Time-Aware Neighbor Sampling on Temporal Graphs

Yiwei Wang[1] Yujun Cai[2] Yuxuan Liang[1] Henghui Ding[3] Changhu Wang[3] Bryan Hooi[1]
[1] *National University of Singapore* [2] *Nanyang Technological University* [3] *ByteDance*
wangyw_seu@foxmail.com

***Abstract*—We present a new neighbor sampling method on temporal graphs. In a temporal graph, predicting different nodes' time-varying properties can require the receptive neighborhood of various temporal scales. In this work, we propose the TNS (*Time-aware Neighbor Sampling*) method: TNS learns from temporal information to provide an adaptive receptive neighborhood for every node at any time. Learning how to sample neighbors is non-trivial, since the neighbor indices in time order are discrete and not differentiable. To address this challenge, we transform neighbor indices from discrete values to continuous ones by interpolating the neighbors' messages. TNS can be flexibly incorporated into popular temporal graph networks to improve their effectiveness without increasing their time complexity. TNS can be trained in an end-to-end manner. It requires no extra supervision and is automatically and implicitly guided to sample the neighbors that are most beneficial for prediction. Empirical results on multiple standard datasets show that TNS yields significant gains on edge prediction and node classification.**

***Index Terms*—temporal graph, neighboring sampling**

I. Introduction

Many real-world graphs are not static but evolving, e.g., edges can appear at any time [1]. Nodes may interact either due to gradual trends or fortuitous encounters. These graphs are called temporal (or dynamic) graphs [2]. Using static graph methods, e.g., GraphSAGE [3], to model temporal graphs is suboptimal since they cannot capture the evolutionary patterns. Recently, temporal graph networks (TGNs) [1], [2], [4] have been proposed to support learning on temporal graphs.

Advanced TGNs utilize a temporal graph aggregation module to obtain a target node's embedding [1], [2], which aggregate the messages from the target node's neighbors. The embedding is used to predict the target node's properties [5]. To prevent the number of neighbors from increasing without limitation as time flows, TGNs sample neighbors for the message aggregation (see Fig. 1), which improves their efficiency and stabilization [1]–[3]. Specifically, [3] and [2] utilize uniform neighbor sampling, that samples every neighbor with the same probability. [1] improves the neighbor sampling by incorporating temporal information, which samples the neighbors that interacted with the target node most recently.

Although sampling the most recent neighbors is generally more effective than uniform sampling [1], this may not be the only or the best option to utilize temporal information for neighbor sampling. First, sampling successive neighbors along the time axis can induce information redundancy. For example, a student can repeatedly interact with his classmates for several times in a short period because they are interested in a new computer game. Repeatedly considering such redundant interactions does not provide more useful information. Second, sampling the most recent neighbors limits the temporal scale of the receptive neighborhood [6], i.e., the neighborhood contributing to the target node's embedding. Effectively predicting a node's properties may require long-range dependencies. For example, a woman meets her fitness coach for exercising every Saturday. Sampling the most recent neighbors cannot capture her long-range exercising trends.

To address the above issues, we propose an expanded neighbor sampling approach, which inserts spacing between the sampled neighbors on the time axis. We define the expansion rate to control the spacing size. With the expansion rate as r, every sampled neighbor skips the next $r-1$ neighbors in time order (see Fig. 2). The inserted spacing reduces the redundancy between the sampled neighbors, while extending the temporal scale of the receptive neighborhood.

The above expanded neighbor sampling fixes a unified expansion rate for all the nodes at any time. However, this may be suboptimal, since predicting different nodes' time-varying properties can require various expansion rates. Hence, beyond the unified expansion rate, we aim to learn suitable and adaptive expansion rates from temporal information to offer appropriate receptive neighborhoods. Learning how to sample neighbors is non-trivial, since the neighbor indices in time order are discrete and not differentiable. In this work, we view the neighbors as image pixels and neighbors' messages as pixel values, and compare the neighbor sampling to the process of image rendering [7]. Concretely, we transfer the neighbor indices from discrete values to continuous ones by interpolating neighbors' messages, so that the neighbors' messages of any index in time order, even if not an integer, can be accessed. We encapsulate this idea into a new neighbor sampling method, called **TNS** (*Time-aware Neighbor Sampling*), which learns expansion rates with an expansion learning module. The learned expansion rates are then utilized to guide the neighbor sampling for message aggregation.

TNS can be incorporated into the popular TGN models to enhance their performance. It needs no extra supervision and can be trained in an end-to-end style. We analyze the back-propagation on TNS and find that TNS is automatically and implicitly guided to sample the neighbors that are most beneficial for prediction. In addition, theoretical analysis shows that using our TNS to improve the effectiveness of TGNs does

Should We Rely on Entity Mentions for Relation Extraction? Debiasing Relation Extraction with Counterfactual Analysis

Yiwei Wang[1] **Muhao Chen**[2] **Wenxuan Zhou** [2] **Yujun Cai**[3] **Yuxuan Liang**[1]
Dayiheng Liu[4] **Baosong Yang**[4] **Juncheng Liu**[1] **Bryan Hooi**[1]
[1] National University of Singapore [2] University of Southern California
[3] Nanyang Technological University [4]Alibaba Group
wangyw_seu@foxmail.com

Abstract

Recent literature focuses on utilizing the entity information in the sentence-level relation extraction (RE), but this risks leaking superficial and spurious clues of relations. As a result, RE still suffers from unintended **entity bias**, i.e., the spurious correlation between **entity mentions (names)** and relations. Entity bias can mislead the RE models to extract the relations that do not exist in the text. To combat this issue, some previous work masks the entity mentions to prevent the RE models from over-fitting entity mentions. However, this strategy degrades the RE performance because it loses the semantic information of entities. In this paper, we propose the CoRE (**Counterfactual Analysis based Relation Extraction**) debiasing method that guides the RE models to focus on the main effects of **textual context** without losing the entity information. We first construct a causal graph for RE, which models the dependencies between variables in RE models. Then, we propose to conduct counterfactual analysis on our causal graph to distill and mitigate the entity bias, that captures the causal effects of specific entity mentions in each instance. Note that our CoRE method is model-agnostic to debias existing RE systems during inference without changing their training processes. Extensive experimental results demonstrate that our CoRE yields significant gains on both effectiveness and generalization for RE. The source code is provided at: https://github.com/vanoracai/CoRE.

1 Introduction

Sentence-level relation extraction (RE) is an important step to obtain a structural perception of unstructured text (Distiawan et al., 2019) by extracting relations between **entity mentions (names)** from the **textual context**. From human oracle, textual context should be the main source of information that determines the ground-truth relations between entities. Consider a sentence "*Mary gave birth to Jerry.*"[1]. Even if we change the entity mentions from *'Jerry'* and *'Mary'* to other people's names, the relation 'parents' still holds between the subject and object as described by the textual context *"gave birth to"*.

Recently, some work aims to utilize entity mentions for RE (Yamada et al., 2020; Zhou and Chen, 2021), which, however, leak superficial and spurious clues about the relations (Zhang et al., 2018). In our work, we observe that entity information can lead to **biased** relation prediction by misleading RE models to extract relations that do not exist in the text. Fig. 1 visualizes a relation prediction from a state-of-the-art RE model (Alt et al., 2020) (see more examples in Tab. 7). Although the context describes no relation between the highlighted entity pair, the model extracts the relation as "*countries_of_residence*". Such an erroneous result can come from the spurious correlation between entity mentions and relations, or the **entity bias** in short. For example, if the model sees the relation "*countries_of_residence*" many more times than other relations when the object entity is *Switzerland* during training, the model can associate this relation with *Switzerland* during inference even though the relation does not exist in the text.

