CS 590-MLG Project Proposal

Elnard Utiushev, Minh Luong Nguyen, Viraat Das

October 8th, 2020

1 Introduction

Graph network models have proven useful in various domains including social network analysis, image recognition, and DNA mapping. Thus, there has been increasing attention paid to adversarial attack. Due to the nature of graph models, its ubiquity, and the massive risk it poses in the wrong hands, adversarial attacks and defense are becoming a more studied topic both in academia and industry.

Adversarial attacks on graph data need to be understood from both the attacker and defender perspective. One idea that attackers employ is generating adversarial sample. Modifying nodes or edges from the original graph can pave ways for various attacks (Sun et al. 2018). The idea behind perturbation is to create a modified graph $\hat{G}$. The following metrics help defined "imperceptible perturbation:"

1. **Edge-level Perturbation** - modifying edges; hard to detect for dynamic networks
2. **Node-level Perturbation** - modifying nodes or changing features of particular nodes
3. **Structure Preserving Perturbation** - attempts to preserve more structure properties of original graph such as total degree and node distribution
4. **Attribute Preserving Perturbation** - modifying features of nodes or edges

There are two types of attack:

1. **Poisoning** - adds adversarial samples into the training dataset; causes poor classification when model is retrained
2. **Evasion** - affects model at the testing phase; generates adversarial samples of the trained model and only changes testing data

The recent focus on adversarial attack techniques such as NETTACK (Zügner, Akbarnejad, and Günnemann 2019), focus in the defense have also been shifted. Literature features multiple ways of going about defense. Our project is based around the detection of adversarial attacks.

Detecting an attack follows under the assumption that data has already been polluted. Using different attack methods such as Fast Gradient Attack (FGA) (Chen et al. 2019) and Nettack (Zügner, Akbarnejad, and Günnemann 2019), we attempt to improve graphical models against them. We hihglight the different metrics and attack techniques to employ them within the context of this project. We propose a framework that will help us detect and localize adversarial attacks on "poisoned" graphs due to adversarial attacks.

2 Motivation

There are two ways to defend machine learning models from adversarial attacks: improving training algorithm to create more robust models that are less susceptible to adversarial attacks, or detecting whether or not an input has been maliciously altered. Much theoretical interest has been given to the first approach; however, in practice, it's easier to build a model to detect a fake or attacked input because a robust classification model should be able to differentiate between a benign input and a malicious one, while the second approach focuses

solely on the task of detecting an adversarial attack. Our project will focus on detecting the presence of adversarial attacks. Another point of theoretical interest in building an adversarial attack detection model is that since the attack is propagated throughout the network, the descendants of an attacked node will have a high probability of getting detected as an attacked node by our model. Therefore, it is possible to "triangulate" the source of the attack without having to run the model on every nodes of the graph. This is especially useful in intractably large graphs such as social networks. Last but not least, if we successfully train a model to differentiate between attacked and real input, then this model could serve as a starting point to training an adversarially robust model for node classification. It is also possible that a model jointly trained to perform both node classification and attack detection would be better at both tasks than when trained individually.

3 Related Works

Zügner, Akbarnejad, and Günnemann (2019) introduce a NETTACK algorithm. They use a surrogate model in order to determine what kind of feature or structural changes will result in incorrect predictions by the main model.

Xu et al. (2020) summarize the ways we can detect graph perturbations. They use a weighted average of LinkPred (which uses a GNN to determine the probability of 2 nodes being connected), GraphGenDetect (which uses a generative model to predict which edges are least likely to be connected), and OutlierDetect (which uses the node class distribution of the neighbourhood nodes) to determine a score for every edge.

Jin et al. (2020) proposes a framework called Pro-GNN that learn a structural graph along with a robust graph neural network model. They first learn the poisoned graph and then train a GNN model that is based on the learned graph. Experimentally, they were able to show that Pro-GNN has a high node classification accuracy after perturbation attack.

4 Challenges

The majority of work in this project would focus on training a model to efficiently discriminate between benign (real) input and modified adversarial input. Our first experiment will be: using adversarial generation algorithms (i.e. NETTACK), we obtain an affected graph with the proper labels for affected and unaffected nodes. Next, we feed the altered graph through a graph neural network and train it to recognize the affected nodes. We will train the model to recognize altered nodes both with and without information about the adversarial algorithm being used. If the first experiment returns positive results, then we will perform joint training between attacked node detection and node classification to see if the accuracy of both tasks are improved. The final challenge for us to tackle would be identifying attacked nodes by running the algorithm only on a small subset of nodes in the graph. If there are multiple nodes with high probability to be attacked, then an algorithm can be used to identify the common ancestors of these nodes, which are likely to be the original attacked node.

References

Sun, Lichao et al. (2018). "Adversarial Attack and Defense on Graph Data: A Survey". In: pp. 1–18. arXiv: 1812.10528. URL: http://arxiv.org/abs/1812.10528.

Chen, Jinyin et al. (2019). "Time-aware Gradient Attack on Dynamic Network Link Prediction". In: pp. 1–9. arXiv: 1911.10561. URL: http://arxiv.org/abs/1911.10561.

Zügner, Daniel, Amir Akbarnejad, and Stephan Günnemann (2019). "Adversarial attacks on neural networks for graph data". In: *IJCAI International Joint Conference on Artificial Intelligence* 2019-Augus, pp. 6246–6250. ISSN: 10450823. DOI: 10.24963/ijcai.2019/872.

Jin, Wei et al. (2020). "Graph Structure Learning for Robust Graph Neural Networks". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 66–74. DOI: 10.1145/3394486.3403049. arXiv: 2005.10203.

Xu, X et al. (2020). "EDoG: Adversarial Edge Detection For Graph Neural Networks". In: