Graph Few-shot Learning via Knowledge Transfer

Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh V. Chawla, Zhenhui Li [1]

Zhiming Xu

zhiming.xu@polyu.edu.hk

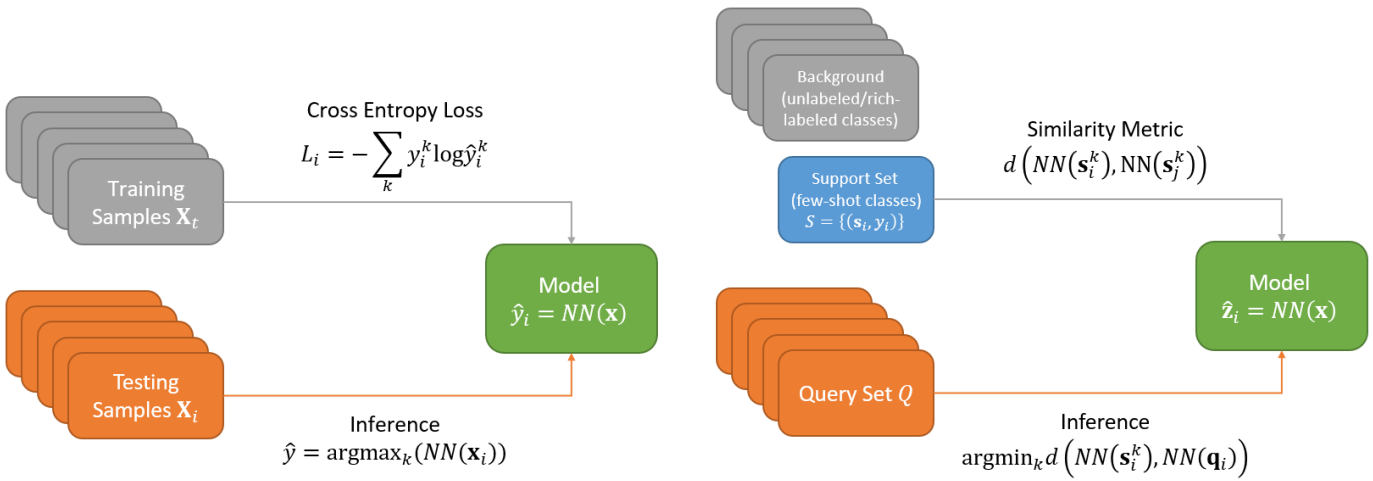
Department of Computing
The Hong Kong Polytechnic University

January 21, 2021

Overview

Few-shot Learning

- One way of meta learning. Contrary to traditional supervised learning practices, only a small number of labeled data (a *few shots*) is available for training.
- Given a support set where there are some samples in each class, and a query set where there are queries of unknown classes, train a model to match each query with the class it belongs to.



(a) Traditional supervised learning. (b) Few-shot learning.

Figure 1: Comparison between supervised and few-shot learning.

Knowledge Transfer

- Store knowledge gained when solving one problem and apply it to another related problem.
- Example: Train a classification model on animals, and the knowledge learned can help classify plants.

Graph Few-shot Learning Problem

Input

- A sequence of graphs $\{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_{N_t}\}$ from a distribution $\mathcal{E}$.
- Support set $\mathcal{S}_i = \{(\mathbf{x}_{i,j}^{s_i}, y_{i,j}^{s_i})\}_{j=1}^{n^{s_i}}$ of n^{s_i} labeled nodes.
- Query set $\mathcal{Q}_i = \{(\mathbf{x}_{i,j}^{q_i}, y_{i,j}^{q_i})\}_{j=1}^{n^{q_i}}$ of n^{q_i} unknown nodes.

Goal

- For each query in $\mathcal{Q}$, predict its corresponding label.

