Professorship of Data Mining and Analytics
Department of Informatics
Technical University of Munich

Adversarial Attacks on Neural Networks for Graph Data

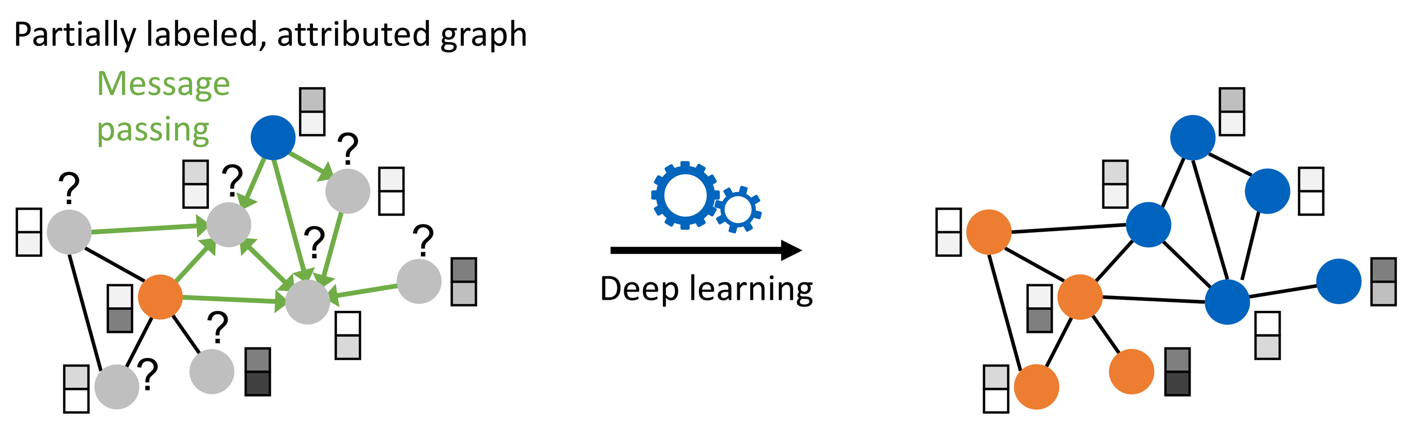
Daniel Zügner, Amir Akbarnejad, Stephan Günnemann

Motivation

- Deep learning for graphs has brought great improvements on many tasks, e.g. node representation learning or node classification.
- Adversaries are a real threat in the applications where these models are likely to be used, e.g. search engines, recommender systems.
- Before deploying these models, we have to answer the question:

Are deep learning models for graphs robust with respect to adversarial attacks?

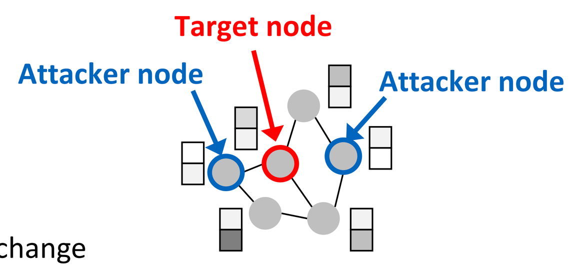
Focus: Semi-supervised Node Classification



- Message passing could lead to higher robustness due to local averaging; or lower robustness due to propagation effects in the network.

Possible Attacks on Graphs

- In addition to feature vectors, adversarial attacks on graphs can modify the graph structure.
- Even more critical: a node can be attacked without direct access due to network effects.

