Graph Structure Learning for Robust Graph Neural Networks

KDD 2020

Team10
姜林寬 109065510
鄭宏彬 109062657

Paper Introduction

Pro-GNN Loss Function

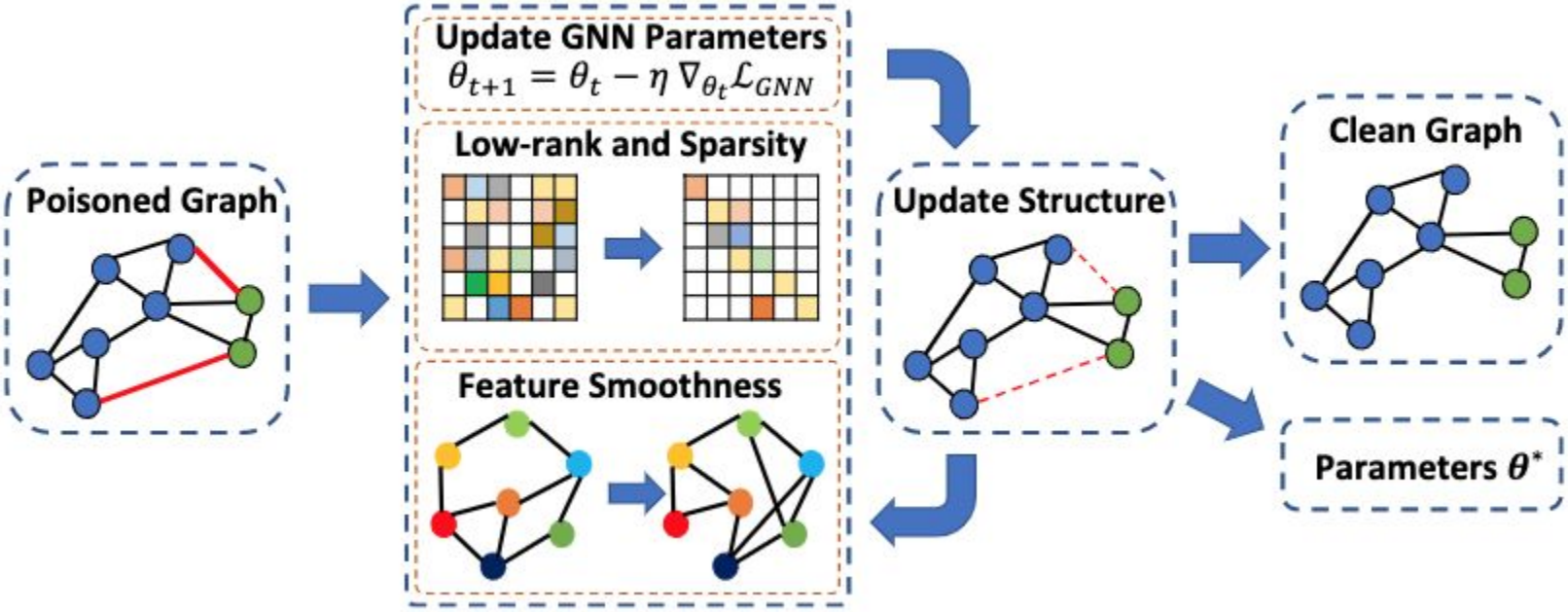
- α, β, γ, λ are four key predefined parameters to control Sparsity, Rank, GNN Loss, feature Smoothness.
- The GNN loss also has the effect of reducing the impact of the attack. Model can learn the parameters of the GNN model while restoring to a

$$\arg\min_{\mathbf{S}\in\mathcal{S},\theta} \mathcal{L} = \mathcal{L}_0 + \lambda\mathcal{L}_s + \gamma\mathcal{L}_{GNN} \tag{9}$$

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

$$s.t. \qquad \mathbf{S} = \mathbf{S}^\top,$$

Pro-GNN framework



Algorithm

Algorithm 1: Pro-GNN

Data: Adjacency matrix $\mathbf{A}$, Attribute matrix $\mathbf{X}$, Labels $\mathcal{Y}_L$, Hyper-parameters $\alpha, \beta, \gamma, \lambda, \tau$, Learning rate η, η'

