UD-GNN: Uncertainty-aware Debiased Training on Semi-Homophilous Graphs

Yang Liu¹; Xiang Ao¹*; Fuli Feng²; Qing He¹*

柳 阳¹; 敖 翔¹*; 冯福利²; 何 清¹*

¹ Institute of Computing Technology, Chinese Academy of Sciences
² University of Science and Technology of China

* denotes corresponding author.
➢ Background and Motivation

➢ Method – UD-GNN

➢ Experiment

➢ Conclusion and Future Work
Background

Homophily: The tendency of individuals to associate and bond with similar others

- People with the same interest are more closely connected
- Researchers who focus on the same research area are more likely to establish a connection

Images credit to Easley and Kleinberg, Networks, crowds, and markets. 2010
**Heterophily:** Nodes from different classes tend to connect to each other

- Fraudsters connect to benign users to camouflage themselves
- Interdisciplinary papers cite papers from other research areas
Related Work

GNNs for Homophilous Graphs


GNNs for Heterophilous Graphs

Preliminaries

Homophilous Graph

Semi-Homophilous Graph

Heterophilous Graph

Homophily Ratio

Definitions:

\[ \mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X}, \mathbf{Y}) \]

\[ \mathcal{E}_{\text{intra}} = \{ (u, v) | \mathbf{A}_{uv} = 1 \land y_u = y_v \} \]

\[ \mathcal{E}_v = \{ (u, v) | \mathbf{A}_{uv} = 1 \} \]

➢ Graph-level Homophily Ratio:

\[ h_{\mathcal{G}} = \frac{|\mathcal{E}_{\text{intra}}|}{|\mathcal{A}|} = \frac{3}{7} \]

➢ Node-level Homophily Ratio:

\[ h_v = \frac{|\mathcal{E}_v \cap \mathcal{E}_{\text{intra}}|}{|\mathcal{E}_v|} \]
Preliminaries -- Categorization of Graph Benchmarks

- **Heterophilous Graph**
- **Semi-Homophilous Graph**
- **Homophilous Graph**

### Node-level Homophily Ratio

$$h_v = \frac{|\mathcal{E}_v \cap \mathcal{E}_{\text{intra}}|}{|\mathcal{E}_v|}$$

**Strong Heterophily**

- Ratio: 0
- Represented by orange

**Weak Heterophily**

- Ratio: 0.25
- Represented by orange

**Weak Homophily**

- Ratio: 0.50
- Represented by orange

**Strong Homophily**

- Ratio: 0.75
- Represented by orange

**Strong Heterophily**

- Ratio: 1
- Represented by orange

Motivation

<table>
<thead>
<tr>
<th>Strong Heterophily</th>
<th>Weak Heterophily</th>
<th>Weak Homophily</th>
<th>Strong Homophily</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.25</td>
<td>0.50</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Node-level Homophily Ratio

\[ h_v = \frac{|\mathcal{E}_v \cap \mathcal{E}_{\text{intra}}|}{|\mathcal{E}_v|} \]

➢ For strong homophilous nodes, the accuracy is close to 0.99.

➢ For strong heterophilous nodes, the accuracy ranges from 0.48 to 0.74.

➢ For the four GNNs, the performance gap exists with a range from 0.25 to 0.5.

Group-wise Performance

<table>
<thead>
<tr>
<th>Acc</th>
<th>GCN</th>
<th>GAT</th>
<th>H2GCN</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Semi-Homophilous Graph

- For strong homophilous nodes, the accuracy is close to 0.99.
- For strong heterophilous nodes, the accuracy ranges from 0.48 to 0.74.
- For the four GNNs, the performance gap exists with a range from 0.25 to 0.5.
➢ Background and Motivation

➢ Method – UD-GNN
  • Observation and Overview
  • Uncertainty Estimation
  • Debiased Training

➢ Experiment

➢ Conclusion and Future Work
Observation

For transductive node classification

- The node-level homophily ratio computed from incomplete labels are unreliable
  - Some node labels are unavailable during the training phase

- The output of GNNs exhibit high uncertainty for heterophilous nodes
  - The output uncertainty may help to identify heterophilous nodes
UD-GNN: Uncertainty-aware Debiasd Graph Neural Network