Graph Few-shot Learning Problem (cont'd)

Label Prediction

- With some similarity metric d, associate query node j's embedding with one in support set $\mathcal{S}$, then predict the label according to that of the support node.

$$\text{label}(q_{i,j}) = \arg_{y_{i,j}} \min d\left(f_\theta(\mathbf{A}, x_{i,j}^{q_{i,j}}), (f_\theta(\mathbf{A}, x_{i,j}^{s_i}), y_{i,j}^{s_i})\right) \quad (1)$$

- Alternatively, associate j with the closest geometric center (*prototype*) of nodes' representations in some class k: $\mathbf{c}_i^k = \sum_{x_{i,j}^{s_i} \in S_i^k} f_\theta(\mathbf{A}, x_{i,j}^{s_i}) / \left|\mathcal{S}_i^k\right|$.

Graph Few-shot Learning Problem (cont'd)

Loss

- For each graph $\mathcal{G}_i$, the loss is defined as

$$\mathcal{L}_i^k = - \sum_{\left(\mathbf{x}_{i,j}^{q_i}, y_{i,j}^{q_i} \in \mathcal{Q}_i^k\right)} \log \frac{\exp\left(-d\left(f_\theta(\mathbf{A}, \mathbf{x}_{i,j}^{q_i}), \mathbf{c}_i^k\right)\right)}{\sum_{k'} \exp\left(-d\left(f_\theta(\mathbf{A}, \mathbf{x}_{i,j}^{q_i}), \mathbf{c}_i^{k'}\right)\right)} \tag{2}$$

  $\mathcal{Q}_i^k$ is the support set of class k in graph $\mathcal{G}_i$.

- The loss of all nodes in graph $\mathcal{G}_i$ is the summation over all classes, i.e., $\mathcal{L}_i = \sum_k \mathcal{L}_i^k$.

Objective

- Learn a GNN representation f_θ, such that $\mathcal{L}_i$ is minimized for every observed $\mathcal{G}_i, \mathcal{G}_i \sim \mathcal{E}$.

$$\min_\theta \sum_i \mathcal{L}_i \tag{3}$$

Challenges

- Different nodes in then same graph are not identically and independently distributed as different images.
- Knowledge transfer formulated as parameter initialization and metric space is not sufficient to capture complex, multi-level relations and structures in graphs.

Table of Contents

GFL: Overall Framework

Authors propose Graph Few-shot Learning (GFL) algorithm, aiming to adapt graph-structure knowledge gained from observed graphs to newly discovered ones by exploiting both node and graph level relations and structures.

Specifically, it contains the following components:

- *Prototype GNN* (PGNN): Learn each class k's prototype representation with support set $\mathcal{S}_i^k, \forall k$.
- *Hierarchical Graph Representation Gate*: Learn graph-level representation and ensure knowledge transfer among similar graphs.
- *Auxiliary Graph Reconstruction*: Enhance training stability and representation quality.

They corresponds to Part (a), (b), and (c) in Fig. 2, respectively.

