Membership Inference Attacks on Graph Neural Networks

Marcus Lassila, Chalmers University of Technology
Department of Electrical Engineering

Motivation

Despite the recent popularity of studying membership inference attacks as an empirical tool to measure the information leakage of a machine learning model about its training data, relatively little work has been done on membership inference attacks against models trained on graph data. Previous work on node-level membership inference attacks[3,4] leaves several key questions unanswered, most notably how to take advantage of the additional graph structure in a principled way. In addition, there has been a recent paradigm shift[1] in how membership inference attacks ought to be evaluated in terms of worst–case metrics rather than average metrics, and previous work on membership inference attacks on graph neural networks must be re-evaluated.

Membership Inference Attacks

In a membership inference attack, an adversary attempts to answer if a particular data sample was part of the training dataset of a machine learning model, given query access to the model. State of the art membership inference attacks[1,2] models the attack as a hypothesis test, defining the hypotheses as some version of

- H_0: The target model was not trained on target sample (x, y)
- H_1: The target model was trained on target sample (x, y)

The distributions of these hypotheses are approximated training many ”shadow” models including and excluding the target sample and computing the empirical distribution of loss values on the target point.

Node Membership Inference Attacks

There are several way to adapt membership inference attacks to graph neural networks. We focus on attacking node classification models, with the goal of determining the membership status of a given node feature-label pair.
In the case of a GNN, there is no unique loss value of a target node since it will depend also on the neighboring nodes in the graph used to query the model. This leads to some new research questions:

- What is the optimal query to use for maximal attack performance?
- Can we compute the optimal query in a practical attack?
- Can we design a new attack tailored to graph structured data that outperforms adaptions of LiRA[1] and RMIA[2] which are the current state of the art membership inference attacks?

References

1. Nicholas Carlini et al. “Membership Inference Attacks From First Principles”
2. Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. “Low-Cost High-Power Membership Inference”
3. Iyiola E. Olatunji, Wolfgang Nejdl, and Megha Khosla. “Membership Inference Attack on Graph Neural Networks”
4. Xinlei He et al. “Node-Level Membership Inference Attacks Against Graph Neural Networks”

Graph Neural Networks

Message-passing graph neural networks follow a general strategy in creating node-embeddings for downstream tasks. The node embedding of a particular node is initialized to its node feature vector and updated by aggregating over the node embeddings of its neighboring nodes,

$$\mathbf{h}_n^{(k+1)} = \mathsf{Update}(\mathbf{h}_n^{(k)}, \mathsf{Aggregate}(\{\mathbf{h}_m^{(k)} : m \in \mathcal{N}(n)\}))$$

where Update and Aggregate are arbitrary differentiable functions. The final node embeddings can be used for node, edge or graph prediction tasks. In our work, we consider node classification models and assume that the node features are private data.

Results

We evaluate the true positive rate at low false positive rate (<1%) on a balanced set of member and non-member nodes. It is much more important that an attack can confidently infer a few (or even one) member node, rather than inferring a lot of member nodes with low confidence[1]. We restrict ourselves to 2-layer GNNs, and investigate the attack predictions using 0-hop, 1-hop and 2-hop queries.

The relevant findings so far are:

- Different nodes are identified as members depending on the neighborhood around the target node used to query the model.
- 1-hop and 2-hop queries performs poorly when the neighborhood contains nodes with different membership status than the target node.
- By building an ensembled attack that combines the predictions made by queries using different neighborhoods, it is possible beat the 0-hop query attack performance (indicating that we have successfully used the graph structure to achieve a stronger attack).
- Sampling many neighborhoods from the underlying graph population distribution and computing a Monte Carlo estimate of the attack statistic, it is possible to also improve the attack performance without assuming knowledge of the exact 2-hop neighborhood.

Open problems

- How to determine the optimal query for a given target node?
- Why does a 0-hop query leak more membership information for only for some nodes?

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM