An Efficient Adversarial Attack on Graph Structured Data

Zhengyi Wang, Hang Su

Tsinghua University

Jan. 7th, 2021
1. Background

2. Formulation

3. Challenge

4. Cluster Attack
Graph Adversarial Attack

- Graph Adversarial Attack = Graph Neural Network + Adversarial Attack.
- Add *fake nodes* and extra connections.
- Lower the performance of node classification.
Problem Formulation

• Given \( G = (A, X) \), where \( A \in \{0, 1\}^{N \times N} \) and \( X \in \{0, 1\}^{N \times D} \)

• Target node set \( V_{\text{target}} \subseteq V \).

• Add \( N_{\text{fake}} \) fake nodes, leading to \( G^+ = (A^+, X^+) \). where
  \[
  A^+ = \begin{bmatrix} A & B^T \\ B & A_{\text{fake}} \end{bmatrix} \quad \text{and} \quad X^+ = \begin{bmatrix} X \\ X_{\text{fake}} \end{bmatrix}.
  \] (Mild perturbation.)

• Minimize the adversarial loss
  \[
  \min_{A_{\text{fake}}, B, X_{\text{fake}}} \mathbb{L}(G^+, V_{\text{target}}) \quad \text{s.t.} \quad \|B\|_0 \leq \Delta_{\text{edge}}
  \]
Challenge

\[
\min_{A_{\text{fake}}, B, X_{\text{fake}}} \mathbb{L}(G^+, V_{\text{target}}) \quad \text{s.t.} \quad \|B\|_0 \leq \Delta_{\text{edge}}
\]

- Discrete optimization. Gradient method is prohibitive.
- Relax \( A^+ \in \{0, 1\}^{(N+N_{\text{fake}}) \times (N+N_{\text{fake}})} \) to \( A^+ \in [0, 1]^{(N+N_{\text{fake}}) \times (N+N_{\text{fake}})} \) to use PGD (Xu et al., 2019) ⇒ loss of accuracy.
- RL-based method (Sun et al., 2019)/Greedy search (Wang et al., 2018) ⇒ time consuming.
Algorithm

- Motivation - divide target nodes into several clusters to gain better performance and lower the complexity.
- Target nodes in same cluster are supposed share a certain kind of similarity.
- How to denote this similarity? ⇒ *Most Adversarial Feature*.
- Features of fake nodes derive from cluster centers of Most Adversarial Feature.
表 1: Success rate of targeted attack adding 4 fake nodes. $T$ denotes number of target nodes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cora</th>
<th></th>
<th>Citeseer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T = 3$</td>
<td>$T = 5$</td>
<td>$T = 7$</td>
<td>$T = 10$</td>
</tr>
<tr>
<td>Random</td>
<td>0.07</td>
<td>0.08</td>
<td>0.04</td>
<td>0.05</td>
</tr>
<tr>
<td>NETTACK</td>
<td>0.61</td>
<td>0.57</td>
<td>0.55</td>
<td>0.53</td>
</tr>
<tr>
<td>Sequential Greedy</td>
<td>0.68</td>
<td>0.73</td>
<td>0.72</td>
<td>0.70</td>
</tr>
<tr>
<td>Cluster Attack</td>
<td><strong>0.99</strong></td>
<td><strong>0.93</strong></td>
<td><strong>0.84</strong></td>
<td><strong>0.72</strong></td>
</tr>
</tbody>
</table>

- Generally, our algorithm outperforms existing baselines.
Thanks!