GFL: Illustration

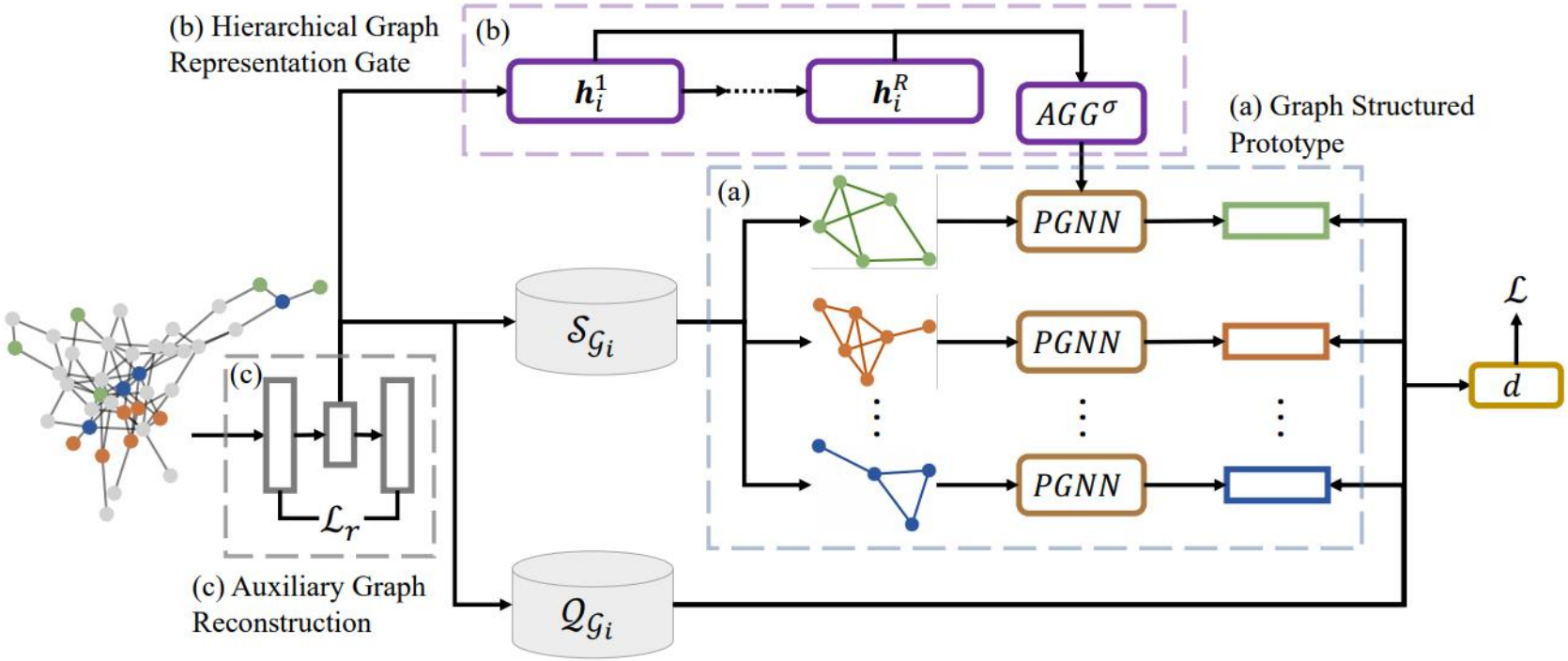


Figure 2: The proposed Graph Few-shot Learning (GFL) model.

Table of Contents

Prototype GNN

Model the relational structure among nodes in support set. The relational structure $\mathcal{R}_i^k$ in $\mathcal{G}_i$ is constructed with some similarity metrics. Hence the support nodes' interactions with such relation is $\mathrm{PGNN}_\phi\left(\mathcal{R}_i^k, f_\theta(\mathcal{S}_i^k)\right)$. The prototype of each class k in $\mathcal{G}_i$ is

$$\mathbf{c}_i^k = \mathrm{Pool}_{j=1}^{n^{s_i^k}}\left(\mathrm{PGNN}_\phi\left(\mathcal{R}_i^k, f_\theta(\mathcal{S}_i^k)\right)[j]\right) \tag{4}$$

PGNN is parameterized by ϕ. Pool denotes some kind of pooling operator, and $n^{s_i^k}$ is the number of class k's nodes in $\mathcal{S}_k^i$.

Table of Contents

Hierarchical Graph Representation Gate

A simple PGNN may fail to capture complex graph structures, authors introduce *hierarchical graph representation gate* to extract graph-specific information and incorporate it with ϕ. For each level, there are two stages:

- *Node Assignment*: Every low-level node is assigned to a high-level community with an assignment GNN (AGNN).
- *Representation Fusion*: The adjacency matrix is transformed by assignment matrix while the feature matrix is calculated by applying assignment matrix on a fusion GNN (FGNN). Hence next-level feature representation are aggregated from the transformed node representations.

