

Nettack

Adversarial Attacks on Neural Networks for Graph Data

KDD 2018 图神经网络对抗攻击开山之作

[CSDN笔记](#)

整个图: Mettack

单目标攻击: Nettack 通过攻击某个节点 (攻击者 attacker) 实现让另一个节点 (目标 target) 的误分类

攻击

攻击理论

攻击目标分两类:

- 图结构攻击 structure attacks
- 特征攻击 feature attacks

攻击节点分两类:

- Target 目标节点: 让模型错误分类的结点
- Attackers 攻击者结点: 攻击者可以操作的结点

攻击方式:

- direct attack 直接攻击: 攻击者可以直接操作目标结点, 目标结点 == 攻击者结点
- influence attack 推理攻击: 攻击者只能操作除目标结点以外的结点, 目标结点 \notin 攻击者结点

Target node $t \in V$: node whose classification label the attacker wants to change

Attacker nodes $S \subset V$: nodes the attacker can modify

Direct attack ($S = \{t\}$)

- Modify the **target's** features



Example

Change website content

- Add connections to the **target**



Buy likes/followers

- Remove connections from the **target**



Unfollow untrusted users

Indirect attack ($t \notin S$)

- Modify the **attackers'** features



Example

Hijack friends of target

- Add connections to the **attackers**



Create a link/spam farm

- Remove connections from the **attackers**



http://blog.cern.net/weixin_49393427

设定攻击范围 Δ :

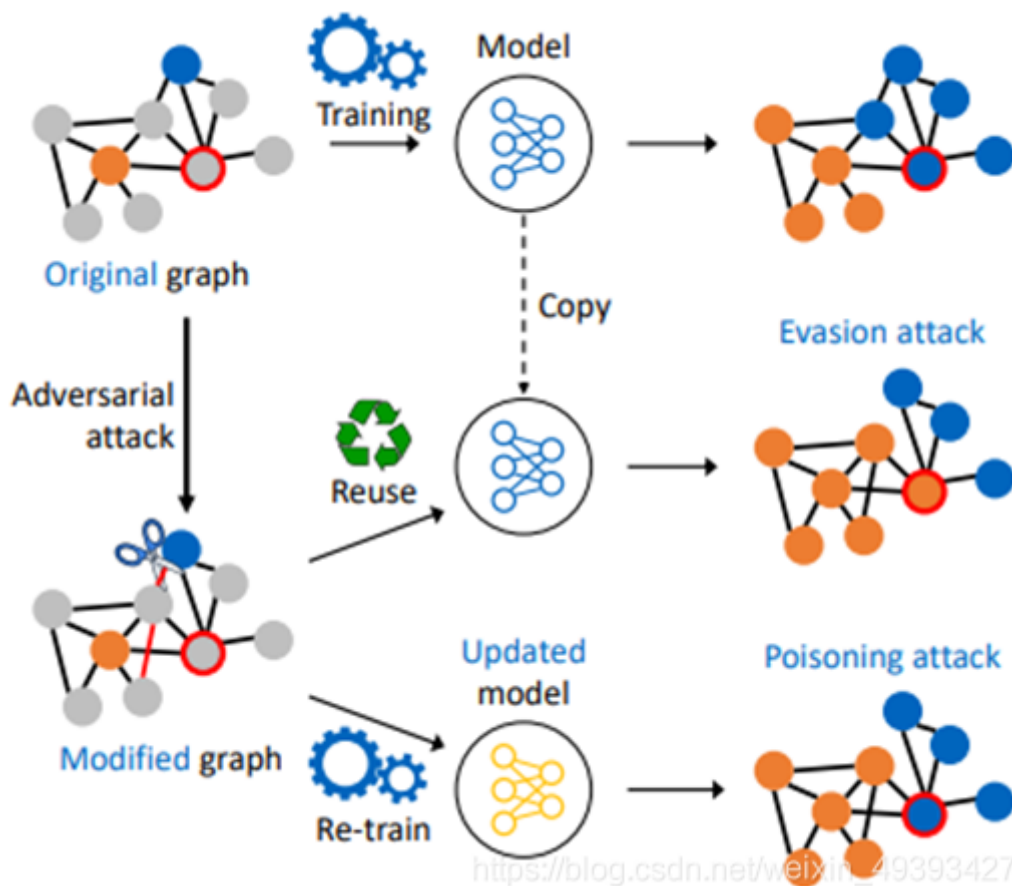
$$\sum_u \sum_i |X_{ui}^{(0)} - X'_{ui}| + \sum_{u < v} |A_{uv}^{(0)} - A'_{uv}| \leq \Delta$$

攻击目标:

$$\begin{aligned} & \arg \max_{(A', X') \in \mathcal{P}_{\Delta, \mathcal{A}}^{G_0}} \max_{c \neq c_{old}} \ln Z_{v_0, c}^* - \ln Z_{v_0, c_{old}}^* \\ & \text{subject to } Z^* = f_{\theta^*}(A', X') \text{ with } \theta^* = \arg \min_{\theta} L(\theta; A', X') \end{aligned}$$

