graph Neural Networks

- A typical GNN architecture consists of some key components.



Expressive Power & Universality of GNNs

- Many studies have explored the expressive power and universality of GNNs.
- Spectral GNNs: Design universal graph filters.
- Spatial GNNs: Explore the connection between expressive power of GNNs and the WL-test.

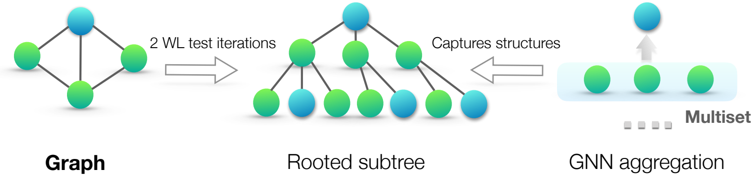


Image taken from Xu et al., How Powerful are Graph Neural Networks? (ICLR 2019)

GNNs and Geometric / Physical Objects

- Growing interest in exploring the connections between GNNs and various geometric and physical objects, such as:
    - graph curvature,
    - oscillators...
- No works have attempted to define the universality from a geometric perspective. — The gap our paper aims to fill.

Equivalent & Congruent

Definition (Equivalent)

Two embedding matrices $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)} \in \mathbb{R}^{n\times d}$ of a graph G are equivalent (denoted as $\boldsymbol{Z}^{(1)} \equiv_E \boldsymbol{Z}^{(2)}$) if $\|\boldsymbol{Z}^{(1)}_{i:} - \boldsymbol{Z}^{(1)}_{j:}\|_2 = \|\boldsymbol{Z}^{(2)}_{i:} - \boldsymbol{Z}^{(2)}_{j:}\|_2$ for all $(i,j) \in E$.

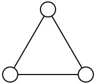
Definition (Congruent)

Two embedding matrices $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)} \in \mathbb{R}^{n\times d}$ of a graph G are congruent (denoted as $\boldsymbol{Z}^{(1)} \cong_{V^2} \boldsymbol{Z}^{(2)}$) if $\|\boldsymbol{Z}^{(1)}_{i:} - \boldsymbol{Z}^{(1)}_{j:}\|_2 = \|\boldsymbol{Z}^{(2)}_{i:} - \boldsymbol{Z}^{(2)}_{j:}\|_2$ for all $i,j \in V$.

Graph Rigidity

Definition (Globally Rigid)

An embedding matrix Z of a graph G is globally rigid if all its equivalent embedding matrices Z' are also congruent to Z.



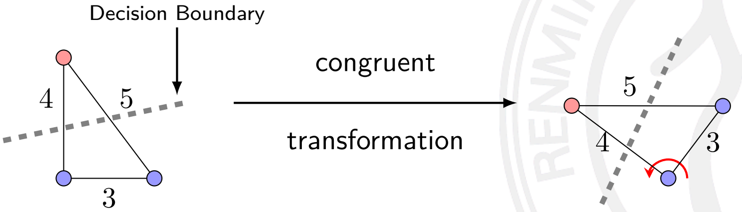
Metric Matrix

Definition (The Metric Matrix of an Embedding Matrix)

The metric matrix of an embedding matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times d}$ is defined as a matrix that contains all pairwise distances between the embedding vectors, i.e., $(\boldsymbol{M_Z})_{ij} = \|\boldsymbol{Z}_{i:} - \boldsymbol{Z}_{j:}\|_2$.

- We also define a mapping from an embedding matrix $\boldsymbol{Z}$ to its metric matrix $\boldsymbol{M_Z}$ as $\boldsymbol{M_Z} = M(\boldsymbol{Z})$.