To combat this issue, some work (Zhang et al., 2017, 2018) proposes masking entities to prevent the RE models from over-fitting entity mentions. On the other hand, some other work (Peng et al., 2020; Zhou and Chen, 2021) finds that this strategy degrades the performance of RE because it loses the semantic information of entities.

For both machines and humans, RE requires a combined understanding of textual context and entity mentions (Peng et al., 2020). Humans can avoid the entity bias and make unbiased decisions by correctly referring to the textual context that describes the relation. The underlying mechanism is

[1]We use underline and wavy line to denote subject and object respectively by default.

GRAPHCACHE: Message Passing as Caching for Sentence-Level Relation Extraction

Yiwei Wang[1] **Muhao Chen**[2] **Wenxuan Zhou** [2] **Yujun Cai**[3]
Yuxuan Liang[1] **Bryan Hooi**[1]
[1] National University of Singapore [2] University of Southern California
[3] Nanyang Technological University
wangyw_seu@foxmail.com

Abstract

Entity types and **textual context** are essential properties for sentence-level relation extraction (RE). Existing work only encodes these properties within individual instances, which limits the performance of RE given the insufficient features in a single sentence. In contrast, we model these properties from the whole dataset and use the dataset-level information to enrich the semantics of every instance. We propose the GRAPHCACHE (**Graph Neural Network as Caching**) module, that propagates the features across sentences to learn better representations for RE. GRAPHCACHE aggregates the features from sentences in the whole dataset to learn **global** representations of properties, and use them to augment the **local** features within individual sentences. The global property features act as dataset-level prior knowledge for RE, and a complement to the sentence-level features. Inspired by the classical caching technique in computer systems, we develop GRAPHCACHE to update the property representations in an online manner. Overall, GRAPHCACHE yields significant effectiveness gains on RE and enables efficient message passing across all sentences in the dataset.

1 Introduction

Sentence-level relation extraction (RE) aims at identifying the relationship between two entities mentioned in a sentence. RE is crucial to the structural perception of human language, and also benefits many NLP applications such as automated knowledge base construction (Distiawan et al., 2019), event understanding (Wang et al., 2020a), discourse understanding (Yu et al., 2020), and question answering (Zhao et al., 2020). The modern tools of choice for RE are the large-scale pre-trained language models (PLMs) that are used to encode individual sentences, therefore obtaining the sentence-level representations (Liu et al., 2019; Joshi et al., 2020; Yamada et al., 2020).

Existing work considers **entity types** and **textual context** as essential **properties** for RE (Peng et al., 2020; Peters et al., 2019; Zhou and Chen, 2021). Nonetheless, most existing RE models only capture these properties *locally* within individual instances, while not *globally* modeling them from the whole dataset. Given the insufficient features of a single sentence, it is beneficial to model these properties from the whole dataset and use them to enrich the semantics of individual instances.

To overcome the aforementioned limitation, we propose to mine the entity and contextual information beyond individual instances so as to further improve the relation representations. Particularly, we first construct a heterogeneous graph to connect the instances sharing common properties for RE. This graph includes the sentences and *property caches*. Each cache represents a property of entity types or contextual topics. We connect every sentence to the corresponding property caches (see Fig. 1), and perform message passing over edges based on a graph neural network (GNN). In this way, the property caches aggregate the features from connected sentences, which will act as a complement to the sentence-level features and provide prior knowledge when identifying relations.

The constructed graph connecting sentences has the same scale as the whole dataset, which leads to high computational complexity of the GNN. To address this issue, our idea is to view the message passing of GNNs as data loading in computer systems, adapting the classical caching techniques to efficiently mining the property information from all sentences. We encapsulate this computational idea in a new GNN module, called GRAPHCACHE (**Graph Neural Network as Caching**), that uses an online updating strategy to refresh the property caches' representations. In addition, we design an attention-based global-local fusion module to augment the sentence-level representations using the property caches with adaptive weights.

Dangling-Aware Entity Alignment with Mixed High-Order Proximities

Juncheng Liu[1] **Zequn Sun**[2] **Bryan Hooi**[1] **Yiwei Wang**[1]
Dayiheng Liu[3] **Baosong Yang**[3] **Xiaokui Xiao**[1] **Muhao Chen**[4]
[1]National University of Singapore [2]Nanjing University
[3]Alibaba Group [4]University of Southern California
juncheng.liu@u.nus.edu, zqsun.nju@gmail.com
muhaoche@usc.edu

Abstract

We study dangling-aware entity alignment in knowledge graphs (KGs), which is an underexplored but important problem. As different KGs are naturally constructed by different sets of entities, a KG commonly contains some dangling entities that cannot find counterparts in other KGs. Therefore, dangling-aware entity alignment is more realistic than the conventional entity alignment where prior studies simply ignore dangling entities. We propose a framework using mixed high-order proximities on dangling-aware entity alignment. Our framework utilizes both the local high-order proximity in a nearest neighbor subgraph and the global high-order proximity in an embedding space for both dangling detection and entity alignment. Extensive experiments with two evaluation settings shows that our framework more precisely detects dangling entities, and better aligns matchable entities. Further investigations demonstrate that our framework can mitigate the hubness problem on dangling-aware entity alignment.

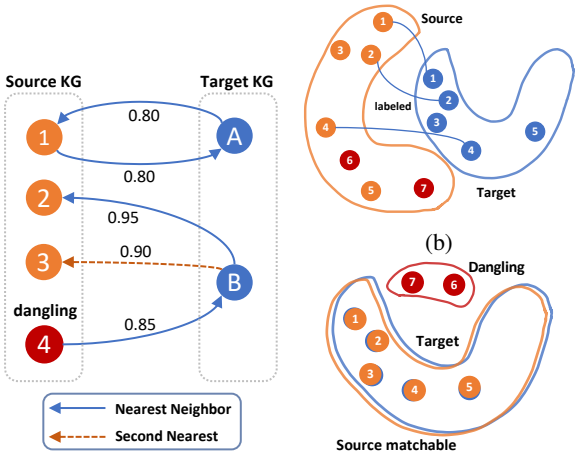

Figure 1: Toy examples for mixed high-order proximities. (a) Nearest neighbor (NN) subgraph where entities connect to NNs in the other KG using embedding similarities. 4 and its nearest neighbor B have 0.85 similarity. B prefers 2 and 3 with higher similarities. 1 and A are mutual nearest neighbors. (b) Labeled alignments and dangling entities. (c) Aligning matchable source and target distributions rather than only labeled alignments.

1 Introduction

Knowledge graphs (KGs) have become the backbone of many intelligent applications (Ji et al., 2021). In spite of their importance, many KGs are independently created without considering the interrelated and interchangeable nature of individually created knowledge (Chen et al., 2020). To allow complementary knowledge to be automatically combined and migrated across individual KGs, entity alignment seeks to identify equivalent entities in distinct KGs (Sun et al., 2020a). Recent literature has focused on learning embedding representations of multiple KGs where identical entities are aligned based on their embedding similarity (Chen et al., 2017; Cao et al., 2019; Fey et al., 2020; Sun et al., 2020a; Liu et al., 2021).

Aside from the surge of research effort on entity alignment (Zeng et al., 2021), an unresolved but important challenge that existing methods face is the *dangling entity* problem. Dangling entities are those unique entities in a KG that cannot find counterparts in another KG. Considering that individually created KGs are unlikely to share the same set of entities, identifying dangling entities is undoubtedly an indispensable step of any practical solution to entity alignment. However, nearly all prior studies have neglected dangling entities and assume there must be one-to-one entity mapping from the source KG to the target one (Sun et al., 2020c). This assumption prevents prior methods from practically supporting the alignment between KGs in real-world scenarios. To fill the gap, Sun et al. (2021) formally define a more practical problem setting where a model needs to both determine whether each given source entity is a matchable one, as well as retrieve counterparts for the predicted matchable entities.

A Unified 3D Human Motion Synthesis Model via Conditional Variational Auto-Encoder *

Yujun Cai[1], Yiwei Wang[2], Yiheng Zhu[6], Tat-Jen Cham[1], Jianfei Cai[3], Junsong Yuan[5], Jun Liu[7], Chuanxia Zheng[1], Sijie Yan[8], Henghui Ding[1], Xiaohui Shen[6], Ding Liu[6], Nadia Magnenat Thalmann[4]
[1]Nanyang Technological University, Singapore.
{yujun001,ding0093,chuanxia001}@e.ntu.edu.sg, astjcham@ntu.edu.sg
[2]National University of Singapore wangyw_seu@foxmail.com
[3]Monash University, Australia jianfei.cai@monash.edu, [4] University of Geneva Thalmann@miralab.ch
[5]State University of New York at Buffalo University, Buffalo, NY, USA jsyuang@buffalo.edu
[6]ByteDance Research {yiheng.zhu,shenxiaohui,liuding}@bytedance.com
[7]SUTD, Singapore jun_liu@sutd.edu.sg, [8]The Chinese University of Hong Kong yysijie@gmail.com

Abstract

We present a unified and flexible framework to address the generalized problem of 3D motion synthesis that covers the tasks of motion prediction, completion, interpolation, and spatial-temporal recovery. Since these tasks have different input constraints and various fidelity and diversity requirements, most existing approaches only cater to a specific task or use different architectures to address various tasks. Here we propose a unified framework based on Conditional Variational Auto-Encoder (CVAE), where we treat any arbitrary input as a masked motion series. Notably, by considering this problem as a conditional generation process, we estimate a parametric distribution of the missing regions based on the input conditions, from which to sample and synthesize the full motion series. To further allow the flexibility of manipulating the motion style of the generated series, we design an Action-Adaptive Modulation (AAM) to propagate the given semantic guidance through the whole sequence. We also introduce a cross-attention mechanism to exploit distant relations among decoder and encoder features for better realism and global consistency. We conducted extensive experiments on Human 3.6M and CMU-Mocap. The results show that our method produces coherent and realistic results for various motion synthesis tasks, with the synthesized motions distinctly adapted by the given action labels.

*This research is supported by Institute for Media Innovation, Nanyang Technological University (IMI-NTU) and the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. This research is also supported in part by Monash FIT Start-up Grant and SenseTime Gift Fund, National Science Foundation Grant CNS1951952 and SUTD project PIE-SGP-AI-2020-02.

1. Introduction

Generating realistic and plausible human body animation with specified actions has been a widely explored but challenging task in computer vision and graphics [29, 4]. To synthesize smooth and natural motions, traditional methods [30, 32] rely on the availability of complex pose specifications, which are time-consuming and expensive to obtain.

Recent deep learning approaches [5, 60, 56, 55, 17, 20, 61, 58] have investigated generating plausible human motions. However, since different motion synthesis tasks have different goals and expectations (as seen in Figure 1), many approaches are either restricted to one type of motion synthesis task or use different methods to address the various tasks. For example, much work [6, 14, 37, 63] is focused on the motion prediction task, typically adopting recurrent neural network (RNN) architectures to predict future frames sequentially, with new ones dependent only on previously generated frames. Although performing well in motion prediction, these approaches are not directly suited for generalizing to other motion synthesis tasks such as motion completion, interpolation, and spatial-temporal recovery, as shown in Figure 1, for which both forward and backward dependencies should be exploited. Moreover, many methods [5, 60, 56] are focused on minimizing the reconstruction error between the ground truth and generated motion sequences, while less considering motion diversity and human-likeness, which are also significant for realistic generation. Furthermore, in precise motion animation, it is

Modeling Trajectories with Neural Ordinary Differential Equations

Yuxuan Liang, Kun Ouyang, Hanshu Yan, Yiwei Wang, Zekun Tong, Roger Zimmermann
National University of Singapore, Singapore
{yuxliang,ouyangk,y-wang,rogerz}@comp.nus.edu.sg; {hanshu.yan,zekuntong}@u.nus.edu

Abstract

Recent advances in location-acquisition techniques have generated massive spatial trajectory data. Recurrent Neural Networks (RNNs) are modern tools for modeling such trajectory data. After revisiting RNN-based methods for trajectory modeling, we expose two common critical drawbacks in the existing uses. First, RNNs are discrete-time models that only update the hidden states upon the arrival of new observations, which makes them an awkward fit for learning real-world trajectories with continuous-time dynamics. Second, real-world trajectories are never perfectly accurate due to unexpected sensor noise. Most RNN-based approaches are deterministic and thereby vulnerable to such noise. To tackle these challenges, we devise a novel method entitled TrajODE for more natural modeling of trajectories. It combines the continuous-time characteristic of Neural Ordinary Differential Equations (ODE) with the robustness of stochastic latent spaces. Extensive experiments on the task of trajectory classification demonstrate the superiority of our framework against the RNN counterparts.

1 Introduction

A *spatial trajectory* is a sequence derived from a moving object in geographical spaces, formulated by a series of chronologically ordered points, i.e., $T = p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$. Each entry $p_i = (a_i, b_i, t_i)$ contains a set of geospatial coordinates (i.e., longitude a_i and latitude b_i) and a timestamp $t_i \in \mathbb{R}^+$. Modeling such trajectories allows us to analytically understand the moving objects and locations, facilitating a broad range of applications in smart transportation [Ruan *et al.*, 2020] and trip recommendation [Zhu *et al.*, 2017].

Recurrent Neural Networks (RNNs) are the modern tools for modeling spatial trajectories [Wu *et al.*, 2017]. They are powerful in learning sequences with variable lengths and significantly reduce human effort in trajectory feature engineering, compared to traditional models such as SVMs and Random Forests [Zheng *et al.*, 2008b]. Standard RNNs assume regular time intervals, while most trajectories are *irregularly-sampled* due to many reasons like communication loads, battery issues, and weather conditions [Zheng, 2015]. To tackle this irregularity, a simple trick is to concatenate the time interval information to the input of RNNs [Liu and Lee, 2017; Qin *et al.*, 2019]. As an alternative, GRU-D [Che *et al.*, 2018] used an exponential decay mechanism on the hidden state until the next observation is made. Moreover, [Zhu *et al.*, 2017; Che *et al.*, 2018; Liu *et al.*, 2019] enhanced RNNs with temporal gating mechanisms, where the time interval information is used to control the confidence of the input state.

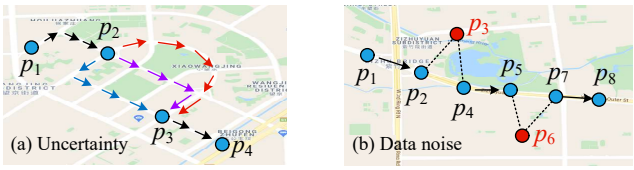

Figure 1: Illustration of uncertainty and data noise.