对于参数 θ 的考虑，对于攻击后的图 G' ，应当使用新训练的 θ^* ，考虑到过渡性学习（transductive learning），使用静态参数：原始图像的训练参数。

场景



- 投毒攻击（poisoning attack）
 - 发生在模型被训练前，攻击者可以在训练数据中投毒，导致训练的模型出现故障
- 逃逸攻击（evasion attack）
 - 发生在模型被训练以后或者测试阶段，模型已经固定了，攻击者无法对模型的参数或者结构产生影响

主要问题

如何有效的攻击

图像：连续特征，可以采用基于梯度构造干扰

图：离散型数据，没有梯度

第一，扰动是不被注意到的

第二，确保攻击者不能修改整个图，允许的扰动数目是有限制的

保留图的结构性 (固有特征)

图结构最突出的特征是它的度分布，使用幂律分布来描述：

$$p(x) \propto x^{-\alpha}$$

缩放参数 α 的表达式：

$$\alpha_G \approx 1 + |\mathcal{D}_G| \cdot \left[\sum_{d_i \in \mathcal{D}_G} \log \frac{d_i}{d_{\min} - \frac{1}{2}} \right]^{-1}$$

最大似然估计：

$$l(\mathcal{D}_x) = |\mathcal{D}_x| \cdot \log \alpha_x + |\mathcal{D}_x| \cdot \alpha_x \cdot \log d_{\min} - (\alpha_x + 1) \sum_{d_i \in \mathcal{D}_x} \log d_i$$

保留节点特征

特征的共现关系

反例：如果两个节点都没某个特征，经过攻击，两个节点都有了这个特征，就能增加节点的相似性。

在特征共现图上随机游走，如果有相当大的概率到达一个新加入的特征，那么就认为这个扰动的加入是不被注意的

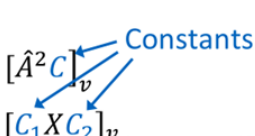
攻击

代理模型 Surrogate model

- ❑ Linear **surrogate model** based on two-layer GCN.
- ❑ Enables computation of the **exact impact** of a perturbation **efficiently** and in **closed form**.
- ❑ Attacker chooses perturbation that **maximizes loss** on the surrogate model (one at a time).

Linearize classifier: $Z = f_{\theta}(A, X) = \text{softmax}(\hat{A} \cancel{R} \cancel{X} LU(\hat{A} X W^{(1)}) W^{(2)})$

Simplified equation: $\log Z' = \hat{A}^2 X W'$

Structure perturbations: $\max_{\hat{A}} \mathcal{L}'(\log Z'_v) \text{ where } \log Z'_v = [\hat{A}^2 C]_v$ 

Feature perturbations: $\max_X \mathcal{L}'(\log Z'_v) \text{ where } \log Z'_v = [C_1 X C_2]_v$

为了能够量化扰动的效果，同时简便计算，所以提出了一个替代模型

代理模型使用两层的GCN，把激活函数做了线性的替换

扰动评价

代理模型损失函数：

$$\mathcal{L}_s(A, X; W, v_0) = \max_{c \neq c_{old}} [\hat{A}^2 X W]_{v_0 c} - [\hat{A}^2 X W]_{v_0 c_{old}}$$

目标：找到扰动的图损失最大

$$\arg \max_{(A', X') \in \hat{\mathcal{P}}_{\Delta, \mathcal{A}}^{G^0}} \mathcal{L}_s(A', X'; W, v_0).$$

评分函数:

$$s_{struct}(e; G, v_0) := \mathcal{L}_s(A', X; W, v_0)$$

$$s_{feat}(f; G, v_0) := \mathcal{L}_s(A, X'; W, v_0)$$

算法

Algorithm 1: NETTACK: Adversarial attacks on graphs

Input: Graph $G^{(0)} \leftarrow (A^{(0)}, X^{(0)})$, target node v_0 ,
attacker nodes \mathcal{A} , modification budget Δ
Output: Modified Graph $G' = (A', X')$

Train surrogate model on $G^{(0)}$ to obtain W // Eq. (13);

$t \leftarrow 0$;

while $|A^{(t)} - A^{(0)}| + |X^{(t)} - X^{(0)}| < \Delta$ **do**

$C_{struct} \leftarrow \text{candidate_edge_perturbations}(A^{(t)}, \mathcal{A})$;

$e^* = (u^*, v^*) \leftarrow \arg \max_{e \in C_{struct}} s_{struct}(e; G^{(t)}, v_0)$;

$C_{feat} \leftarrow \text{candidate_feature_perturbations}(X^{(t)}, \mathcal{A})$;

$f^* = (u^*, i^*) \leftarrow \arg \max_{f \in C_{feat}} s_{feat}(f; G^{(t)}, v_0)$;

if $s_{struct}(e^*; G^{(t)}, v_0) > s_{feat}(f^*; G^{(t)}, v_0)$ **then**

$G^{(t+1)} \leftarrow G^{(t)} \pm e^*$;

else $G^{(t+1)} \leftarrow G^{(t)} \pm f^*$;

$t \leftarrow t + 1$;

return $G^{(t)}$

// Train final graph model on the corrupted graph $G^{(t)}$;

类似贪心思想，每次找到使得Loss最大的扰动

缺点：由于是贪心思想，会陷入局部最优

想法

类似HoneyPot 蜜罐攻击的思想，认为制造类似梯度陷阱的漏洞，让攻击陷入其中

图像数据比较大，能够进行陷阱制作但不影响模型效果，但是图结构不行，图的结构性是度分布。

实现

(i) the 10 nodes with highest margin of classification, i.e. they are clearly correctly classified

(ii) the 10 nodes with lowest margin (but still correctly classified)

(iii) 20 more nodes randomly

直接攻击, Netack

影响攻击, Netack-In (从目标的附近随机挑选5个节点作为攻击者)

效果

分类正确率:

Attack method	Cora			Citeseer			Polblogs		
	GCN	CLN	DW	GCN	CLN	DW	GCN	CLN	DW
Clean	0.90	0.84	0.82	0.88	0.76	0.71	0.93	0.92	0.63
NETTACK	0.01	0.17	0.02	0.02	0.20	0.01	0.06	0.47	0.06
FGSM	0.03	0.18	0.10	0.07	0.23	0.05	0.41	0.55	0.37
RND	0.61	0.52	0.46	0.60	0.52	0.38	0.36	0.56	0.30
NETTACK-IN	0.67	0.68	0.59	0.62	0.54	0.48	0.86	0.62	0.91

直接攻击的效果要比推理攻击更好

- FGSM, 快速梯度下降法, 基于梯度的方法应用于离散数据并不是一个好的选择, 实验表明在邻接矩阵中改变元素时, 梯度和实际的损失变化不一致
- RND, 改变图的结构, 随机采样点然后添加边

分类效果图:

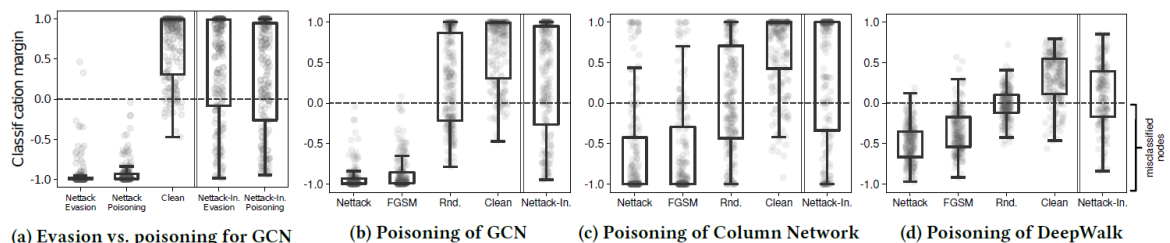


Figure 6: Results on Cora data using different attack algorithms. Clean indicates the original data. Lower scores are better.

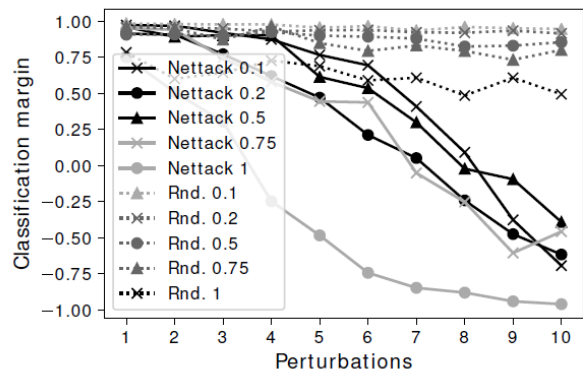
不同目标度数的分类精度:

度数越高越难受到攻击

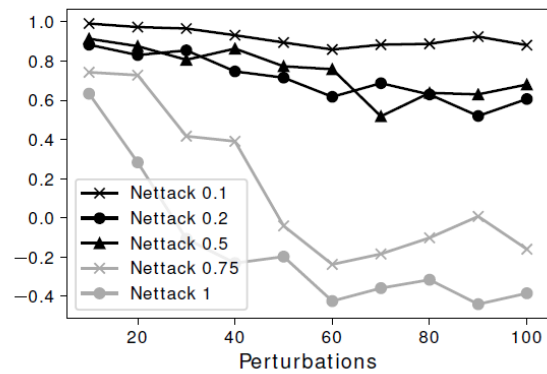
	[1;5]	[6;10]	[11;20]	[21;100]	[100; ∞)
Clean	0.878	0.823	1.0	1.0	1.0
NETTACK	0.003	0.009	0.014	0.036	0.05

知识受限的情况下, 攻击效果

知识受限: 攻击时候替代模型只在目标节点附近的一定区域内训练



(a) Direct attack



(b) Influence attack

Figure 7: Attacks with limited knowledge about the data