Meta-Inductive Node Classification across Graphs

Zhihao Wen, Yuan Fang, Zemin Liu
School of Computing and Information Systems
Singapore Management University
Outline

- Introduction
- Methodology
- Experiment
- Conclusion & Future Work
Semi-supervised node classification on graphs

(a) Transductive approach

(b) Conventional inductive approach
Two challenges of inductive semi-supervised node classification (Q1 & Q2)

Inductive approach

Aware of

Customize to

Differences across graphs

Two questions

Q1: How do we **dynamically adjust** the inductive model?

Q2: What form of **general knowledge** can empower semi-supervised node classification on a new graph?
Q1: How to dynamically adjust the inductive model?

We propose a **learning-to-train** framework.

Learn a prior that can be adapted to semi-supervised node classification on different graphs (an instance of **meta-learning**)

\[ \theta \]

Graph 1 \((\theta_1)\)

Graph 2 \((\theta_2)\)

Graph 3 \((\theta_3)\)

Prior

Adapt
Q2: What form of general knowledge (i.e. prior)?

- Task-level general knowledge
- Graph-level general knowledge

Condition

Adapt

Final model adapted to semi-supervised node classification tasks on new graphs
Outline

- Introduction & Challenge
- Methodology
- Experiment
- Conclusion & Future Work
MI-GNN: Meta-Inductive GNN

Our meta-inductive approach (MI-GNN)
Overall framework

(a) Training/testing graphs and tasks

(b) Graph-level adaptation

(c) Task-level adaptation

Support loss $L(S_i, \theta_i)$

Dual-adapted model $\theta'_i$

Task (query) loss $L(Q_i, \theta'_i)$

Optimize via backpropagation

Query embeddings $f(v, \theta'_i), \forall v \in Q_j$

Dual-adapted model $\theta'_j$

Support loss $L(S_j, \theta_j)$

Fine-tune

Predict

Sample $(S_i, Q_i)$ on $G_i \in G_{tr}$

Given $(S_j, Q_j)$ on $G_j \in G_{te}$

$G_i$-conditioned task prior $\theta_i$

$G_i$-specific transformation ($\gamma_i$: scaling, $\beta_i$: shifting)

$G_j$-conditioned task prior $\theta_j$

$G_j$-specific transformation ($\gamma_j$: scaling, $\beta_j$: shifting)

General knowledge

Task prior $\theta$

Graph prior $\phi$

Support node with label $c$

Query node with label $c$

(in training)

Query node without label

(in testing)

Support node with label $c$

Query node with label $c$

(in training)

Query node without label

(in testing)
Task prior

Our task prior $\theta$ takes the form of GNNs model, i.e.,

$$\theta = (W^1, W^2, \ldots)$$
Graph-level adaptation

• Graph prior $\phi$.
  • We employ a graph prior $\phi$ to condition the task prior $\theta$

Graph-conditioned task prior

\[
\theta_i = \tau(\theta, g; \phi) = (\gamma_i + 1) \circ \theta + \beta_i
\]
Graph conditioned transformation

- We use MLPs to generate the scaling and shifting vector $\gamma_i$ & $\beta_i$

\[ \gamma_i = MLP_\gamma(g_i; \phi_\gamma) \]
\[ \beta_i = MLP_\beta(g_i; \phi_\beta) \]

Both are vectors

Learnable parameters of the two MLPs

g_i is a graph-level representation of graph i

Graph-conditioned task prior

\[ \theta_i = \tau(\theta, g; \phi) = (\gamma_i + 1) \circ \theta + \beta_i \]

Graph-level representation
\[ \phi = (\phi_\gamma, \phi_\beta) \]
Vector of ones
Element-wise multiplication & addition
Task-level adaptation

\[ \theta'_i = \theta_i - \alpha \frac{\partial L(S_i, Q_i)}{\partial \theta_i} \]

- Task-level adaptation learning rate
- Cross-entropy classification loss of the support set \( S_i \)
Overall training objective

• After the dual adaptations on $G_i$, the goal is to **optimize the general knowledge** and the optimal $(\theta, \phi)$ are given by

$$\arg \min_{\theta, \phi} \sum_{G_i \in G_{tr}} (L(Q_i, \tau'(\theta, g; \phi)) + \lambda(\|\gamma_i\|_2 + \|\beta_i\|_2))$$

Task cross-entropy loss on query nodes $Q_i$

Dual-adapted model ($\theta'_i$)

$L_2$ regularization on scaling & shifting

a hyper-parameter
Outline

- Introduction & Challenge
- Methodology
- Experiment
- Conclusion & Future Work
## Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Flickr</th>
<th>Yelp</th>
<th>Cuneiform</th>
<th>COX2</th>
<th>DHFR</th>
</tr>
</thead>
<tbody>
<tr>
<td># Graphs</td>
<td>800</td>
<td>800</td>
<td>267</td>
<td>467</td>
<td>756</td>
</tr>
<tr>
<td># Edges (avg.)</td>
<td>13.1</td>
<td>43.5</td>
<td>20.1</td>
<td>44.8</td>
<td>44.5</td>
</tr>
<tr>
<td># Nodes (avg.)</td>
<td>12.5</td>
<td>6.9</td>
<td>21.3</td>
<td>41.2</td>
<td>42.4</td>
</tr>
<tr>
<td># Node features</td>
<td>500</td>
<td>300</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td># Node classes</td>
<td>7</td>
<td>10</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Multi-label?</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

(All 5 datasets are from PyTorch Geometric Datasets: https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html)
## Performance comparison with baselines

We report the average accuracy and micro-F1 with 95% confidence interval, in percent. In each column, the best result is bolded and the runner-up is underlined. Improvement by MI-GNN is calculated relative to the best baseline. **/***/* denotes the difference between MI-GNN and the best baseline is statistically significant at the 0.01/0.05/0.1 level under the two-tail t-test.

