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

Adversarial Attacks on Node Embeddings via Graph Poisoning

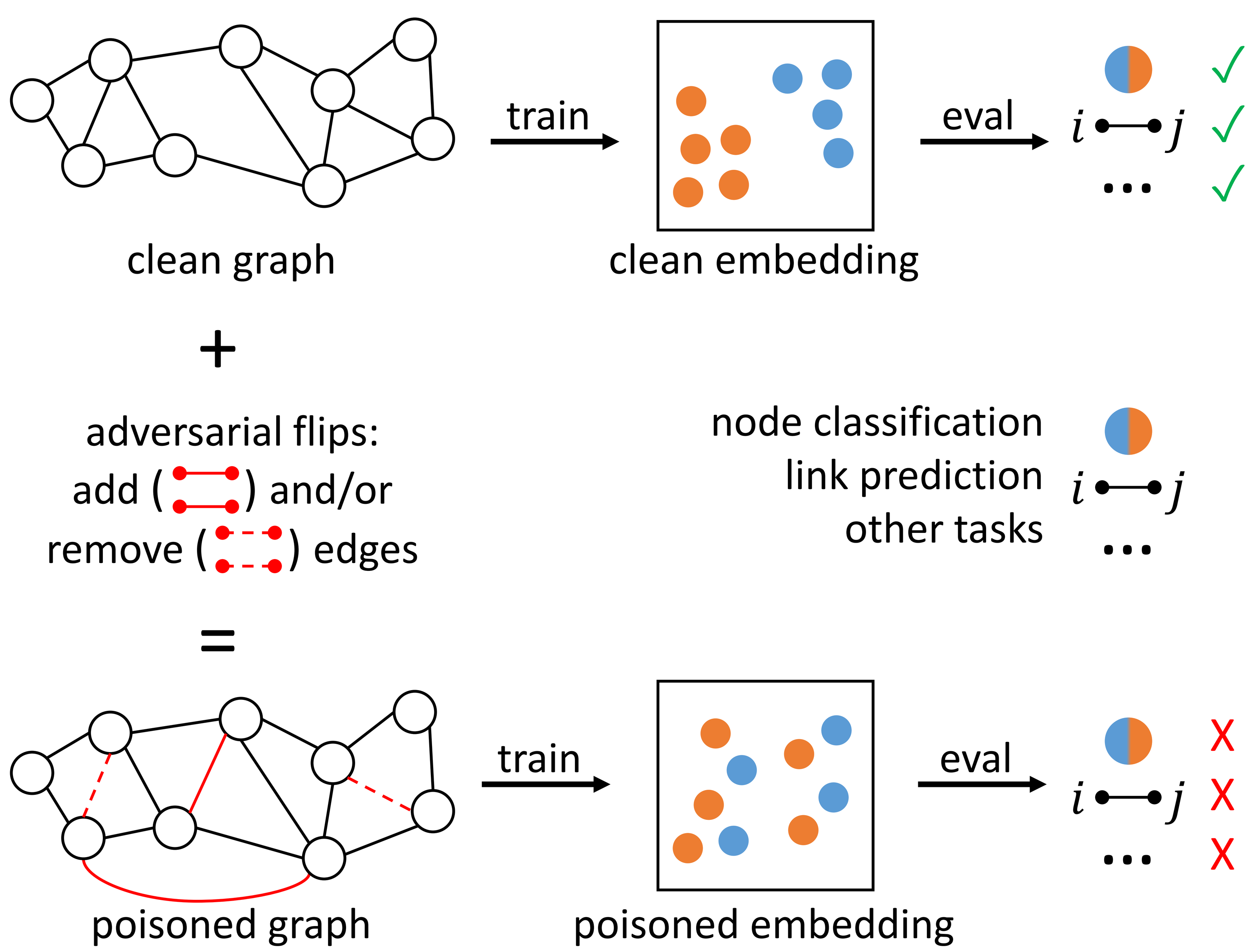
Aleksandar Bojchevski, Stephan Günnemann

Overview

- ❑ Node embeddings are vulnerable to adversarial attacks.
- ❑ Exploit connections to matrix factorization and the graph spectrum to find adversarial edges.
- ❑ Relatively few perturbations degrade the embedding quality and the performance on downstream tasks.

Motivation

In domains where we use node embeddings (e.g. the Web) adversaries are common and false data is easy to inject.
Research question: Are node embeddings robust to attacks?