An Observation

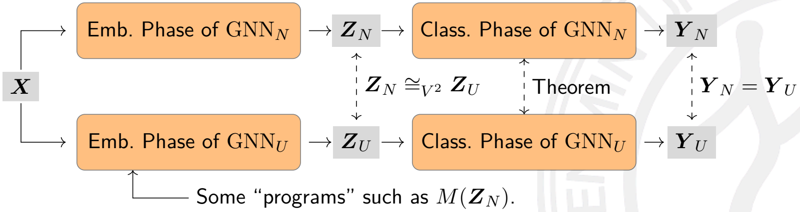


Formalize the Observation

Theorem (MLPs Are Congruent-Insensitive)

Given two congruent embedding matrices $\boldsymbol{Z}_1$ *and* $\boldsymbol{Z}_2$*, for any* MLP_M*, there always exists another* MLP_N *such that they produce identical predictions, i.e.,* $\mathrm{MLP}_M(\boldsymbol{Z}_1) = \mathrm{MLP}_N(\boldsymbol{Z}_2)$.

Spatial-Universality



- **Spatial-Universal**: It can generate an embedding with the given metric matrix!
- The metric matrix serves as a guiding program to arrange the nodes.
- Closely related to the **Distance Geometry Problem (DGP)**.

The Distance Geometry Problem (DGP)

Distance Geometry Problem (DGP)

Given a positive integer d, a graph $G = (V, E)$, and a symmetric non-negative metric matrix $\boldsymbol{M}$, decide whether there exists an embedding matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times d}$, such that

$$\forall (i,j) \in E, \|\boldsymbol{Z}_{i:} - \boldsymbol{Z}_{j:}\| = \boldsymbol{M}_{ij}.$$

Balance Efficiency and Expressive Power

- Full metric matrix: $O(n^2) \Rightarrow$ partial metric matrix: $O(m)$;

  Does this change affect the expressive power?
  Yes.

- For any globally rigid graph, the full “shape” is determined by partial metric matrix;
- For other cases, the “shape” cannot be determined, which weakens the expressive power.
- The challenge is, deciding global rigidity and solving the DGP are both NP-hard (Saxes 1979), making it difficult to effectively find an embedding that satisfy the metric constraint.

Optimization Objective

- To address this, we utilize an optimization objective to approximately arrange the nodes.

$$E_p(\boldsymbol{Z};\boldsymbol{M},E) = \frac{1}{2}\|\boldsymbol{A}\odot(M(\boldsymbol{Z})-\boldsymbol{M})\|_F^2 = \sum_{(i,j)\in E}\frac{1}{2}\left(\|\boldsymbol{Z}_{i:}-\boldsymbol{Z}_{j:}\|_2-\boldsymbol{M}_{ij}\right)^2.$$

- This objective is derived from the raw stress function in the **Multidimensional Scaling (MDS)** problem.

Align with Other GNNs

- To make our optimization objective consistent with other GNNs, we make certain modifications:
  - Re-parameterize $\boldsymbol{Z}$ as $\boldsymbol{D}^{-1/2}\boldsymbol{Z}$ to obtain the normalized Laplacian matrix, aligning it with representative GNNs;
  - Introduce a trade-off regularization term $\|\boldsymbol{Z}-\boldsymbol{Z}^{(0)}\|_F^2$ to align with graph signal de-noising and other optimization derived GNNs;
- Then we get the final form of the objective function:

$$\begin{aligned}\mathcal{L}(\boldsymbol{Z};\boldsymbol{Z}^{(0)},\boldsymbol{M},E) &= (1-\alpha)\tilde{E}_p(\boldsymbol{Z};\boldsymbol{M},E)+\alpha\|\boldsymbol{Z}-\boldsymbol{Z}^{(0)}\|_F^2 \\ &= (1-\alpha)E_p(\boldsymbol{D}^{1/2}\boldsymbol{Z};\boldsymbol{M},E)+\alpha\|\boldsymbol{Z}-\boldsymbol{Z}^{(0)}\|_F^2.\end{aligned}$$

About the Metric Matrix

- In scenarios where we have prior knowledge about the distances between nodes, like,
    - molecular conformation generation, or
    - graph drawing,

  we can directly use that pre-designed metric matrix.
- In other scenarios without a pre-designed metric matrix, we need to learn one from data.

About the Metric Matrix

- General idea: Increase the distances between dissimilar nodes and reduce the distances between similar nodes.
- Introduce edge attention $\alpha_{ij} \in [-1, 1]$:
  - α_{ij} approaches $1 \Leftrightarrow i$, j tend to belong to the same class;
  - α_{ij} approaches $-1 \Leftrightarrow i$, j tend to belong to different classes;

  inspired by research on heterophilic graphs and signed graphs.

About the Metric Matrix

1. Map the initial embedding $\boldsymbol{Z}^{(0)}$ (defined later) to a hidden matrix $\boldsymbol{H}$ via an MLP;
2. Use attention mechanisms, such as
   - concatenation: $\alpha_{ij} = \tanh\left(\boldsymbol{a}^\top [\boldsymbol{H}_{i:}^\top \| \boldsymbol{H}_{j:}^\top]\right)$;
   - bilinear: $\alpha_{ij} = \tanh\left(\boldsymbol{H}_{i:} \boldsymbol{W} \boldsymbol{H}_{j:}^\top\right)$;

   to learn the edge attention;
3. Then we can set $\boldsymbol{M}_{ij} = \frac{1-\alpha_{ij}}{1+\alpha_{ij}+\varepsilon} \|\boldsymbol{Z}_{i:}^{(0)} - \boldsymbol{Z}_{j:}^{(0)}\|$, where ε is a small positive number.

The Embedding Function

- The first part is the embedding function $f_\theta(\boldsymbol{X})$, which maps node features into a d-dimensional latent space to get the initial embedding $\boldsymbol{Z}^{(0)}$.
- Common choices for this function include:
    - Linear layers $f(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{W} + \mathbf{1}\boldsymbol{b}^\top$ in linear GNNs, or
    - Shallow MLPs $f(\boldsymbol{X}) = \sigma(\sigma(\boldsymbol{X}\boldsymbol{W}_1 + \mathbf{1}\boldsymbol{b}_1^\top)\boldsymbol{W}_2 + \mathbf{1}\boldsymbol{b}_2^\top)$ in spectral GNNs.

Propagation

- The second part is the propagation module.
- Goal: Design a graph propagation method that minimizes the objective function.
- Since the objective is typically non-convex, finding its global minimum is challenging.
- Following related works that optimize the raw stress function $E_p(\boldsymbol{Z}; \boldsymbol{M}, E)$, we use stationary point iteration method.

Propagation

- By computing the gradient of $\mathcal{L}$, setting it to 0, and rearranging the terms, we obtain the following equation:

$$\boldsymbol{Z} = (1-\alpha)\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{Z} + (1-\alpha)\boldsymbol{D}^{-1/2}\boldsymbol{L}_{\boldsymbol{H}}\boldsymbol{D}^{-1/2}\boldsymbol{Z} + \alpha\boldsymbol{Z}^{(0)},$$

where $\boldsymbol{H} = \boldsymbol{A} \odot \boldsymbol{M} \odot M(\boldsymbol{D}^{-1/2}\boldsymbol{Z})^{\odot -1}$, and $\boldsymbol{L}_{\boldsymbol{H}} = \mathrm{diag}(\boldsymbol{H}\boldsymbol{1}) - \boldsymbol{H}$;

Propagation

- Rewriting it as an iteration form and substituting $1-\alpha$ with β to allow more flexibility, it leads to the final propagation equation:

$$\boldsymbol{Z}^{(k+1)} = (1-\alpha)\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{Z}^{(k)} + \beta\boldsymbol{D}^{-1/2}\boldsymbol{L_H}\boldsymbol{D}^{-1/2}\boldsymbol{Z}^{(k)} + \alpha\boldsymbol{Z}^{(0)};$$

