Graph Structure Learning for Robust Graph Neural Networks

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, Jiliang Tang
KDD 2020
Adversarial Attacks on GNN

![Graph before and after adversarial attack]

GNN

8 Predicted as: <green>

GNN

8 Predicted as: <blue>
Consequences

- Financial Systems
  - Credit Card Fraud Detection
- Recommender Systems
  - Social Recommendation
  - Product Recommendation
- ...
Pro-GNN: Defend Against Adversarial Attacks

Attack Setting
• Untargeted structure attack
• Poisoning attack
• Node classification
  ▪ Graph dataset $G = (A, X)$
  ▪ Graph neural network $f: f(x_i) \rightarrow \hat{y}_i$

Defense Goal
• Improve the overall performance of GNN on the perturbed graph
Pro-GNN: Defend Against Adversarial Attacks

Graph Properties

• Low-rank
• Sparsity
• Feature smoothness

Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.
Pro-GNN: Defend Against Adversarial Attacks

Graph Properties

• Low-rank
• Sparsity
• Feature smoothness

(b) Rank Growth
Pro-GNN: Defend Against Adversarial Attacks

Graph Properties
• Low-rank
• Sparsity
• Feature smoothness

Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.
Pro-GNN: Defend Against Adversarial Attacks

Graph Properties

• Low-rank
• Sparsity
• Feature smoothness

<table>
<thead>
<tr>
<th>Dataset</th>
<th>r(%)</th>
<th>edge+</th>
<th>edge-</th>
<th>edges</th>
<th>ranks</th>
<th>clustering coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5069</td>
<td>2192</td>
<td>0.2376</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>226</td>
<td>27</td>
<td>5268</td>
<td>2263</td>
<td>0.2228</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>408</td>
<td>98</td>
<td>5380</td>
<td>2278</td>
<td>0.2132</td>
</tr>
<tr>
<td>Cora</td>
<td>15</td>
<td>604</td>
<td>156</td>
<td>5518</td>
<td>2300</td>
<td>0.2071</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>788</td>
<td>245</td>
<td>5633</td>
<td>2305</td>
<td>0.1983</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>981</td>
<td>287</td>
<td>5763</td>
<td>2321</td>
<td>0.1943</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3668</td>
<td>1778</td>
<td>0.1711</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>181</td>
<td>2</td>
<td>3847</td>
<td>1850</td>
<td>0.1616</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>341</td>
<td>25</td>
<td>3985</td>
<td>1874</td>
<td>0.1565</td>
</tr>
<tr>
<td>Citeseer</td>
<td>15</td>
<td>485</td>
<td>65</td>
<td>4089</td>
<td>1890</td>
<td>0.1523</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>614</td>
<td>119</td>
<td>4164</td>
<td>1902</td>
<td>0.1483</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>743</td>
<td>174</td>
<td>4236</td>
<td>1888</td>
<td>0.1467</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16714</td>
<td>1060</td>
<td>0.3203</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>732</td>
<td>103</td>
<td>17343</td>
<td>1133</td>
<td>0.2719</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1347</td>
<td>324</td>
<td>17737</td>
<td>1170</td>
<td>0.2825</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1915</td>
<td>592</td>
<td>18038</td>
<td>1193</td>
<td>0.2851</td>
</tr>
<tr>
<td>Polblogs</td>
<td>20</td>
<td>2304</td>
<td>1038</td>
<td>17980</td>
<td>1193</td>
<td>0.2877</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>2500</td>
<td>1678</td>
<td>17536</td>
<td>1197</td>
<td>0.2723</td>
</tr>
</tbody>
</table>

Table Credit: Adversarial Attacks and Defenses on Graphs: A Review and Empirical Study

Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.
Pro-GNN: Defend Against Adversarial Attacks

Graph Properties
• Low-rank
• Sparsity
• Feature smoothness
Pro-GNN: Framework

Figure 2: Overall framework of Pro-GNN. Dash lines indicate smaller weights.
Pro-GNN: Modelling