<table>
<thead>
<tr>
<th></th>
<th>Flickr</th>
<th>Yelp</th>
<th>Cuneiform</th>
<th>COX2</th>
<th>DHFR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Micro-F1</td>
<td>Accuracy</td>
<td>Micro-F1</td>
<td>Accuracy</td>
</tr>
<tr>
<td>DeepWalk</td>
<td>39.88±2.42</td>
<td>30.01±1.21</td>
<td>63.27±2.73</td>
<td>57.11±6.29</td>
<td>74.61±0.60</td>
</tr>
<tr>
<td>Transduct-GNN</td>
<td>13.61±1.22</td>
<td>10.71±1.20</td>
<td>24.87±15.4</td>
<td>23.85±14.6</td>
<td>49.63±0.95</td>
</tr>
<tr>
<td>Induct-GNN</td>
<td>40.48±1.69</td>
<td>29.67±1.77</td>
<td>65.95±0.56</td>
<td>56.61±1.81</td>
<td>74.89±0.35</td>
</tr>
<tr>
<td>K-NN</td>
<td>34.11±1.76</td>
<td>26.39±1.39</td>
<td>61.70±0.90</td>
<td>57.35±1.42</td>
<td>70.36±0.27</td>
</tr>
<tr>
<td>AGF</td>
<td>40.58±1.61</td>
<td>28.99±2.09</td>
<td>65.96±0.54</td>
<td>56.64±1.83</td>
<td>74.89±0.37</td>
</tr>
<tr>
<td>GFL</td>
<td>30.24±0.68</td>
<td>29.51±0.69</td>
<td>61.62±0.97</td>
<td>58.88±2.03</td>
<td>63.72±0.37</td>
</tr>
<tr>
<td>Meta-GNN</td>
<td>39.66±0.92</td>
<td>30.02±2.49</td>
<td>66.24±0.84</td>
<td>56.20±1.81</td>
<td>75.12±0.33</td>
</tr>
<tr>
<td>MI-GNN (improv.)</td>
<td>44.45±2.18</td>
<td>33.79±1.87</td>
<td>67.92±0.69</td>
<td>60.20±2.23</td>
<td>81.48±0.47</td>
</tr>
<tr>
<td>(improv.)</td>
<td>**</td>
<td>**</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
</tbody>
</table>

**/***/* denotes the difference between MI-GNN and the best baseline is statistically significant at the 0.01/0.05/0.1 level under the two-tail t-test.
Alternative GNN architectures

Performance comparison of L2T-GNN and a few key baselines, whilst using GCN and GraphSAGE as the GNN architecture. We report the average accuracy with 95% confidence interval.

<table>
<thead>
<tr>
<th></th>
<th>GCN as the GNN Architecture</th>
<th>GraphSAGE as the GNN Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flickr</td>
<td>Yelp</td>
</tr>
<tr>
<td>Transduct-GNN</td>
<td>14.89±0.94</td>
<td>50.92±0.95</td>
</tr>
<tr>
<td>Induct-GNN</td>
<td>12.08±3.98</td>
<td>55.04±1.77</td>
</tr>
<tr>
<td>AGF</td>
<td>11.94±2.45</td>
<td>53.66±3.04</td>
</tr>
<tr>
<td>Meta-GNN</td>
<td>22.51±3.05</td>
<td>54.80±1.86</td>
</tr>
<tr>
<td>MI-GNN</td>
<td>29.91±6.85</td>
<td>57.22±1.79</td>
</tr>
</tbody>
</table>
Ablation Study

Ablated versions of **MI-GNN**:  
- **Fine-tune only**  
  Neither graph- nor task-level adaptations, but a simple fine-tuning step.
- **Graph-level only**  
  Remove the task-level adaptation from MI-GNN.
- **Task-level only**  
  Remove graph-level adaptation from MI-GNN.
Performance case study

• Transductive methods are **not influenced** by similarity between testing and training graphs.

• Performance of inductive models **correlates** to such similarity.

• Our meta-inductive approach is **robust** due to the **dual-adaptation**.
Outline

Introduction & Challenge
Methodology
Experiment

Conclusion & Future Work
Conclusion & Future Work

• Conclusion:
  • Studied the problem of inductive semi-supervised node classification across graphs.
  • Proposed a novel framework called MI-GNN, containing a dual adaptation mechanism at both the graph and task levels.
  • Conduct extensive experiments on five real-world graph collections.

• Future work:
  • Consider the node level adaptation, additionally
  • Apply to heterogeneous network
THANK YOU FOR YOUR ATTENTION

Paper, code, data...

www.yfang.site