**Method - Overview**

1. **Uncertainty Estimation**
   - \( W_d = (1 - Z) \odot W_b + \text{stop
dradient}(Z \odot W_b) \)
   - \( \mathbb{E} \left[ \hat{Y} \mid A, X \right] = \int \hat{Y} \ p(\hat{Y} \mid A, X) d\hat{Y} \approx \frac{1}{T} \sum_{t=1}^{T} \hat{Y}_t \)
   - \( U(\hat{Y} \mid A, X) = \text{Var}(\hat{Y} \mid A, X) \approx \frac{1}{T} \sum_{t=1}^{T} \left( \hat{Y}_t - \mathbb{E}[\hat{Y} \mid A, X] \right)^2 \)

2. **Mask parameters for \( \mathcal{V}_{\text{con}} \)**

3. **Debiased Training**
   - (a) Retrain
   - (b) Prune
   - (c) Train
   - (d) Frozen
   - (e) Masked with zeros
   - Weights to be trained
   - Well-trained weights
   - Frozen weights

- Two node classes
- Unknown node class
- Homophilous nodes
- Heterophilous nodes
Method

➢ Uncertainty Estimation

- Monte Carlo dropout variational inference
- Estimated from \( \{ \hat{W}_t \}_t^{T} \) GNN predictors

\[
\mathcal{L}(W_b) = -\frac{1}{T} \sum_{t=1}^{T} Y \log(f_{\hat{W}_t}(A, X)) + \frac{1 - \theta}{2T} \| W_b \|^2
\]

\[
\hat{Y}_t = f_{\hat{W}_t}(A, X)
\]

\[
\mathbb{E}[\hat{Y}|A, X] = \int \hat{Y} p(\hat{Y}|A, X) d\hat{Y} \approx \frac{1}{T} \sum_{t=1}^{T} \hat{Y}_t
\]

\[
U[\hat{Y}|A, X] = \text{Var}(\hat{Y}|A, X) \approx \frac{1}{T} \sum_{t=1}^{T} (\hat{Y}_t - \mathbb{E}[\hat{Y}|A, X])^2
\]
**Debiased Training**

- Divide $\mathcal{V}_{train}$ into $\mathcal{V}_{con}$ and $\mathcal{V}_{unc}$ according to $U$ such that debiasing ratio $\gamma = \frac{|\mathcal{V}_{con}|}{|\mathcal{V}_{unc}|}$
- Prune the parameters close to zero in $W_b$ with 0-1 mask $Z$ and retrain the remained parameters with $\mathcal{V}_{con}$
- Freeze $Z \odot W_b$ and train $(1 - Z) \odot W_b$ with $\mathcal{V}_{unc}$ to obtain $W_d$

$$W_d = (1 - Z) \odot W_b + \text{stop\_gradient}(Z \odot W_b)$$

$$L(W_d) = -\frac{1}{|\mathcal{V}_{unc}|} \sum_{v \in \mathcal{V}_{unc}} y_v \log(f_{W_d}(A, X_{v \cup N_v}))$$
➢ Background and Motivation

➢ Method – UD-GNN

➢ Experiment
  • RQ1: Does UD-GNN outperform the state-of-the-art methods on semi-homophilous graphs?
  • RQ2: How do the key components contribute to the results?
  • RQ3: Does UD-GNN work well on heterophilous graphs?
  • RQ4: What is the sensitivity of UD-GNN with respect to different debiasing ratios, mixing ratios and number of classes?

➢ Conclusion and Future Work
Data

➢ Public benchmark
  • **cSBM**: contextual stochastic block models
  • **Penn94**: a friendship network from the Facebook100 networks
  • **Cora-full**: a citation network labeled on the paper topic
  • **Ogbn-arxiv**: the citation network between all Computer Science (CS) arXiv papers indexed by MAG

➢ Train/Valid/Test:
  • **cSBM**: 40%/20%/40%
  • **Penn94**: 80%/10%/10%
  • **Cora-full**: 70%/10%/20%
  • **Ogbn-arxiv**: 2017/2018/2019