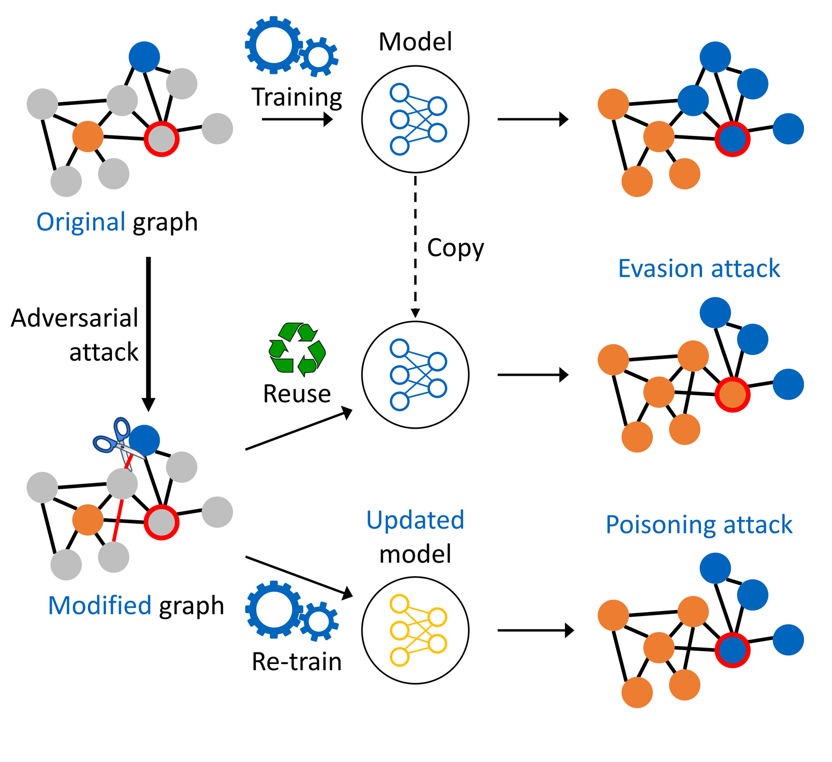


Target node $t \in V$: node whose classification label the attacker wants to change

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

Direct attack ($S = \{t\}$)	Example	Indirect attack ($t \notin S$)	Example
• Modify the target's features	Change website content	• Modify the attackers' features	Hijack friends of target
• Add connections to the target	Buy likes/ followers	• Add connections to the attackers	Create a link/ spam farm
• Remove connections from the target	Unfollow untrusted users	• Remove connections from the attackers	Create a link/ spam farm

Poisoning vs. Evasion Attacks



- Transductive learning: data consists of labeled and unlabeled samples; all data used for training.
- Evasion attack: Modify data to fool a static classifier.
- But: modifications are on the training data (transductive setting).
- Re-training can restore the predictions
- Important for a realistic attack: Impact after model re-training (poisoning).

Formal Problem Definition

- Goal: maximize classification loss of a single target node.
- Measure impact after training the classifier on the modified data (poisoning attack).
- Enforce unnoticeability constraints for subtle perturbations.
- Challenge: discrete data (gradient less reliable because we need to extrapolate).

$$(A^c, X^c) = \arg\max_{A',X'} \max_{c \neq c_{old}} \log Z^*_{v,c} - \log Z^*_{v,c_{old}}$$

Classifier: $\text{where } Z^* = f_{\theta^*}(A', X')$,

Training: $\text{with } \theta^* = \arg\min_\theta \mathcal{L}(\theta; A', X')$

Unnoticeability: $s.t.\ (A', X') \approx (A, X)$

Budget constraint: $\text{and } \|A' - A\|_0 + \|X' - X\|_0 < \Delta$

$A \in \{0,1\}^{N\times N}$: adjacency matrix
$X \in \{0,1\}^{N\times D}$: node attributes
A': modified structure
X': modified features
v : target node

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: $Z = f_\theta(A, X) = \cancel{soft}max(\hat{A}\ \cancel{ReLU}(\hat{A}XW^{(1)})W^{(2)})$

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

Constants

Unnoticeability Constraints

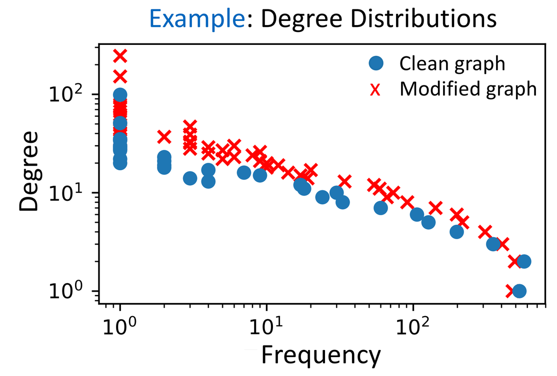
$(A', X') \approx (A, X)$: What are sensible measures of 'closeness' for graphs?

Structure perturbations

- Fundamental property of graphs: degree distribution.
- Hypothesis test: Were the original and modified degree distributions D and D' generated by the same underlying powerlaw distribution?

H_0: D and D' come from the same powerlaw distribution.
H_1: they come from separate powerlaw distributions.

Example: Degree Distributions



Approx. power law exponent: $\alpha(D) \approx 1 + |D| \cdot \left[\sum_{d_i \in D} \log \frac{d_i}{d_{min} - 0.5}\right]^{-1}$

Log likelihood: $l(D) = |D| \cdot \log \alpha [1 + \log d_{min}] + [\alpha + 1] \sum_{d_i \in D} \log d_i$

Hypothesis likelihoods: $l(H_0) = l(D \cap D')$; $l(H_1) = l(D) + l(D')$

Test statistic: $\Lambda(D, D') = -2 \cdot l(H_0) + 2 \cdot l(H_1)$

- For large N, Λ follows a χ^2 distribution with one degree of freedom.
- We choose a very conservative p-value constraint (p=0.95), i.e. we can detect small changes.
- Bookkeeping enables incremental updates to all terms in constant time.