Result: Learned adjacency $\mathbf{S}$, GNN parameters θ

Initialize $\mathbf{S} \leftarrow \mathbf{A}$

Randomly initialize θ

while *Stopping condition is not met* **do**

  $\mathbf{S} \leftarrow \mathbf{S} - \eta \nabla_S (\|\mathbf{S} - \mathbf{A}\|_F^2 + \gamma \mathcal{L}_{GNN} + \lambda \mathcal{L}_s)$

  $\mathbf{S} \leftarrow \text{prox}_{\eta\beta||.||_*}(\mathbf{S})$

  $\mathbf{S} \leftarrow \text{prox}_{\eta\alpha||.||_1}(\mathbf{S})$

  $\mathbf{S} \leftarrow P_{\mathcal{S}}(\mathbf{S})$

  for *i=1 to* τ **do**

    $g \leftarrow \frac{\partial \mathcal{L}_{GNN}(\theta, \mathbf{S}, \mathbf{X}, \mathcal{Y}_L)}{\partial \theta}$

    $\theta \leftarrow \theta - \eta' g$

Return $\mathbf{S}, \theta$

Paper Weaknesses

Weakness

- 在比較one-stage與two-stages的實驗中，可觀察到two-stages在高擾動圖的防禦上有較好的效果，作者還是得出了one-stage比較好的結論，這部分需要更多的實驗來支持。
- 由於Non-targeted attack的目標是要降低整體GNN的節點分類準確率，其擾動的邊會散布在整個Graph上，我們可以預期在現實世界的大圖上，執行Pro-GNN的方法來重建Graph以防禦metattack需要花費龐大時間。統計顯示由於metattack會傾向於連接兩個特徵不相干的節點，因此我們針對這個攻擊特性引入Overlapping Community Detection Approach，針對整個Graph先做前處理，將Graph做一次分群，利用其演算法時間複雜度低的優勢來輔助大圖結構的去噪還原。

Our Experiments and Achievements

Task 1 Pro-GNN vs Pro-GNN-two

- 而我們重做這個實驗後發現，即使在高擾動率的 Graph 下，one-stage 與 two-stages 的效果也不會差太多，以攻擊方的角度來看，有讓攻擊不被發現的限制在，所以擾動率不會到非常大，即使提高攻擊預算做擾動，one-stage也能發揮效果，所以我們認同作者的結論，Pro-GNN 應該採用 one-stage 。