Hierarchical Graph Representation Gate: Assignment

- K^r: # of nodes. $\mathbf{A}_i^r, \mathbf{X}_i^r$: adjacency/feature matrix. r: level no.
- Assignment from node k^r in level r to node k^{r+1} in level $r+1$, i.e., $p_i^{k^r \to k^{r+1}}$ is defined as

$$p_i^{k^r \to k^{r+1}} = \frac{\exp\left(\mathrm{AGNN}\left(\mathbf{A}_i^r, \mathbf{X}_i^r\right)\left[k^r, k^{r+1}\right]\right)}{\sum_{\bar{k}_i^r=1}^{K^r+1} \exp\left(\mathrm{AGNN}\left(\mathbf{A}_i^r, \mathbf{X}_i^r\right)\left[k^r, \bar{k}^{r+1}\right]\right)} \tag{5}$$

$\mathrm{AGNN}\left(\mathbf{A}_i^r, \mathbf{X}_i^r\right)\left[k^r, k^{r+1}\right] \in \mathbb{R}^1$ denotes the assignment value from node k^r in level r to node k^{r+1} in level $r+1$. $\mathbf{P}_i^{r \to r+1} \in \mathbb{R}^{K^r \times K^{r+1}}$ is the whole assignment matrix.

Hierarchical Graph Representation Gate: Fusion

After calculating assignment matrix $\mathbf{P}_i^{r\to r+1}$. The transformed adjacency and feature matrix are calculated as follows

$$\begin{aligned}\mathbf{A}_i^{r+1} &= \left(\mathbf{P}_i^{r\to r+1}\right)^\top \mathbf{A}_i^r \mathbf{P}_i^{r\to r+1} \\ \mathbf{X}_i^{r+1} &= \left(\mathbf{P}_i^{r\to r+1}\right)^\top \mathrm{FGNN}\left(\mathbf{A}_i^r, \mathbf{X}_i^r\right)\end{aligned} \tag{6}$$

Hence the feature representation $\mathbf{h}_i^{r+1}$ of level $r+1$ is aggregated by

$$\mathbf{h}_i^{r+1} = \mathrm{Pool}_{k^{r+1}}^{K^{r+1}}\left(\left(\mathbf{P}_i^{r\to r+1}\right)^\top \mathrm{FGNN}\left(\mathbf{A}_i^r, \mathbf{X}_i^r\right)\left[k^{r+1}\right]\right) \tag{7}$$

Fusion (cont'd)

After calculating each level's representation, a set of $\mathbf{h_i}$'s, i.e., $\mathbf{H}_i = \{\mathbf{h}_i^1, \cdots, \mathbf{h}_i^R\}$ encodes $\mathcal{G}_i$'s structure from different levels. To obtain the final graph representation $\mathbf{h}_i$, another aggregator is used

$$\mathrm{AGG_t}(\mathbf{H}_i) = \begin{cases} \frac{1}{R}\sum_r^R \mathbf{h}_i^r, & \mathrm{t = mean} \\ \frac{1}{R}\sum_r^R \beta_i^r \mathbf{h}_i^r = \frac{1}{R}\sum_r^R \frac{\mathbf{q}_i^\top \mathbf{h}_i^r)}{\sum_{r'=1}^R} \mathbf{q_i}^\top \mathbf{h}_i^{r'} \mathbf{h}_i^r, & \mathrm{t = att} \end{cases} \tag{8}$$

The final representation $\mathbf{h}_i$ is expected to be graph-specific. Additionally, a gate function $\mathbf{g}_i = \mathcal{T}(\mathbf{h}_i)$ is introduced to tailor graph structure specific information. Thus the global transferable knowledge is gated by $\mathbf{g}_i$

$$\begin{aligned} \phi_i &= \mathbf{g}_i \odot \phi \\ \mathbf{g}_i &= \mathcal{T}(\mathbf{h}_i) = \sigma(\mathbf{W}_g \mathbf{h}_i + \mathbf{b}_g) \end{aligned} \tag{9}$$

Table of Contents

Auxiliary Graph Reconstruction

Matching loss is usually insufficient for training informative node representation. $f_\theta(\cdot)$ is refined as an autoencoder and an additional reconstruction loss is applied

$$\mathcal{L}_r(\mathbf{A}_i, \mathbf{X}_i) = \left\| \mathbf{A}_i - \mathrm{GNN}_{\mathrm{dec}}(\mathbf{Z}_i)\mathrm{GNN}_{\mathrm{dec}}^{\top}(\mathbf{Z}_i) \right\|_F^2 \tag{10}$$

$\mathbf{Z}_i = \mathrm{GNN}(\mathbf{A}_i, \mathbf{X}_i)$ is the representation of $\mathcal{G}_i$. $\|\cdot\|_F$ is the Frobenius norm.