Though RNNs with the aforementioned heuristics can address the irregularity to some extent, they are still insufficient for modeling real-world trajectories. That is because the time interval between irregular samples in real data can be tens of seconds or even several minutes (e.g., see Figure 6 for the interval distributions in two datasets). A larger interval induces larger *uncertainty* between observations, especially for high-speed moving objects. For example, as shown in Figure 1(a), the GPS coordinates of a car are recorded every few minutes, leading to multiple possible paths between two consecutive points (e.g., between p_2 and p_3). Since the existing RNN approaches can only update their states upon the occurrence of a new point, they cannot adequately model such uncertainty, resulting in degenerated performances. To better match reality, we need a method that can inherently consider the underlying *continuous-time* dynamics of the trajectories.

Moreover, trajectory data are never perfectly accurate due to atmospheric conditions and signal blockage [Zheng, 2015]. Figure 1(b) shows a trajectory with noise. Sometimes, these errors significantly impact model accuracy. Most of the RNN models for trajectory modeling are deterministic and infeasible to defeat such noise. An intuitive idea is to perform noise filtering [Zheng, 2015] before training our models. However, it requires extra human efforts to carefully specify the distance threshold and may significantly reduce the number of points in trajectories. Therefore, how to enhance the model *robustness* against data noise remains a challenge.

Mixup for Node and Graph Classification

Yiwei Wang
National University of Singapore
Singapore
wangyw_seu@foxmail.com

Wei Wang
National University of Singapore
Singapore
wangwei@comp.nus.edu.sg

Yuxuan Liang
National University of Singapore
Singapore
yuxliang@outlook.com

Yujun Cai
Nanyang Technological University
Singapore
yujun001@e.ntu.edu.sg

Bryan Hooi
National University of Singapore
Singapore
bhooi@comp.nus.edu.sg

ABSTRACT

Mixup is an advanced data augmentation method for training neural network based image classifiers, which interpolates both features and labels of a pair of images to produce synthetic samples. However, devising the Mixup methods for graph learning is challenging due to the irregularity and connectivity of graph data. In this paper, we propose the Mixup methods for two fundamental tasks in graph learning: node and graph classification. To interpolate the irregular graph topology, we propose the two-branch graph convolution to mix the receptive field subgraphs for the paired nodes. Mixup on different node pairs can interfere with the mixed features for each other due to the connectivity between nodes. To block this interference, we propose the two-stage Mixup framework, which uses each node's neighbors' representations before Mixup for graph convolutions. For graph classification, we interpolate complex and diverse graphs in the semantic space. Qualitatively, our Mixup methods enable GNNs to learn more discriminative features and reduce over-fitting. Quantitative results show that our method yields consistent gains in terms of test accuracy and F1-micro scores on standard datasets, for both node and graph classification. Overall, our method effectively regularizes popular graph neural networks for better generalization without increasing their time complexity.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; **Neural networks**; *Regularization.*

KEYWORDS

data augmentation, node classification, graph classification

ACM Reference Format:
Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. 2021. Mixup for Node and Graph Classification. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449796

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449796

1 INTRODUCTION

Graph neural networks (GNNs) have achieved state-of-the-art performance on graph learning tasks, including node classification [27], [65], and graph classification [16], [60]. GNNs are capable of making predictions based on complex graph structures, thanks to their advanced representational power. However, the increased representational capacity comes with higher model complexity, which can induce over-fitting and weaken the generalization ability of GNNs. In this case, a trained GNN may capture random error or noise instead of the underlying data distribution [66], which is not what we expect.

To combat the over-fitting of neural networks, data augmentation has been demonstrated to be effective [38]. For node classification specifically, [40] proposes a data augmentation method named DropEdge. DropEdge follows the Vicinal Risk Minimization (VRM) principle [7] to define a vicinity around each node through randomly removing edges. Then, it draws additional virtual examples from the vicinity distribution to enlarge the support of the training distribution. In other words, it assumes that nodes have their class labels unchanged after the edge removals. However, whether this assumption holds is dataset-dependent and thus requires expert knowledge for usage. Furthermore, although DropEdge models the vicinity for the nodes sharing the same class, it does not describe the vicinity relation across samples of different classes.

Motivated by the above issues, we aim to design Mixup [67] methods for graph learning. Mixup is a recently proposed data augmentation method for image classification. Through linearly interpolating pixels of random image pairs and their training targets, Mixup generates synthetic images for training (see Fig. 1). Mixup does not need the ground-truth labels to be unchanged with the augmented features. In contrast, it incorporates the prior knowledge that interpolations of features should lead to interpolations of the associated targets [67]. Thus, Mixup extends the training distribution by constructing virtual training samples across all classes. From this vantage, Mixup acts as an effective regularization strategy for training image classifiers, which smoothens decision boundaries and improves the arrangements of hidden representations [52].

Although Mixup is effective in augmenting the image data, designing Mixup methods for graph learning is challenging. The challenges are rooted in the irregularity and connectivity of graph data. GNNs learn nodes' representations via the 'message passing' mechanism, which aggregates the representations between each node and its neighbors at each layer [58]. As a result, the

GraphAnoGAN: Detecting Anomalous Snapshots from Attributed Graphs

Siddharth Bhatia(✉)[1], Yiwei Wang[1], Bryan Hooi[1], and Tanmoy Chakraborty[2]

[1] National University of Singapore
{siddharth, y-wang, bhooi}@comp.nus.edu.sg
[2] IIIT-Delhi, India
tanmoy@iiitd.ac.in

Abstract. Finding anomalous snapshots from a graph has garnered huge attention recently. Existing studies address the problem using shallow learning mechanisms such as subspace selection, ego-network, or community analysis. These models do not take into account the multifaceted interactions between the structure and attributes in the network. In this paper, we propose GraphAnoGAN, an anomalous snapshot ranking framework, which consists of two core components – generative and discriminative models. Specifically, the generative model learns to approximate the distribution of anomalous samples from the candidate set of graph snapshots, and the discriminative model detects whether the sampled snapshot is from the ground-truth or not. Experiments on 4 real-world networks show that GraphAnoGAN outperforms 6 baselines with a significant margin (28.29% and 22.01% higher precision and recall, respectively compared to the best baseline, averaged across all datasets).

Keywords: Anomaly detection, graph snapshot, generative adversarial network

1 Introduction

Anomaly detection on graphs is a well-researched problem and plays a critical role in cybersecurity, especially network security [13]. Majority of the proposed approaches focus on anomalous nodes [27, 2, 24, 34], anomalous edges [43, 38, 18], community structures [44], or sudden surprising changes in graphs [10, 8, 14].

However, we focus our attention on ***detecting anomalous snapshots from attributed graphs***. This problem is motivated by the following cybersecurity threats: (a) fraudulent customers controlling the sentiment (customers operate in a way that they can not be tracked individually), (b) hackers targeting the network (attacks such as DDOS, phishing), (c) black-market syndicates in online social media [17], and (d) camouflaged financial transactions.