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Node</th>
<th>#Edge</th>
<th>#Class</th>
<th>#Feat</th>
<th>%Heter</th>
</tr>
</thead>
<tbody>
<tr>
<td>cSBM</td>
<td>20,000</td>
<td>998,766</td>
<td>10</td>
<td>1,024</td>
<td>50%</td>
</tr>
<tr>
<td>Penn94</td>
<td>41,554</td>
<td>1,362,229</td>
<td>2</td>
<td>4,814</td>
<td>49%</td>
</tr>
<tr>
<td>Cora-full</td>
<td>19,793</td>
<td>126,842</td>
<td>70</td>
<td>8,710</td>
<td>44%</td>
</tr>
<tr>
<td>Ogbn-arxiv</td>
<td>169,343</td>
<td>1,166,243</td>
<td>40</td>
<td>128</td>
<td>37%</td>
</tr>
</tbody>
</table>
Experiment

➢ Compared methods

• **GCN, GAT**: Traditional graph convolutional network and graph attention network
• **Mixhop**: Repeatedly mixing feature representations of neighbors at various distances
• **GPR-GNN**: Generalized PageRank GNN
• **JK-Net**: Jump Knowledge
• **H2GCN, CPGNN**: GNNs for heterophilous graphs
• **WRGAT**: Improving the assortativity of graphs with local mixing patterns
• **U-GNN**: Universal GCN extracting information from 1-hop, 2-hop and kNN networks

➢ Metrics

• **Accuracy**: $\text{eval}(Y, f_W(A, X))$
• **Relative bias**: $\delta = \text{eval}(Y, f_{W^*}(A, X)) - \text{eval}(Y, f_W(A, X))$
RQ1: Does UD-GNN outperform the state-of-the-art methods on semi-homophilous graphs?

- UD-GNN achieves the best accuracy with the lowest relative bias
RQ2: How do the key components contribute to the results?

- D-GNN removes uncertainty estimation and trains a separate model to discriminate homophilous nodes from heterophilous nodes for debiasing.
- U-GNN removes debiased training and applies Focal loss based on the estimated uncertainty scores.
RQ3: Does UD-GNN work well on heterophilous graphs?

- UD-GNN improves the performance due to the refining of uncertain nodes in the debiased training.
- UD-GNN achieves the best results on Chameleon and Squirrel, with comparable performance on Wisconsin.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Node</th>
<th>#Edge</th>
<th>#Class</th>
<th>#Feat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chameleon</td>
<td>2,277</td>
<td>36,101</td>
<td>5</td>
<td>2,325</td>
</tr>
<tr>
<td>Squirrel</td>
<td>5,201</td>
<td>217,073</td>
<td>5</td>
<td>2,089</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>251</td>
<td>515</td>
<td>5</td>
<td>1,703</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mixhop</th>
<th></th>
<th></th>
<th>GPRGNN</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VA</td>
<td>UD</td>
<td></td>
<td>VA</td>
<td>UD</td>
<td></td>
</tr>
<tr>
<td>Chameleon</td>
<td>58.25±1.83</td>
<td>59.23±1.24</td>
<td>42.86±1.48</td>
<td>64.36±0.87</td>
<td>66.23±1.03</td>
<td>73.83±6.82</td>
</tr>
<tr>
<td>Squirrel</td>
<td>42.86±1.48</td>
<td>43.92±1.42</td>
<td></td>
<td>46.83±0.84</td>
<td>47.92±0.25</td>
<td></td>
</tr>
<tr>
<td>Wisconsin</td>
<td></td>
<td>74.23±5.92</td>
<td></td>
<td>79.23±3.81</td>
<td>81.29±2.34</td>
<td></td>
</tr>
</tbody>
</table>
RQ4: What is the sensitivity of UD-GNN with respect to different debiasing ratios, mixing ratios and number of classes?
➢ Background and Motivation

➢ Method – UD-GNN

➢ Experiment

➢ Conclusion and Future Work
Conclusion

- We investigate the bias issue between homophily and heterophily on semi-homophilous graphs.
- We propose an Uncertainty-aware Debiasing framework to mitigate the bias.
- Experiments on four benchmark semi-homophilous graph datasets demonstrate the effectiveness of the proposed framework.

Future Work

- New message passing architecture for semi-homophilous graphs
- Spectral filter for semi-homophilous graphs
Thanks for listening!

If you have any question, feel free to contact us at
liuyang520ict@gmail.com
aoxiang@ict.ac.cn

Paper and slides are available at
https://ponderly.github.io/