- We also have the message-passing form of the propagation rule:

$$\boldsymbol{Z}_{i:}^{(k+1)} = (1-\alpha)\sum_{j\in N(i)}\frac{\boldsymbol{Z}_{i:}^{(k)}}{\sqrt{d_i d_j}} + \beta\sum_{j\in\mathcal{N}(i)}\frac{\boldsymbol{M}_{ij}\left(\boldsymbol{Z}_{i:}^{(k)}-\boldsymbol{Z}_{j:}^{(k)}\right)}{\sqrt{d_i d_j}\left\|\frac{\boldsymbol{Z}_{i:}^{(k)}}{\sqrt{d_i}}-\frac{\boldsymbol{Z}_{j:}^{(k)}}{\sqrt{d_j}}\right\|_2} + \alpha\boldsymbol{Z}_{i:}^{(0)}.$$

Optional Linear and Non-Linear Transformations

- The third part is the optional linear and non-linear transformations.
- After each propagation step, we have the flexibility to incorporate them into our model.
- In our experiments without pre-designed metric matrices, such as node classification, we adopt three designs from the GCNII model:
  - a linear transformation,
  - the identity mapping,
  - and a non-linear transformation (ReLU).

The Classification Function

- The last part is the final classification function $g_\theta(\boldsymbol{Z}^{(L)})$, which maps the embeddings to the output dimension.
- We choose a linear layer

$$g(\boldsymbol{Z}^{(L)}) = \boldsymbol{Z}^{(L)}\boldsymbol{W} + \mathbf{1}\boldsymbol{b}^\top$$

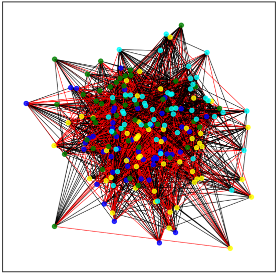
to be the classification function.

Experiments

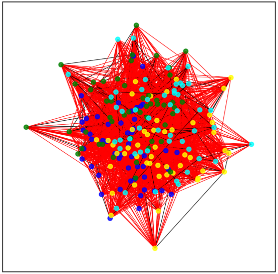
- We conducted “Arranging Nodes with Given Metric Matrices” experiments on synthetic graphs.
- Additionally, we also performed supervised node classification and graph regression experiments using our MGNN model.

Experiments

- In the first experiment, we generate two stochastic block model (SBM) graphs, one homophilic and one heterophilic, with 4 blocks, each containing 50 nodes;
- The nodes features are sampled from two 2-dimensional Gaussian distributions.
- We visualize the graphs in the figures below.



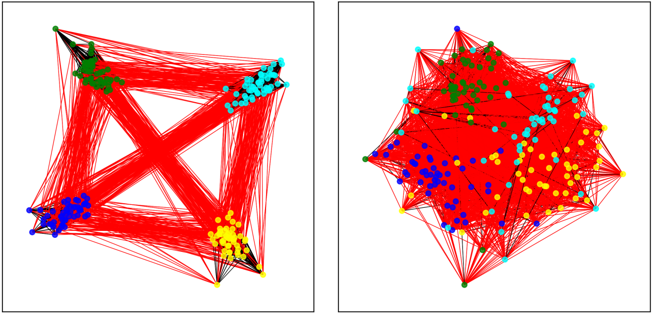
original



original

Experiments

- For the metric matrix, if i and j are in the same class, we set $\boldsymbol{M}_{ij} = 0$; otherwise, we set $\boldsymbol{M}_{ij} = 5$;
- We pass the node features through 8 MGNN propagation layers, with $\alpha = 0.05$, $\beta = 0.5$.
- The results show that our MGNN model separates the blocks.



MGNN-8 MGNN-8

Experiments

- We also conducted supervised node classification and graph regression experiments, and the results are promising.
- For detailed experiment information and results, please refer to our paper.

Summary

- We introduced the concept of spatial-universal GNNs;
- We proposed an optimization objective and designed the MGNN model, to balance efficiency and expressive power;
- We demonstrated the effectiveness of our model through extensive experiments.

Thanks!

Q&A

Contact us: cuiguanyu@ruc.edu.cn, zhewei@ruc.edu.cn