Combining $\mathcal{L}_r$ with PGNN's few-shot loss, the training objective of proposed GFL is

$$\min_{\Theta} \sum_{i=1}^{N_t} \mathcal{L}_i + \eta \mathcal{L}_r(\mathbf{A}_i, \mathbf{X}_i) \tag{11}$$

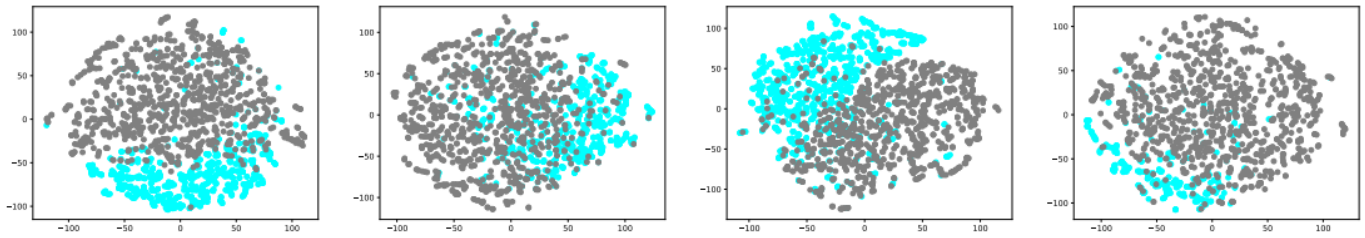
Accuracy on node classification

Model	Collaboration	Reddit	Citation	Pubmed
LP (Zhu and Ghahramani 2002)	$61.09 \pm 1.36\%$	$23.40 \pm 1.63\%$	$67.00 \pm 4.50\%$	$48.55 \pm 6.01\%$
Planetoid (Yang, Cohen, and Salakhudinov 2016)	$62.95 \pm 1.23\%$	$50.97 \pm 3.81\%$	$61.94 \pm 2.14\%$	$51.43 \pm 3.98\%$
Deepwalk (Perozzi, Al-Rfou, and Skiena 2014)	$51.74 \pm 1.59\%$	$34.81 \pm 2.81\%$	$56.56 \pm 5.25\%$	$44.33 \pm 4.88\%$
node2vec (Grover and Leskovec 2016)	$59.77 \pm 1.67\%$	$43.57 \pm 2.23\%$	$54.66 \pm 5.16\%$	$41.89 \pm 4.83\%$
Non-transfer-GCN (Kipf and Welling 2017)	$63.16 \pm 1.47\%$	$46.21 \pm 1.43\%$	$63.95 \pm 5.93\%$	$54.87 \pm 3.60\%$
All-Graph-Finetune (AGF)	$76.09 \pm 0.56\%$	$54.13 \pm 0.57\%$	$88.93 \pm 0.72\%$	$83.06 \pm 0.72\%$
K-NN	$67.53 \pm 1.33\%$	$56.06 \pm 1.36\%$	$78.18 \pm 1.70\%$	$74.33 \pm 0.52\%$
Matchingnet (Vinyals et al. 2016)	$80.87 \pm 0.76\%$	$56.21 \pm 1.87\%$	$94.38 \pm 0.45\%$	$85.65 \pm 0.21\%$
MAML (Finn, Abbeel, and Levine 2017)	$79.37 \pm 0.41\%$	$59.39 \pm 0.28\%$	$95.71 \pm 0.23\%$	$88.44 \pm 0.46\%$
Protonet (Snell, Swersky, and Zemel 2017)	$80.49 \pm 0.55\%$	$60.46 \pm 0.67\%$	$95.12 \pm 0.17\%$	$87.90 \pm 0.54\%$
GFL-mean (Ours)	$83.51 \pm 0.38\%$	$62.66 \pm 0.57\%$	$\mathbf{96.51 \pm 0.31\%}$	$\mathbf{89.37 \pm 0.41\%}$
GFL-att (Ours)	$\mathbf{83.79 \pm 0.39\%}$	$\mathbf{63.14 \pm 0.51\%}$	$95.85 \pm 0.26\%$	$88.96 \pm 0.43\%$