• Low rank and sparsity

\[
\arg\min_{S \in S} \mathcal{L}_0 = \|A - S\|_F^2 + \alpha \|S\|_1 + \beta \|S\|_*, \text{ s.t., } S = S^T
\]

\[
||S||_1 = \Sigma_{ij} |S_{ij}| \quad ||S||_* = \Sigma_{i=1}^{rank(S)} \sigma_i
\]
Pro-GNN: Modelling

• Feature smoothness

\[ \mathcal{L}_s = \text{tr}(X^T \hat{L} X) = \frac{1}{2} \sum_{i,j=1}^{N} S_{ij} (x_i - x_j)^2 \]
Pro-GNN: Modelling

• Overall objective

\[
\arg \min_{S \in S, \theta} \mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_s + \gamma \mathcal{L}_{GNN}
\]
Pro-GNN: Optimization

• Overall objective

\[
\arg \min_{S \in S, \theta} \mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_S + \gamma \mathcal{L}_{GNN}
\]

\[
= \|A - S\|_F^2 + \alpha \|S\|_1 + \beta \|S\|_\infty + \gamma \mathcal{L}_{GNN}(\theta, S, X, y_L) + \lambda \text{tr}(X^T \hat{L}X)
\]

s.t. \quad S = S^T,

Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.
Pro-GNN: Optimization

• Overall objective

$$\arg\min_{S \in S, \theta} \mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_S + \gamma \mathcal{L}_{GNN}$$

$$= \|A - S\|_F^2 + \alpha \|S\|_1 + \beta \|S\|_* + \gamma \mathcal{L}_{GNN}(\theta, S, X, Y_L) + \lambda tr(X^T \hat{L} X)$$

s.t. \hspace{1cm} S = S^T,$$
Pro-GNN: Optimization

• Alternating Optimization

**Update \( \theta \):**
\[
\min_{\theta} \mathcal{L}_{GNN}(\theta, S, X, Y_L) = \sum_{u \in \mathcal{V}_L} \ell(f_{\theta}(X, S)_u, y_u)
\]

**Update \( S \):**
\[
\min_{S} \mathcal{L}(S, A) + \alpha \|S\|_1 + \beta \|S\|_* \quad \text{s.t.,} \quad S = S^T, S \in S,
\]

where \( \mathcal{L}(S, A) = \|A - S\|_F^2 + \mathcal{L}_{GNN}(\theta, S, X, Y) + \lambda tr(X^T \hat{L} X) \).
Pro-GNN: Optimization

• Incremental Proximal Descent method

\[
\min_S \mathcal{L}(S, A) + \alpha \|S\|_1 + \beta \|S\|_*
\]

For each iteration, do

\[
\begin{cases}
S^{(k)} = S^{(k-1)} - \eta \cdot \nabla_S \mathcal{L}(S, A), \\
S^{(k)} = \text{prox}_{\eta \beta \|\cdot\|_*} \left( S^{(k)} \right), \\
S^{(k)} = \text{prox}_{\eta \alpha \|\cdot\|_1} \left( S^{(k)} \right).
\end{cases}
\]
Pro-GNN: Algorithm

**Algorithm 1:** Pro-GNN

**Data:** Adjacency matrix $A$, Attribute matrix $X$, Labels $Y_L$,
Hyper-parameters $\alpha, \beta, \gamma, \lambda, \tau$, Learning rate $\eta, \eta'$

**Result:** Learned adjacency $S$, GNN parameters $\theta$

1. Initialize $S \leftarrow A$
2. Randomly initialize $\theta$
3. **while** Stopping condition is not met **do**
   4. $S \leftarrow S - \eta \nabla S(||S - A||^2_F + \gamma \mathcal{L}_{GNN} + \lambda \mathcal{L}_s)$
   5. $S \leftarrow \text{prox}_{\eta \beta ||.||_2} (S)$
   6. $S \leftarrow \text{prox}_{\eta \alpha ||.||_1} (S)$
   7. $S \leftarrow P_S(S)$
   8. **for** $i=1$ to $\tau$ **do**
      9. $g \leftarrow \frac{\partial \mathcal{L}_{GNN}(\theta, S, X, Y_L)}{\partial \theta}$
      10. $\theta \leftarrow \theta - \eta' g$
3. **Return** $S, \theta$

Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.
Table 2: Node classification performance (Accuracy±Std) under non-targeted attack (*metattack*).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PtB Rate (%)</th>
<th>GCN</th>
<th>GAT</th>
<th>RGCN</th>
<th>GCN-Jaccard²</th>
<th>GCN-SVD</th>
<th>Pro-GNN-fs</th>
<th>Pro-GNN³</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cora</td>
<td>0</td>
<td>83.50±0.44</td>
<td><strong>83.97±0.65</strong></td>
<td>83.09±0.44</td>
<td>82.05±0.51</td>
<td>80.63±0.45</td>
<td>83.42±0.52</td>
<td>82.98±0.23</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>76.55±0.79</td>
<td>80.44±0.74</td>
<td>77.42±0.39</td>
<td>79.13±0.59</td>
<td>78.39±0.54</td>
<td><strong>82.78±0.39</strong></td>
<td>82.27±0.45</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>70.39±1.28</td>
<td>75.61±0.59</td>
<td>72.22±0.38</td>
<td>75.16±0.76</td>
<td>71.47±0.83</td>
<td>77.91±0.86</td>
<td><strong>79.03±0.59</strong></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>65.10±0.71</td>
<td>69.78±1.28</td>
<td>66.82±0.39</td>
<td>71.03±0.64</td>
<td>66.69±1.18</td>
<td>76.01±1.12</td>
<td><strong>76.40±1.27</strong></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>59.56±2.72</td>
<td>59.94±0.92</td>
<td>59.27±0.37</td>
<td>65.71±0.89</td>
<td>58.94±1.13</td>
<td>68.78±5.84</td>
<td><strong>73.32±1.56</strong></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>47.53±1.96</td>
<td>54.78±0.74</td>
<td>50.51±0.78</td>
<td>60.82±1.08</td>
<td>52.06±1.19</td>
<td>56.54±2.58</td>
<td><strong>69.72±1.69</strong></td>
</tr>
</tbody>
</table>
Pro-GNN: Experiments

Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.
Pro-GNN: Importance of Graph Structure Learning

Table 3: Node classification accuracy given the graph under 25% perturbation by metattack.

<table>
<thead>
<tr>
<th></th>
<th>GCN</th>
<th>GCN-NoGraph</th>
<th>Pro-GNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cora</td>
<td>47.53±1.96</td>
<td>62.12±1.55</td>
<td>69.72±1.69</td>
</tr>
<tr>
<td>Citeseer</td>
<td>56.94±2.09</td>
<td>63.75±3.23</td>
<td>68.95±2.78</td>
</tr>
<tr>
<td>Polblogs</td>
<td>49.23±1.36</td>
<td>51.79±0.62</td>
<td>63.18±4.40</td>
</tr>
<tr>
<td>Pubmed</td>
<td>75.50±0.17</td>
<td>84.14±0.11</td>
<td>86.86±0.19</td>
</tr>
</tbody>
</table>
Pro-GNN: Importance of Graph Structure Learning

Figure 5: Weight density distributions of normal and adversarial edges on the learned graph.
Pro-GNN: Ablation Study

(a) Cora

Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.
Conclusion

• We found that graph adversarial attack can break important graph properties
• We introduced a novel defense approach Pro-GNN that learns the graph structure and GNN parameters simultaneously
• Our experiments show that our model consistently improves the overall robustness under various adversarial attacks.

Paper Link:
Code:
https://github.com/ChandlerBang/Pro-GNN
THANK YOU