Challenges

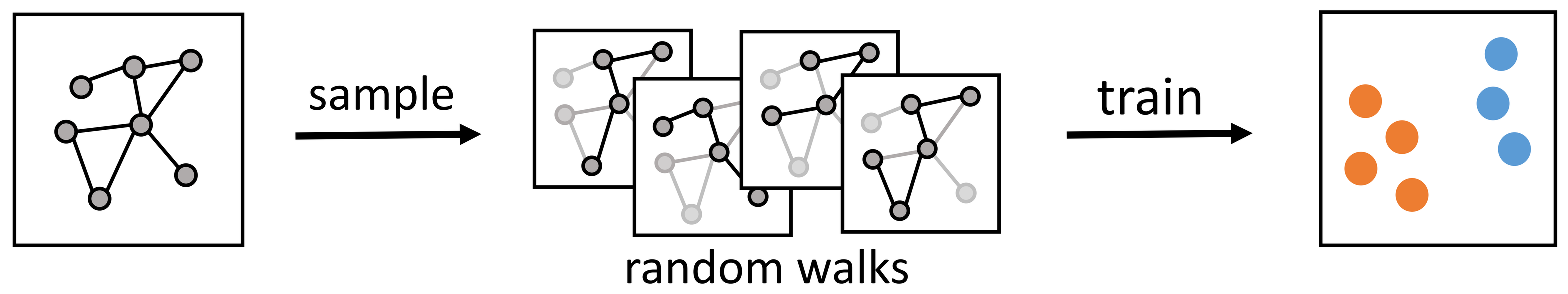
- ❑ Combinatorial bi-level optimization problem.
- ❑ Inner optimization includes non-differentiable sampling.

$$G_{poisoned} = \underset{\substack{G \in all\ graphs \\ |G_{clean} - G| = budget}}{\operatorname{argmax}} \ \min_Z \mathcal{L}(G, Z)$$

$$\mathcal{L}(G,Z) = \mathcal{L}(\{r_1, r_2, \ldots\}^G, Z), \quad r_i = rnd\_walk(G)$$

Background: DeepWalk

Treat random walks as sentences. Train Word2Vec embeddings.



1. DeepWalk as Matrix Factorization

DeepWalk is equivalent to factorizing the Shifted Positive Pointwise Mutual Information (PPMI) matrix.

window size T   transition/degree/adjacency matrix

$$\widetilde{M}_{ij} = \log \max\{cS_{ij}, 1\} \qquad S = \left(\sum_{r=1}^{T} P^r\right) D^{-1} \qquad P = D^{-1}A$$

Embeddings $Z^* = U_K \Sigma_K^{1/2}$ obtained via SVD of $\widetilde{M} = U\Sigma V^T$

2. Express the optimal $\mathcal{L}$ via the graph spectrum

Rewrite S in terms of the generalized spectrum of A. Optimal loss is a function of the eigenvalues ⇒ Inner optimization is eliminated.

$$Au = \lambda Du \qquad S = U\left(\sum_{r=1}^{T} \Lambda^r\right) U^T \qquad \min_Z \mathcal{L}(G,Z) = f(\lambda_i, \lambda_{i+1}, \ldots)$$

generalized eigenvalues/vectors   simple function (sums) of eigenvalues

3. Approximate the poisoned graph's spectrum

Compute the change using Eigenvalue Perturbation Theory.