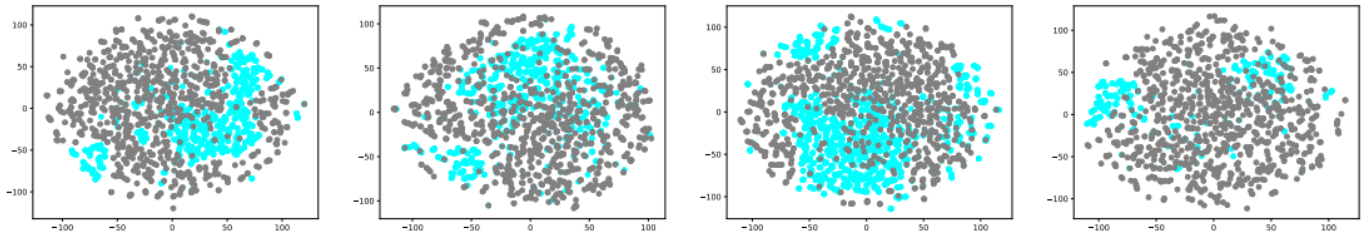
Figure 3: Accuracy on various graph datasets

Ablation Study and Latent Space Analysis

Ablation Model	Collaboration	Reddit	Citation	Pubmed
(M1a): use the mean pooling prototype (i.e., protonet)	$80.49 \pm 0.55\%$	$60.46 \pm 0.67\%$	$95.12 \pm 0.17\%$	$87.90 \pm 0.54\%$
(M2a): remove the hierarchical representation gate	$82.63 \pm 0.45\%$	$61.99 \pm 0.27\%$	$95.33 \pm 0.35\%$	$88.15 \pm 0.55\%$
(M2b): use flat representation rather than hierarchical	$83.45 \pm 0.41\%$	$62.55 \pm 0.65\%$	$95.76 \pm 0.37\%$	$89.08 \pm 0.47\%$
(M3): remove the graph reconstruction loss	$82.98 \pm 0.37\%$	$62.58 \pm 0.47\%$	$95.63 \pm 0.27\%$	$89.11 \pm 0.43\%$
GFL (Ours)	$\mathbf{83.79 \pm 0.39\%}$	$\mathbf{63.14 \pm 0.51\%}$	$\mathbf{96.51 \pm 0.31\%}$	$\mathbf{89.37 \pm 0.41\%}$



(a) : Class 1: GFL (Ours) (b) : Class 2: GFL (Ours) (c) : Class 3: GFL (Ours) (d) : Class 4: GFL (Ours)



(e) : Class 1: Protonet (f) : Class 2: Protonet (g) : Class 3: Protonet (h) : Class 4: Protonet

Figure 4: Upper table: Ablation study of GFL. Lower figures: Node embeddings in latent space

Contributions

- Propose GFL, which adapts metric-based few-shot learning to graph representation learning.
- Integrate node and graph-level knowledge to learn a transferable metric space.
- Improve node classification performances on new graph with knowledge from auxiliary graphs.

Thank You for Your Attention

Q & A

References

[1] H. Yao, C. Zhang, Y. Wei, M. Jiang, S. Wang, J. Huang, N. V. Chawla, and Z. Li, "Graph few-shot learning via knowledge transfer," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 6656–6663. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/6142.