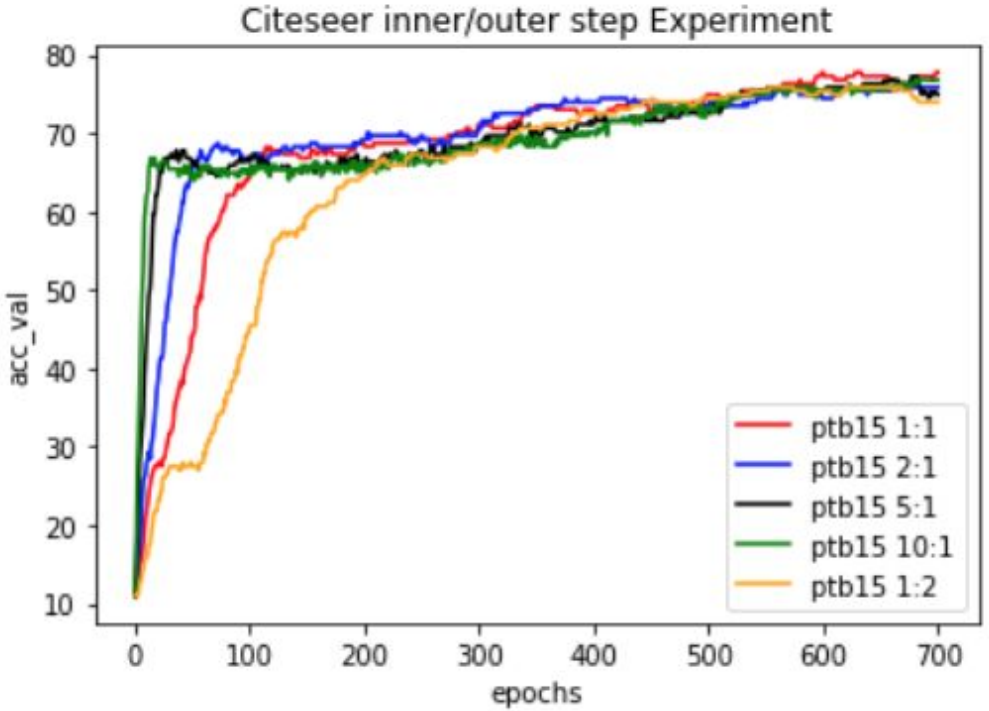
Task 2 Inner-steps vs Outer-steps

- 我們試著找到一個可以兼顧 one-stage 的效果及 two-stages 的效率 (先把圖還原再一次訓練) 的方法來改善 Pro-GNN 的演算法
- Inner-steps指得是每次epoch中對結構還原的次數, Outer-steps指得是對目前結構train gcn的次數, 從 One-Stage vs Two-Stages 的實驗中知道如果先把圖結構都還原再訓練 (two stages), 效果是沒有 one stage 好的, 但我們很好奇一次還原大量的結構是否也會得到相同的結論, 於是設計了以下實驗 :
  - experiment 2-1 : 固定擾動率(皆使用 meta attack)和學習 GNN 參數的 outer-steps , 改變 inner-steps 的次數, 觀察**正確率**和**收斂速度**的變化
  - experiment 2-2 : 固定學習 GNN 參數的 outer-steps , 改變擾動率和 inner-steps 的次數, 觀察在不同擾動程度的圖是否可以得到與前一個實驗相同的結論

Experiment 2-1 : Citeseer

Dataset	ptb rate	Inner-steps	Outer-steps	acc_val (%)	acc_test (%)
Citeseer	15 %	1	2	75.83	70.56
Citeseer	15 %	1	1	77.73	71.27
Citeseer	15 %	2	1	75.83	71.68
Citeseer	15 %	5	1	77.25	72.63
Citeseer	15 %	10	1	77.25	72.75

Citeseer 在不同 inner/outer 比例下 val acc 變化圖



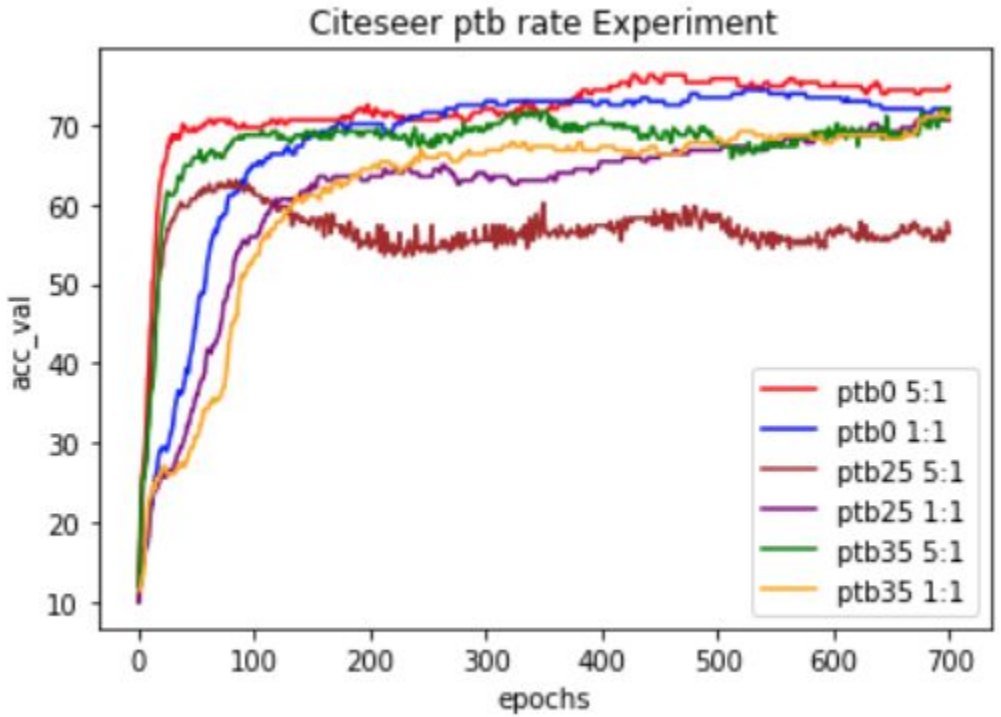
Experiment 2-1 : Cora

Dataset	ptb rate	Inner-steps	Outer-steps	acc_val (%)	test_val (%)
Cora	15 %	1	2	78.71	73.54
Cora	15 %	1	1	80.32	78.22
Cora	15 %	2	1	79.12	76.06
Cora	15 %	5	1	77.11	76.06
Cora	15 %	10	1	75.90	73.14

Experiment 2-2 : Citeseer

Dataset	ptb rate	Inner-steps	Outer-steps	acc_val (%)	acc_test (%)
Citeseer	0 %	1	1	74.41	72.33
Citeseer	0 %	5	1	76.30	74.76
Citeseer	5 %	1	1	76.30	71.92
Citeseer	5 %	5	1	74.41	74.17
Citeseer	15 %	1	1	77.73	71.27
Citeseer	15 %	5	1	77.25	72.63
Citeseer	25 %	1	1	75.83	70.73
Citeseer	25 %	5	1	68.72	68.42
Citeseer	35 %	1	1	73.46	69.73
Citeseer	35 %	5	1	73.46	68.60

Citeseer 在不同擾動率時 inner/outer 比例與 val acc 變化圖



Experiment 2-2 : Cora

Dataset	ptb rate	Inner-steps	Outer-steps	acc_val (%)	test_val (%)
Cora	5 %	1	1	84.74	82.19
Cora	5 %	5	1	84.74	82.95
Cora	15 %	1	1	80.32	78.22
Cora	15 %	5	1	77.11	76.06
Cora	25 %	1	1	77.11	70.17
Cora	25 %	5	1	73.90	67.91

Task 2 Summary

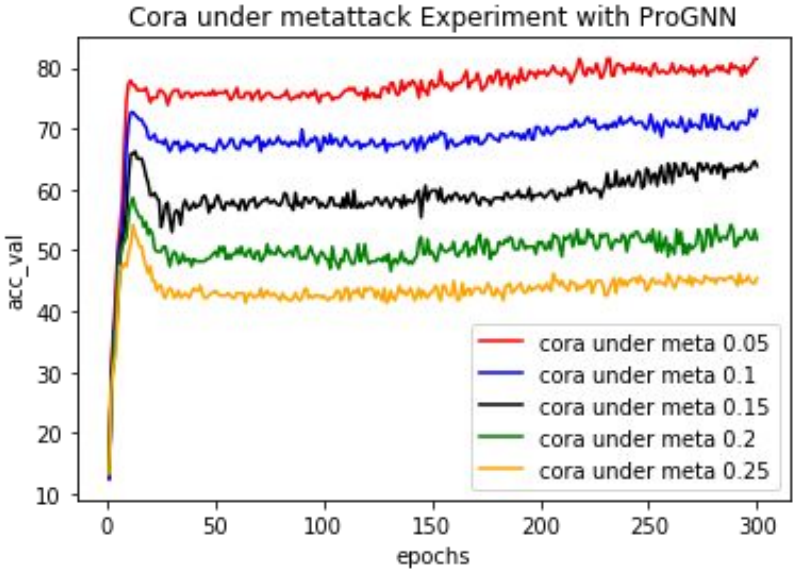
- 提高 inner-steps 比例是個可行的方法，可以把多次結構更新視為一次大的結構更新，但這不等於提高 learning rate，它會是多個結構變動的加總而非單個結構變動的增幅
- 從實驗數據來看，提高 inner-steps 雖然會犧牲一些準確率，但訓練速度可以得到顯著的縮短，也能在一些資料集中得到更好的準確率，同樣印證了原作者提出的訓練圖架構應該與訓練 GNN 參數同時進行，是值得採用的策略

Task 3

- 首先我們重現原作者的 Pro-GNN 在受到不同擾動率 Metattack 攻擊的效果，從結果可以觀察到當擾動率上升時，節點預測準確率會大幅下降。根據實驗延伸一個想法是對於擾動率大的圖結構，我們是否有一個較好的前處理步驟可以施加，讓 Pro-GNN 的準確率可以提升。

Experiment 3-1 : Cora

Dataset	ptb rate	acc_val (%)	test_acc (%)
Cora	5 %	81.53	80.63
Cora	10 %	73.09	68.91
Cora	15 %	66.27	65.64
Cora	20 %	58.63	55.58
Cora	25 %	54.22	51.91

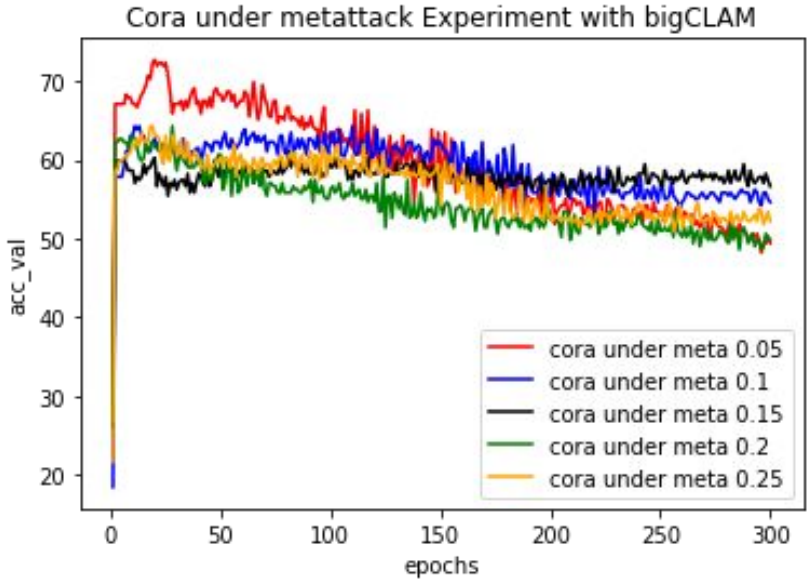


Idea

- 因此以下我們拿受擾動的 Cora 資料先利用 BigCLAM 演算法還原其 labels，並將受擾動的鄰接矩陣以及標籤一同放到 Pro-GNN 中做訓練、驗證以及測試，結果發現我們施加的前處理可以讓 Pro-GNN 訓練出來的 model 準確率維持在一個區間當中，較不容易受到擾動率大小的影響。

Experiment 3-2 : Cora

Dataset	ptb rate	acc_val (%)	test_acc (%)
Cora	5 %	72.69	63.78
Cora	10 %	64.26	60.26
Cora	15 %	61.04	56.79
Cora	20 %	64.26	59.51
Cora	25 %	64.26	58.55



Task 3 Summary

- 施加這個前處理方法的結果雖然讓Pro-GNN在擾動率小的情況之下表現比較差，但根據結果顯示可以發現在圖結構受到較高擾動的情形下，利用BigCLAM可以小幅度的提升Pro-GNN的準確率。由於隨著擾動率提高，鄰接矩陣變得與原圖差異相當巨大，將其配合原標籤放入Pro-GNN中訓練已經無法為模型帶來足夠正確的訓練資訊了。因此前處理是可以用來提高Pro-GNN準確率而採用的策略之一。

Conclusion

Conclusion

- Pro-GNN 應該採用 one-stage 的方式訓練，且在攻擊模型會限制攻擊預算的情況下，one-stage 可以取得比 two-stages 好很多的效果。
- 透過實驗發現，在 Pro-GNN 演算法中，在單個 epoch 中，對圖多進行幾次還原更新再訓練會犧牲一部分正確率換到更快的收斂速度，在某些 dataset 不受影響甚至會有更好的準確率。
- 重現論文在受到 metattack 攻擊之下的模型表現，我們發現加入對擾動圖適當的前處理可以提升 Pro-GNN 在受到高擾動率時的準確率。

END