Feature perturbations

- Constraint to prevent addition of unrealistic features (words).
- Co-occurrence graph of the node attributes in the data:

$C = (F, E), F \in \{0,1\}^D, E \subseteq F \times F$
$(f_1, f_2) \in E$: f_1 and f_2 co-occur in at least one node in the dataset.
Node u's features: $S_u = \{j \mid X_{uj} \neq 0\}$

Top inserted words to papers of the class neural networks:

w/ constraint	w/o constraint
probabilistic	efforts
bayesian	david
inference	family

Probability of reaching a feature by a random walker on C starting at node u in one step:

$$p(i|S_u) = \frac{1}{|S_u|} \sum_{j \in S_u} 1/d_j \cdot E_{ij} > \sigma; \text{ in our experiments we set } \sigma = 0.5 \cdot \frac{1}{|S_u|} \sum_{j \in S_u} 1/d_j.$$

- Encourages addition of features that co-occur with many of node u's features.
- Discourages addition of features that only co-occur with unspecific features, i.e. stopwords.

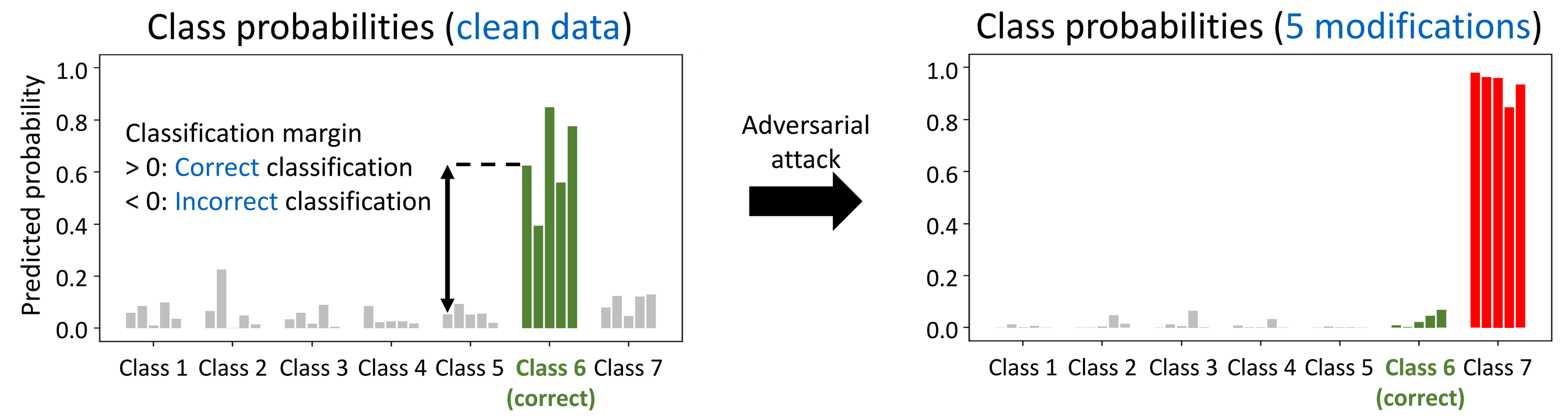
Experimental Results

- Transfer experiments: Graph Convolutional Network (GCN), Column Network, DeepWalk.
- Perturbation budget is d+2, where d is the target's degree.
- Evaluation on 5 different splits; 10x re-training per attack.

Approach

- Attacks via linear surrogate model while enforcing unnoticeability constraints.
- Train deep learning models on the poisoned data and evaluate the drop in performance.
- No access to the classifier is needed for the attack.

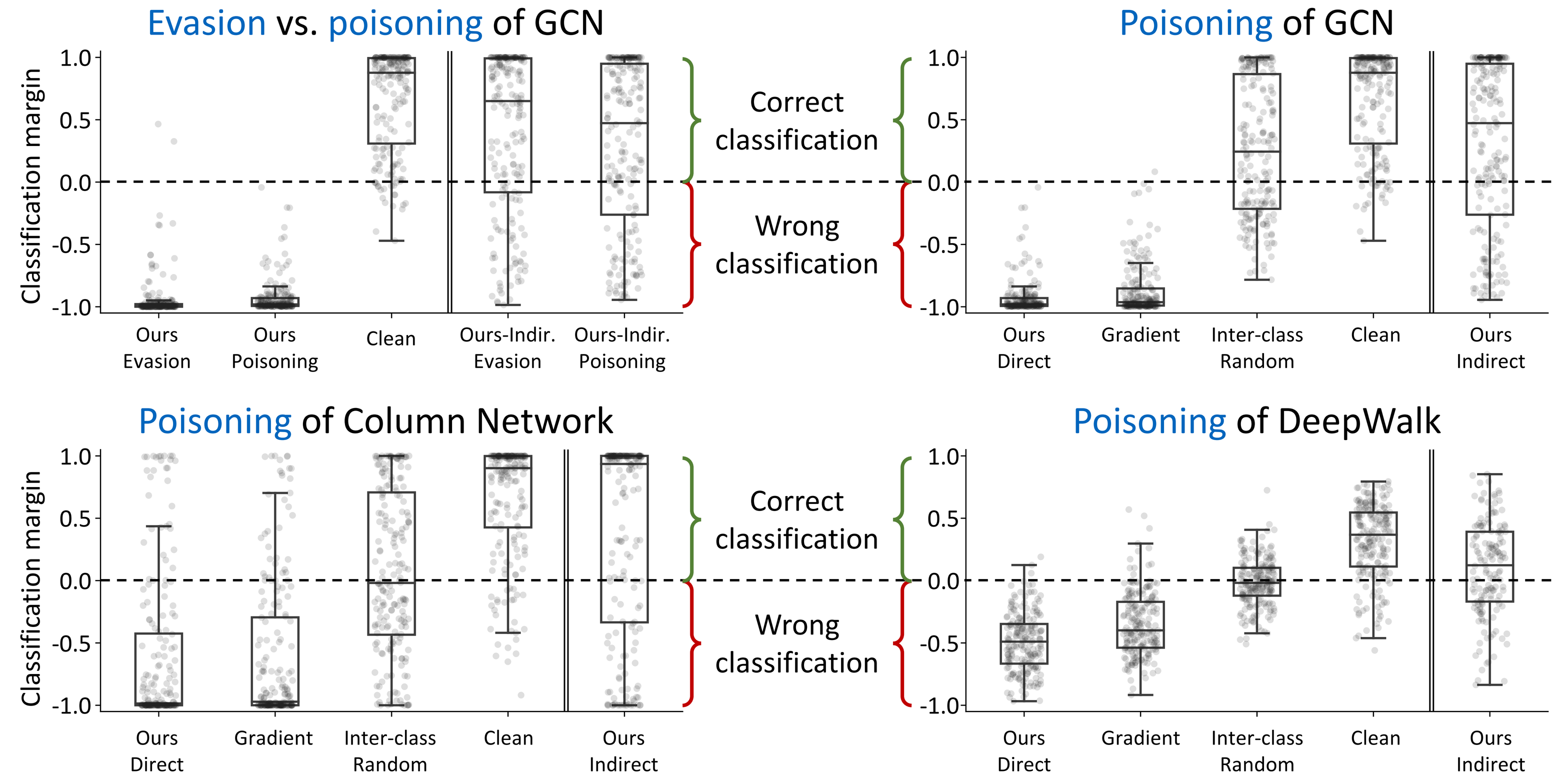
Example Attack



Results

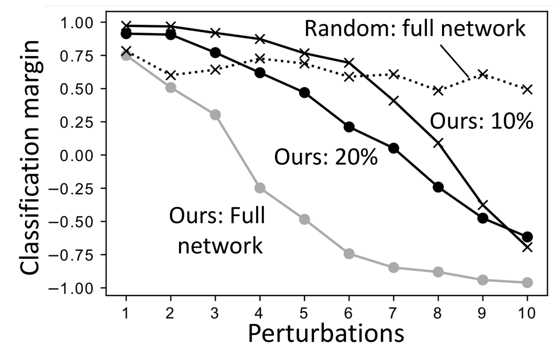
Attack method	Cora GCN	Cora CLN	Cora DW	Citeseer GCN	Citeseer CLN	Citeseer DW	PolBlogs GCN	PolBlogs CLN	PolBlogs DW
Clean	0.90	0.82	0.84	0.88	0.71	0.76	0.93	0.63	0.92
Ours	0.01	0.17	0.02	0.02	0.20	0.01	0.06	0.47	0.06
Gradient	0.03	0.18	0.10	0.07	0.23	0.05	0.41	0.55	0.37
Inter-class Random	0.61	0.52	0.46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

Table: Share of correct classifications of target nodes after attack and re-training



Limited knowledge

- Only provide a small part of the network around the target node to the surrogate model to attack.
- Evaluation with GCN on the complete graph.
- Attacks are successful even in this restricted setting.



Conclusion

- We propose an efficient algorithm for adversarial attacks on deep learning for graphs.
- No access to the classifier is needed for the attack due to the surrogate model approach.
- Our attacks are successful even under restrictive attack scenarios, e.g. no access to target node or limited knowledge about the graph.

KDD 2018

August 19 – 23, 2018
London, United Kingdom

github.com/danielzuegner/nettack