Detecting anomalous snapshots in a graph has received little attention; SpotLight [19] is one of them. However, SpotLight does not take into account the patterns being formed in the graph even if there is no outburst of edges. Moreover, it tends to ignore the node features as well. On the other hand, convolu-

Fine-Grained Urban Flow Prediction

Yuxuan Liang[1], Kun Ouyang[1], Junkai Sun[3], Yiwei Wang[1], Junbo Zhang[3,4], Yu Zheng[3,4,5], David S. Rosenblum[1,2], Roger Zimmermann[1]
[1]School of Computing, National University of Singapore, Singapore
[2]Department of Computer Science, George Mason University, VA, USA
[3]JD iCity, JD Technology, Beijing, China & JD Intelligent Cities Research, Beijing, China
[4]Artificial Intelligence Institute, Southwest Jiaotong University, Chengdu, China [5]Xidian University, Xi'an, China
{yuxliang,ouyangk,y-wang,rogerz,david}@comp.nus.edu.sg;{junkaisun,msjunbozhang,msyuzheng}@outlook.com

ABSTRACT

Urban flow prediction benefits smart cities in many aspects, such as traffic management and risk assessment. However, a critical prerequisite for these benefits is having fine-grained knowledge of the city. Thus, unlike previous works that are limited to coarse-grained data, we extend the horizon of urban flow prediction to fine granularity which raises specific challenges: 1) the predominance of inter-grid transitions observed in fine-grained data makes it more complicated to capture the spatial dependencies among grid cells at a global scale; 2) it is very challenging to learn the impact of external factors (e.g., weather) on a large number of grid cells separately. To address these two challenges, we present a Spatio-Temporal Relation Network (STRN) to predict fine-grained urban flows. First, a backbone network is used to learn high-level representations for each cell. Second, we present a Global Relation Module (GloNet) that captures global spatial dependencies much more efficiently compared to existing methods. Third, we design a Meta Learner that takes external factors and land functions (e.g., POI density) as inputs to produce meta knowledge and boost model performances. We conduct extensive experiments on two real-world datasets. The results show that STRN reduces the errors by 7.1% to 11.5% compared to the state-of-the-art method while using much fewer parameters. Moreover, a cloud-based system called UrbanFlow 3.0 has been deployed to show the practicality of our approach.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → *Artificial intelligence*; *Neural networks*.

KEYWORDS

Urban flow prediction; spatio-temporal data; relational learning; convolution neural networks; urban computing.

ACM Reference Format:
Yuxuan Liang, Kun Ouyang, Junkai Sun, Yiwei Wang, Junbo Zhang, Yu Zheng, David S. Rosenblum and Roger Zimmermann. 2021. Fine-Grained Urban Flow Prediction. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3442381.3449792

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449792

1 INTRODUCTION

Accurately forecasting urban flows, such as predicting the total crowd flows entering and leaving each location (i.e., grid cell) of a city during a given time interval [40, 41], plays an essential role in smart city efforts. It can provide insights to the government for decision making, risk assessment, and traffic management. For example, by foreseeing that an overwhelming crowd will stream into a region ahead of time, the government can carry out traffic control, send warnings or even evacuate people for public safety.

One key property that must be considered in grid-based urban flow prediction is *spatio-temporal (ST) dependencies*: the future of a grid cell is conditioned on its previous readings as well as neighbors' histories. Moreover, urban flows are also impacted by *external factors* such as weather conditions and events. For example, heavy snow can sharply reduce traffic flows in many regions. To address these characteristics, many existing studies [6, 20, 36, 40–42] use convolutional neural networks (CNNs) as the backbone structure to extract spatially near and distant dependencies; the temporal dependencies (e.g., at the recent, daily and weekly levels) are captured using different sub-branches. Meanwhile, the influence of external factors is encoded by some manually-designed subnetworks.

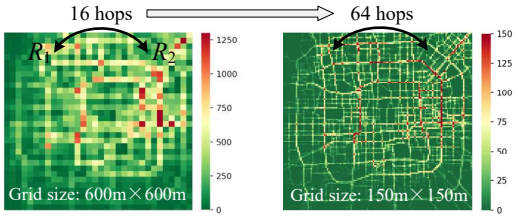

Figure 1: Coarse-grained vs. fine-grained urban flows.

In this paper, we focus on predicting urban flows at a *fine-grained* level, which is important yet unexplored in the community. Fine-grained flows can create exactness of the underlying dynamics of the city, encouraging better decision making. For instance, as shown in Figure 1, acquiring the traffic in a small area of interest with size 150m×150m can help allocate police resources more precisely while knowing that information at a district level with size 600m×600m is less useful. Notice that for a specific city, increasing the granularity (e.g., 600m→150m) is equivalent to obtaining higher resolution (e.g., 32×32→128×128). Thus, we use "high resolution" and "fine granularity" interchangeably. Although previous studies have shown promising results at coarse-grained levels (e.g., 32×32 Beijing [40]), their architectures are not suitable for predicting fine-grained urban flows due to the following specific challenges:

EIGNN: Efficient Infinite-Depth Graph Neural Networks

Juncheng Liu **Kenji Kawaguchi** **Bryan Hooi** **Yiwei Wang** **Xiaokui Xiao**
National University of Singapore
{juncheng,kenji,bhooi}@comp.nus.edu.sg
wangyw_seu@foxmail.com, xkxiao@nus.edu.sg

Abstract

Graph neural networks (GNNs) are widely used for modelling graph-structured data in numerous applications. However, with their inherently finite aggregation layers, existing GNN models may not be able to effectively capture long-range dependencies in the underlying graphs. Motivated by this limitation, we propose a GNN model with infinite depth, which we call Efficient Infinite-Depth Graph Neural Networks (EIGNN), to efficiently capture very long-range dependencies. We theoretically derive a closed-form solution of EIGNN which makes training an infinite-depth GNN model tractable. We then further show that we can achieve more efficient computation for training EIGNN by using eigendecomposition. The empirical results of comprehensive experiments on synthetic and real-world datasets show that EIGNN has a better ability to capture long-range dependencies than recent baselines, and consistently achieves state-of-the-art performance. Furthermore, we show that our model is also more robust against both noise and adversarial perturbations on node features.

1 Introduction

Graph-structured data are ubiquitous in the real world. To model and learn from such data, graph representation learning aims to produce meaningful node representations by simultaneously considering the graph topology and node attributes. It has attracted growing interest in recent years, as well as numerous real-world applications [32].

In particular, graph neural networks (GNNs) are a widely used approach for node, edge, and graph prediction tasks. Recently, many GNN models have been proposed (e.g., graph convolutional network [14], graph attention network [26], simple graph convolution [30]). Most modern GNN models follow a "message passing" scheme: they iteratively aggregate the hidden representations of every node with those of the adjacent nodes to generate new hidden representations, where each iteration is parameterized as a neural network layer with learnable weights.

Despite the success existing GNN models achieve on many different scenarios, they lack the ability to capture long-range dependencies. Specifically, for a predefined number of layers T, these models cannot capture dependencies with a range longer than T-hops away from any given node. A straightforward strategy to capture long-range dependencies is to stack a large number of GNN layers for receiving "messages" from distant nodes. However, existing work has observed poor empirical performance when stacking more than a few layers [16], which has been referred to as oversmoothing. This has been attributed to various reasons, including node representations becoming indistinguishable as depth increases. Besides oversmoothing, GNN models with numerous layers require excessive computational cost in practice since they need to repeatedly propagate representations across many layers. For these two reasons, simply stacking many layers for GNNs is not a suitable way to capture long-range dependencies.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), virtual.