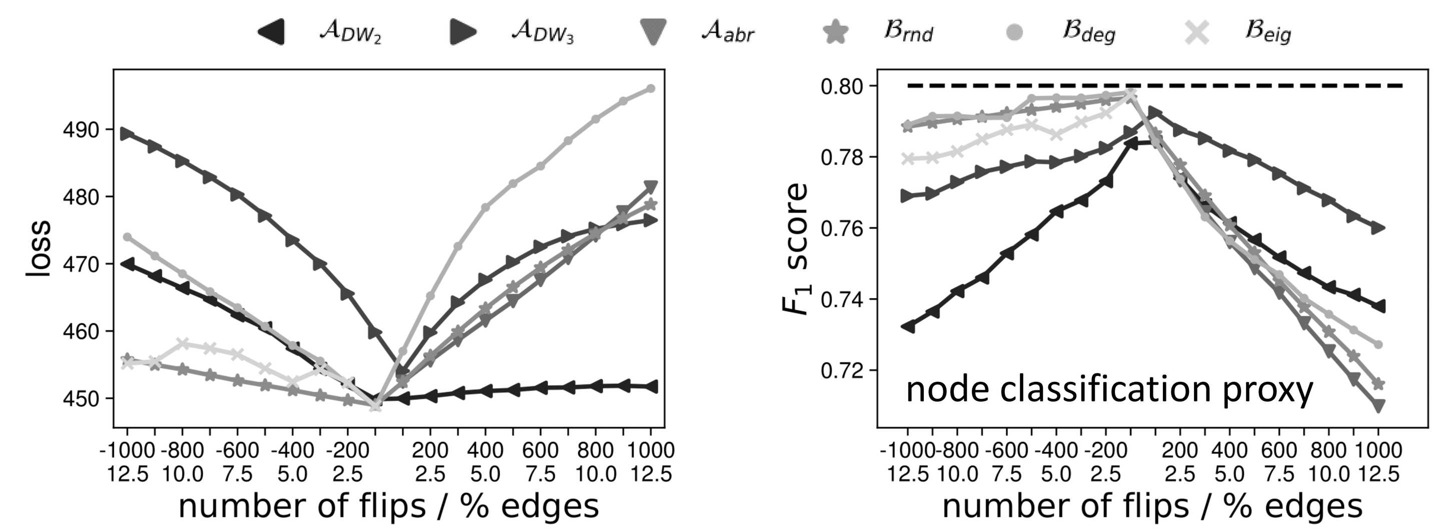
$$A_{pois.} = A_{clean} + \Delta A$$
$$\lambda_{pois.} = \lambda_{clean} + u_{clean}^T(\Delta A + \lambda_{clean}\Delta D)u_{clean}$$
$$u_{pois.} = u_{clean} - (A - \lambda_{clean} D)^{+}(\Delta A - \Delta\lambda D - \lambda_{clean}\Delta D)u_{clean}$$

Overall algorithm:

1. Compute generalized eigenvalues/vectors (Λ/U) of the graph
2. For all candidate edge flips (i,j) compute the change in Λ/U
3. Greedily pick the top candidates leading to largest loss $\mathcal{L}$

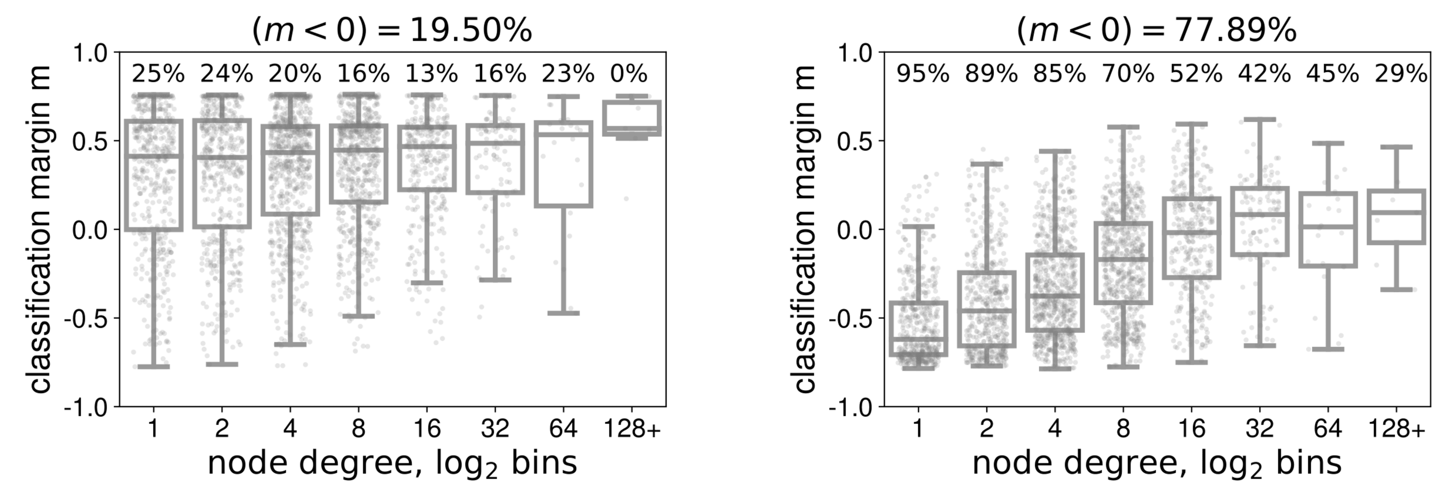
General attack

Goal: decrease the overall quality of the embeddings.



Targeted attack

Goal: attack a specific node and/or a specific downstream task.



Transferability

Our selected adversarial edges transfer to other methods.

budget	DW SVD	DW SGNS	node-2vec	Spectral Embed.	Label Prop.	GCN
250	-7.59	-5.73	-6.45	-3.58	-4.99	-2.21
500	-9.68	-11.47	-10.24	-4.57	-6.27	-8.61

Analysis of adversarial edges

There is no simple heuristic that can find the adversarial edges.