1 Task

Attack graph classification model.

- Change the prediction on a specific vertex.
- Change binary features of the vertex or add/remove edges in the graph.

2 Overview Figure

![Overview Figure]

Figure 1: Small perturbations of the graph structure and node features lead to misclassification of the target.

3 Graph Attack Problem

- **Setting:** Node Classification Task.
  - Graph $G^{(0)} = (A^{(0)}, X^{(0)})$, where $A \in \{0,1\}^{N \times N}$ is a adjacency matrix, and $X \in \{0,1\}^{N \times D}$ is a binary feature matrix (each vertex has $D$ dimension feature).
  - Model $f_\theta(A, X) = Z$ gives a probability distribution $Z$ on labels $c$.
  - Cross-entropy loss $L(\theta; A, X) = \sum_{c \in V_L} \ln Z_{c,v_c}$

- **Problem:**
- Give original graph \( G^{(0)} = (A^{(0)}, X^{(0)}) \), generate a perturbed \( G' = (A', X') \).
- Target: A certain node \( v_0 \)
- The modification is limited to a set of attacker nodes \( A \subseteq V \), that:
  \[
  X^{(0)}_{ui} \neq X'_{ui} \Rightarrow u \in A
  \]
  \[
  A^{(0)}_{uv} \neq A'_{uv} \Rightarrow u \in A \cup v \in A
  \]
- The modification is limited by some budget \( \Delta \), detailed calculation in next part.

- **Formal Definition:**

  **Problem 1** Given a graph \( G^{(0)} = (A^{(0)}, X^{(0)}) \), a target node \( v_0 \) and attacker nodes \( A \). Let \( c_{\text{old}} \) denote the predicted class of \( v_0 \) with \( G^{(0)} \). Determine:

  \[
  \arg\max_{A', X' \in P_{G^{(0)}}} \max_{c \neq c_{\text{old}}} \ln Z^*_v, c - Z^*_v, c_{\text{old}}
  \]

  Subject to \( Z^* = f_{\theta^*}(A', X') \) (1)

  It’s a **poisoning attack** if \( \theta' = \arg\min_{\theta} L(\theta; A', X') \), i.e., the parameter of model \( f \) is retrained.

  It’s a **evasion attack** if \( \theta' = \theta = \arg\min_{\theta} L(\theta; A^{(0)}, X^{(0)}) \)

4 **NETTACK**

- **In one sentence:** Sort the importance of the possible attacks and pick the best ones till budget is filled. A greedy attack.

- **Budget definition:**
  - Evaluate the size of a perturbation is hard in graph (and any discrete data type).
  - For the graph modification part, use a statistical two-sample test to evaluate the similarity between perturbed sample and original sample.
  
  The scaling factor is estimated by:

  \[
  \alpha_G \approx 1 + |D_G| \cdot \left| \sum_{d_i \in D_G} \log \frac{d_i}{d_{\text{min}} - \frac{1}{2}} \right|^{-1}
  \]
Likelihood is estimated by:

\[
l(D_G) = |D_G| \cdot \log \alpha_G + |D_G| \cdot \alpha_G \cdot \log d_{\min} + (\alpha_G + 1) \sum_{d_i \in D_G} \log d_i \tag{3}
\]

And finally the test statistic is

\[
\Lambda(G^{(0)}, G') = -2 \cdot l(D_G^{(0)} \cup D_G) + 2 \cdot (l(D_G^{(0)}) + l(D_G)) < \tau \approx 0.004 \tag{4}
\]

– For the feature modification part, make sure the co-occurrence of features is preserved. That is, if two features are never co-occurred in \( G^{(0)} \), the change is noticeable.

Let \( S_u \) be the all the features present for node \( u \), the addition of a feature \( i \) is acceptable if

\[
p(i|S_u) = \frac{1}{|S_u|} \sum_{j \in S_u} (1/d_j) \cdot E_{ij} > \sigma \tag{5}
\]

• **Attack:**

  – Too hard to handle, so instead attack a surrogate model:

\[
Z = \text{softmax}(\hat{A}\hat{A}X^{(1)}W^{(2)}) = \text{softmax}(\hat{A}^2XW) \tag{6}
\]

Which is original GCN with non-linear part removed.

– Define score of add/remove an edge \( e = (u, v) \) or add/remove a feature \( f = (u, i) \):

\[
\begin{align*}
s_{\text{struc}}(e, G, v_0) &= L_s(A', X; W, v_0) \tag{7} \\
s_{\text{feat}}(f, G, v_0) &= L_s(A, X'; W; v_0) \tag{8}
\end{align*}
\]

Where,

\[
L_s(A, X; W, v_0) = \max_{c \neq c_{old}} [\hat{A}^2XW]_{v_0c} - [\hat{A}^2XW]_{v_0c_{old}} \tag{9}
\]

– Use an iterative approach: Each step find a locally best modification. Stop when the budget is filled.

• **Faster computation:**

  – Calculate only the incremental part of \( \hat{A}^2 \) in a graph edge attack.
  – Calculate only the incremental part of \( X \) in a feature attack.
  – Accelerate the candidate set calculation by pre-processing.
5 Evaluation:

5.1 Dataset

- Cora-ML: A citation dataset with 2,810 nodes, 7,981 edges, and bag-of-word text features.
- Citeseer: A citation dataset with 2,110 nodes, 3,757 edges, and bag-of-word text features.
- POLBLOGS: A blog post dataset with 1,222 nodes, 16,714 edges and bag-of-word text features.

5.2 Baselines

- **Fast Gradient Sign Method** (with projection), which is an attack only on features
- **RND**: Randomly modified the graph structure, which is only applied to the graph not features.

5.3 Result

- Change the prediction of the graph convolutional network model on a certain \( v_0 \).
- Get good result transferred to Column Network and DeepWalk.
- Even with limited knowledge about the data (a subgraph), can still reduce the prediction of \( v_0 \).