CurGraph: Curriculum Learning for Graph Classification

Yiwei Wang
National University of Singapore
Singapore
wangyw_seu@foxmail.com

Wei Wang
National University of Singapore
Singapore
wangwei@comp.nus.edu.sg

Yuxuan Liang
National University of Singapore
Singapore
yuxliang@outlook.com

Yujun Cai
Nanyang Technological University
Singapore
yujun001@e.ntu.edu.sg

Bryan Hooi
National University of Singapore
Singapore
bhooi@comp.nus.edu.sg

ABSTRACT

Graph neural networks (GNNs) have achieved state-of-the-art performance on graph classification tasks. Existing work usually feeds graphs to GNNs in random order for training. However, graphs can vary greatly in their difficulty for classification, and we argue that GNNs can benefit from an easy-to-difficult curriculum, similar to the learning process of humans. Evaluating the difficulty of graphs is challenging due to the high irregularity of graph data. To address this issue, we present the **CurGraph** (Curriculum Learning for Graph Classification) framework, that analyzes the graph difficulty in the high-level semantic feature space. Specifically, we use the infomax method to obtain graph-level embeddings and a neural density estimator to model the embedding distributions. Then we calculate the difficulty scores of graphs based on the intra-class and inter-class distributions of their embeddings. Given the difficulty scores, CurGraph first exposes a GNN to easy graphs, before gradually moving on to hard ones. To provide a soft transition from easy to hard, we propose a smooth-step method, which utilizes a time-variant smooth function to filter out hard graphs. Thanks to CurGraph, a GNN learns from the graphs at the border of its capability, neither too easy or too hard, to gradually expand its border at each training step. Empirically, CurGraph yields significant gains for popular GNN models on graph classification and enables them to achieve superior performance on miscellaneous graphs.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; *Learning from implicit feedback*; *Batch learning*; *Learning latent representations*; *Neural networks.*

KEYWORDS

graph classification, curriculum learning, graph neural networks

ACM Reference Format:
Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. 2021. CurGraph: Curriculum Learning for Graph Classification. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3442381.3450025

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450025

1 INTRODUCTION

Graph classification is a fundamental task on graph data, which aims to predict the class labels of entire graphs. The modern tools of choice for this task are graph neural networks (GNNs). Typically, GNNs build node representations from node features and graph topology via the 'message passing' mechanism and then make graph-level predictions by summarizing the node representations through a readout function [63], [49].

Although a lot of attention has been paid to developing new GNN architectures of higher representational capacity [30], [63], [65], it is also valuable to explore how to design advanced training methods to improve GNNs. Most existing work performs training of GNNs in a straightforward manner, i.e., all graphs are treated equally and presented in random order during training. However, even in the same dataset, graphs can vary significantly in their difficulty levels. For example, some graphs are easy to discriminate by their significant and popular substructures, while others require sophisticated reasoning due to their complicated topology and indistinct patterns (see Fig. 2). Extensive research discovers that feeding the training samples in a meaningful order, starting from easy ones and gradually taking more difficult ones, can benefit machine learning algorithms [4], [51], [62]. This strategy is known as Curriculum Learning.

Curriculum Learning was first formally proposed in [4], inspired by humans' learning process: an infant starts with a simple initial state, and then builds on that to handle more and more sophisticated concepts gradually. Recent years have witnessed successful applications of Curriculum Learning in the fields of Computer Vision [24], [21], [42] and Natural Language Processing [41], [38]. In terms of optimization, Curriculum Learning excludes the negative impacts from difficult or even noisy samples in the early training stages, and guides the model towards better local minima in the parameter space. Motivated by this, we argue that GNNs can benefit from Curriculum Learning on graph classification, which, however, remains under-explored.

A key challenge of designing a Curriculum Learning method for graph classification lies in how to evaluate the difficulty of graphs. The evaluation is non-trivial because the graph data is highly irregular and noisy. Each graph exhibits complicated relationships between nodes, and the number of nodes (#Nodes) and

Adaptive Data Augmentation on Temporal Graphs

Yiwei Wang[1] **Yujun Cai**[2] **Yuxuan Liang**[1] **Henghui Ding**[3]
Changhu Wang[3] **Siddharth Bhatia**[1] **Bryan Hooi**[1]
[1] National University of Singapore
[2] Nanyang Technological University
[3] ByteDance
wangyw_seu@foxmail.com, {yujun001,ding0093}@e.ntu.edu.sg,
yuxliang@outlook.com, changhu.wang@gmail.com,
{siddharth,bhooi}@comp.nus.edu.sg

Abstract

Temporal Graph Networks (TGNs) are powerful on modeling temporal graph data based on their increased complexity. Higher complexity carries with it a higher risk of overfitting, which makes TGNs capture random noise instead of essential semantic information. To address this issue, our idea is to transform the temporal graphs using data augmentation (DA) with adaptive magnitudes, so as to effectively augment the input features and preserve the essential semantic information. Based on this idea, we present the **MeTA** (*Memory Tower Augmentation*) module: a multi-level module that processes the augmented graphs of different magnitudes on separate levels, and performs message passing across levels to provide adaptively augmented inputs for every prediction. MeTA can be flexibly applied to the training of popular TGNs to improve their effectiveness without increasing their time complexity. To complement MeTA, we propose three DA strategies to realistically model noise by modifying both the temporal and topological features. Empirical results on standard datasets show that MeTA yields significant gains for the popular TGN models on edge prediction and node classification in an efficient manner.

1 Introduction

Many real-world graphs are not static but evolving, where every edge (or interaction) has a timestamp to denote its occurrence time. These graphs are called temporal (or dynamic) graphs [39]. Recently, temporal graph networks (TGNs) [25, 39, 16] have been proposed to support learning on temporal graphs. Advanced TGNs utilize an RNN based memory module to represent a node's history as a compact state (see Fig. 1), which is used to predict the node's activities [28]. TGNs are capable of making predictions from complex graph topology and temporal information , thanks to their advanced representational power. However, the increased representational capacity comes with higher model complexity, which can induce over-fitting and weaken their generalization ability. In particular, a trained TGN may capture random noise instead of semantic information, which is not desired [41].

To combat over-fitting, data augmentation (DA) has been demonstrated to be effective [23]. Nevertheless, DA for the temporal graphs remains under-explored, of which the main challenges lie in highly irregular dynamic topology. DA applies transformations to input features so as to model realistic noise for enriching the input data. The magnitudes of the transformations, known as *DA magnitudes*, are controlled by hyper-parameters, which is positively related to the difference between the input features before and after DA [5]. Existing work on image and text data devises *adaptive DA* methods, which apply higher DA magnitudes to the less informative parts of input features, in order to effectively augment the input features while preserving the essential semantic information [38].

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Provably Robust Node Classification via Low-Pass Message Passing

Yiwei Wang*, Shenghua Liu†, Minji Yoon‡, Hemank Lamba‡, Wei Wang*, Christos Faloutsos‡, Bryan Hooi*
*School of Computing, National University of Singapore, Singapore
Email: {y-wang,wangwei,bhooi}@comp.nus.edu.sg
†Institute of Computing Technology, Chinese Academy of Sciences, China
Email: liushenghua@ict.ac.cn
‡Carnegie Mellon University, United States
Email: {minjiy,hlamba,christos}@cs.cmu.edu

***Abstract*—Graph Convolutional Networks (GCNs) have achieved state-of-the-art performance on node classification. However, recent works have shown that GCNs are vulnerable to adversarial attacks, such as additions or deletions of adversarially-chosen edges in the graph, in order to mislead the node classification algorithms. How can we design robust GCNs that are resistant to such adversarial attacks? More challengingly, how can we do this in a way that is provably robust? We propose a robust node classification approach based on a low-pass 'message passing' mechanism, that (a) reduces the effectiveness of adversarial attacks in experiments, and (b) provides theoretical guarantees against adversarial attacks. Our approach can be embedded into the existing GCN architectures to enhance their robustness. Empirical results show that our loss-pass method effectively improves the performance of multiple GCNs under miscellaneous perturbations and helps them to achieve superior performance on various graphs.**

***Keywords*-Graph Convolutional Networks, Node Classification, Robustness**

I. INTRODUCTION

Node classification is a fundamental task on graph data, which is to classify the nodes in an (attributed) graph [1]: for example, predicting protein types in a protein interaction graph [2]. In recent years, graph neural network methods such as Graph Convolutional Networks (GCNs) have achieved state-of-the-art performance for node classification [3]–[5], leading to tremendous interest.

Despite their effectiveness for node classification, GCNs have been found to be vulnerable to adversarial attacks [6]. This is a significant drawback in terms of their reliability in many real-world settings, particularly in risk-sensitive scenarios such as security, healthcare, and finance [7]. It is thus essential to design GCNs that are robust against adversarial attacks.

GCNs apply a 'message passing' mechanism to make predictions, whereby they aggregate semantic representations of each node and its neighbors at each layer. On a clean graph structure, this message passing process tends to produce the similar predictions to the connected nodes [8]. However, when adversarial edges are present, it incurs over-whelming erroneous information and misleads the predictions heavily.

In this paper, we propose a **low-pass 'message passing'** mechanism that provides higher robustness for GCNs without the loss of effectiveness. The mechanism weakens 'message passing' between semantically dissimilar nodes. We design this mechanism based on the observation that node pairs which are very semantically different are the most susceptible to being attacked by the adversary, as perturbations across such pairs have a stronger effect on the prediction results. Therefore, by weakening the 'message passing' between dissimilar nodes, we inhibit the prediction deviations induced by adversarial attacks. We weaken the 'message passing' strength of an adversarial edge to the extent that is positively related to the distance between the semantic representations of the nodes on its ends, because the more different the nodes linked by an adversarial edge are, the more deviations on the predictions are induced by the edge (see Eq. (2)).

Recent research in adversarial machine learning has seen a rapid back-and-forth between adversarial attacks and defenses, in which several advanced defense approaches were broken by newly proposed attack methods [9]. Such 'arms races' leave machine learning systems vulnerable to attack. To circumvent this problem, our solution is to provide *theoretical guarantees* for the robust node classification beyond the intuitive justification, whereby the model output does not excessively change when faced with adversarial perturbations. In particular, we focus on defending against structural perturbations, i.e., adding/deleting edges, which fit well for the characteristics of graph data.

To provide the provable robustness guarantees, a challenge is that the complex non-linear activation functions in GCNs make the relations between inputs and outputs hard to analyze. The irregular and noisy graph structure aggravates this challenge. Another challenge lies in the 'message passing' of GCNs, with which the predictions of widespread nodes would be affected by even a small number of adversarially added/deleted edges. Therefore, we have to consider widespread nodes and edges for analyzing the prediction of a node. To address these challenges, we

Shenghua Liu is also with CAS Key Laboratory of Network Data Science & Technology, CAS, and University of Chinese Academy of Sciences, Beijing 100049, China.

Revisiting Convolutional Neural Networks for Citywide Crowd Flow Analytics

Yuxuan Liang[1] ✉, Kun Ouyang[1], Yiwei Wang[1], Ye Liu[1], Junbo Zhang[2,3,4], Yu Zheng[2,3,4], David S. Rosenblum[1]

[1] School of Computing, National University of Singapore, Singapore
[2] JD Intelligent Cities Research & JD Intelligent Cities Business Unit, Beijing, China
[3] Institute of Artificial Intelligence, Southwest Jiaotong University, China
[4] Xidian University, Xian, China
{yuxliang,msjunbozhang,msyuzheng}@outlook.com
{ouyangk,y-wang,liuye,david}@comp.nus.edu.sg

Abstract. Citywide crowd flow analytics is of great importance to smart city efforts. It aims to model the crowd flow (e.g., inflow and outflow) of each region in a city based on historical observations. Nowadays, Convolutional Neural Networks (CNNs) have been widely adopted in raster-based crowd flow analytics by virtue of their capability in capturing spatial dependencies. After revisiting CNN-based methods for different analytics tasks, we expose two common critical drawbacks in the existing uses: 1) inefficiency in learning global spatial dependencies, and 2) overlooking latent region functions. To tackle these challenges, in this paper we present a novel framework entitled DeepLGR that can be easily generalized to address various citywide crowd flow analytics problems. This framework consists of three parts: 1) a local feature extraction module to learn representations for each region; 2) a global context module to extract global contextual priors and upsample them to generate the global features; and 3) a region-specific predictor based on tensor decomposition to provide customized predictions for each region, which is very parameter-efficient compared to previous methods. Extensive experiments on two typical crowd flow analytics tasks demonstrate the effectiveness, stability, and generality of our framework.

1 Introduction

Citywide crowd flow analytics is very critical to smart city efforts around the world. A typical task is citywide crowd flow prediction [21,20,12], which aims to predict the traffic (e.g., inflows and outflows of every region) for the next time slot, given the historical traffic observations. It can help the governors conduct traffic control and avoid potential catastrophic stampede before a special event. Another important task is to infer the fine-grained crowd flows from available coarse-grained data sources, which can reduce the expense of urban systems [11,13]. Other tasks [19,24] are also actively studied by the community due to the vital impact of citywide crowd flow analytics.

Progressive Supervision for Node Classification

Yiwei Wang[1] (✉), Wei Wang[1], Yuxuan Liang[1], Yujun Cai[2], and Bryan Hooi[1]

[1] School of Computing, National University of Singapore, Singapore
{y-wang,wangwei,yuxliang,bhooi}@comp.nus.edu.sg
[2] Nanyang Technological University, Singapore
yujun001@e.ntu.edu.sg

Abstract. Graph Convolution Networks (GCNs) are a powerful approach for the task of node classification, in which GCNs are trained by minimizing the loss over the final-layer predictions. However, a limitation of this training scheme is that it enforces every node to be classified from the fixed and unified size of receptive fields, which may not be optimal. We propose ProSup (Progressive Supervision), that improves the effectiveness of GCNs by training them in a different way. ProSup supervises all layers progressively to guide their representations towards the characteristics we desire. In addition, we propose a novel technique to reweight the node-wise losses, so as to guide GCNs to pay more attention to the nodes that are hard to classify. The hardness is evaluated progressively following the direction of information flows. Finally, ProSup fuses the rich hierarchical activations from multiple scales to form the final prediction in an adaptive and learnable way. We show that ProSup is effective to enhance the popular GCNs and help them to achieve superior performance on miscellaneous graphs.

Keywords: Graph Convolutional Networks · Progressive Supervision · Node Classification

1 Introduction

Node classification is a fundamental task on graph data, which aims to classify the nodes in an (attributed) graph [14]. For this task, Graph Convolutional Networks (GCNs) have achieved state-of-the-art performance [23]. Typically, GCNs follow a multi-layer structure (see Fig. 1(a)). Across layers, GCNs update node representations via the 'message-passing' mechanism, i.e., they aggregate the representations of each node and its neighbors to produce new ones at the next layer. Denote the subgraph contributing to a node's representation as its receptive field. From bottom to top, the receptive field expands gradually, which is generally a node's l-hop neighborhood at the lth layer [19].

For training GCNs, it is common to minimize the classification loss on the final-layer predictions. This training scheme is convenient, but not necessarily ideal for effectiveness. One limitation is that it enforces GCNs to classify all nodes from the unified size of receptive fields, but nodes can have diverse 'appropriate' receptive fields for classification [24]. In a social network, for example,

NodeAug: Semi-Supervised Node Classification with Data Augmentation

Yiwei Wang
National University of Singapore
Singapore
wangyw_seu@foxmail.com

Wei Wang
National University of Singapore
Singapore
wangwei@comp.nus.edu.sg

Yuxuan Liang
National University of Singapore
Singapore
yuxliang@outlook.com

Yujun Cai
Nanyang Technological University
Singapore
yujun001@e.ntu.edu.sg

Juncheng Liu
National University of Singapore
Singapore
juncheng.liu@u.nus.edu

Bryan Hooi
National University of Singapore
Singapore
bhooi@comp.nus.edu.sg

ABSTRACT

By using Data Augmentation (DA), we present a new method to enhance Graph Convolutional Networks (GCNs), that are the state-of-the-art models for semi-supervised node classification. DA for graph data remains under-explored. Due to the connections built by edges, DA for different nodes influence each other and lead to undesired results, such as uncontrollable DA magnitudes and changes of ground-truth labels. To address this issue, we present the **NodeAug** (Node-Parallel Augmentation) scheme, that creates a 'parallel universe' for each node to conduct DA, to block the undesired effects from other nodes. NodeAug regularizes the model prediction of every node (including unlabeled) to be invariant with respect to changes induced by Data Augmentation (DA), so as to improve the effectiveness. To augment the input features from different aspects, we propose three DA strategies by modifying both node attributes and the graph structure. In addition, we introduce the subgraph mini-batch training for the efficient implementation of NodeAug. The approach takes the subgraph corresponding to the receptive fields of a batch of nodes as the input per iteration, rather than the whole graph that the prior full-batch training takes. Empirically, NodeAug yields significant gains for strong GCN models on the Cora, Citeseer, Pubmed, and two co-authorship networks, with a more efficient training process thanks to the proposed subgraph mini-batch training approach.

CCS CONCEPTS

• **Computing methodologies** → **Semi-supervised learning settings**; **Regularization**; *Neural networks.*

KEYWORDS

graph convolutional networks, data augmentation, graph mining, semi-supervised learning

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403063

ACM Reference Format:
Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, Juncheng Liu, and Bryan Hooi. 2020. NodeAug: Semi-Supervised Node Classification with Data Augmentation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403063

1 INTRODUCTION

Semi-Supervised node classification is a fundamental task on graph data, which aims to classify the nodes in an (attributed) graph given the class labels of a few nodes [19]. For this task, Graph Convolutional Networks (GCNs) have achieved state-of-the-art performance [30]. Typically, GCNs predict the classes via the 'message-passing' mechanism, i.e., they aggregate the semantic representations between each node and its neighbors at each layer to generate the final-layer predictions. Thus, the prediction of a node relies on the attributes of other nodes and the graph structure in addition to its own attributes, all of which act as its feature.

In the general semi-supervised learning (SSL) problem, effectively using unlabelled data is essential [21], since labeled samples are scarce while unlabeled data are typically present in massive quantity and easier to obtain. In contrast, GCNs are trained only over the predictions of the labeled nodes by minimizing the supervised classification loss, but the predictions of the unlabeled nodes do not contribute to the training. This leads to the question: how can we effectively incorporate unlabeled data into GCN models?

Hence, in this work, we regularize the predictions of all nodes to be invariant with respect to the changes induced by Data Augmentation (DA), through minimizing the classification divergence between the original nodes and the augmented ones, a.k.a., consistency training [31]. Intuitively, DA makes changes to the input data in ways that should have relatively trivial effects on the final node classification, based on our human knowledge, but diverse effects to the input features. If we enforce the model predictions to be invariant with respect to these changes, we embed human knowledge on the labels of especially the unlabeled nodes into GCN models, which potentially yields the performance gains. However, to date, the DA techniques for graph data remains under-explored.

Data Augmentation has been applied successfully to the tasks on image data, e.g., image classification [31]. A significant difference between graph and image data is that nodes are connected, while

Learning Progressive Joint Propagation for Human Motion Prediction

Yujun Cai[1], Lin Huang[3], Yiwei Wang[5], Tat-Jen Cham[1], Jianfei Cai[1,2], Junsong Yuan[3], Jun Liu[6], Xu Yang[1], Yiheng Zhu[4], Xiaohui Shen[4], Ding Liu[4], Jing Liu[4], and Nadia Magnenat Thalmann[1]

[1] Nanyang Technological University, Singapore.
{yujun001,s170018}@e.ntu.edu.sg,{astjcham,nadiathalmann}@ntu.edu.sg
[2] Monash University, Australia jianfei.cai@monash.edu
[3] State University of New York at Buffalo University, USA
{lhuang27,jsyuan}@buffalo.edu
[4] ByteDance Research
{shenxiaohui,yiheng.zhu,liuding,jing.liu}@ bytedance.com
[5] National University of Singapore wangyw_seu@foxmail.com
[6] SUTD, Singapore jun_liu@sutd.edu.sg

Abstract. Despite the great progress in human motion prediction, it remains a challenging task due to the complicated structural dynamics of human behaviors. In this paper, we address this problem in three aspects. First, to capture the long-range spatial correlations and temporal dependencies, we apply a transformer-based architecture with the global attention mechanism. Specifically, we feed the network with the sequential joints encoded with the temporal information for spatial and temporal explorations. Second, to further exploit the inherent kinematic chains for better 3D structures, we apply a progressive-decoding strategy, which performs in a central-to-peripheral extension according to the structural connectivity. Last, in order to incorporate a general motion space for high-quality prediction, we build a memory-based dictionary, which aims to preserve the global motion patterns in training data to guide the predictions. We evaluate the proposed method on two challenging benchmark datasets (Human3.6M and CMU-Mocap). Experimental results show our superior performance compared with the state-of-the-art approaches.

Keywords: 3D motion prediction, transformer network, progressive decoding, dictionary module

1 Introduction

Human motion prediction aims to forecast a sequence of future dynamics based on an observed series of human poses. It has extensive applications in robotics, computer graphics, healthcare and public safety [20, 24, 26, 41, 40], such as human robot interaction [25], autonomous driving [35] and human tracking [18].

Detecting Implementation Bugs in Graph Convolutional Network based Node Classifiers

Yiwei Wang*, Wei Wang*, Yujun Cai†, Bryan Hooi*, Beng Chin Ooi*
*School of Computing, National University of Singapore, Singapore
Email: {y-wang,wangwei,bhooi,ooibc}@comp.nus.edu.sg
†Nanyang Technological University, Singapore
Email: yujun001@e.ntu.edu.sg

***Abstract*—Graph convolutional networks (GCNs) have achieved state-of-the-art performance on the task of node classification. However, the performance of GCNs is prone to implementation bugs that do not explicitly produce compile-time or run-time errors but degrade their effectiveness heavily. These bugs are hard to detect, since the way in which the node attributes and graph structures contribute to the outputs is complicated, non-transparent, and not traceable by humans. To address this issue, we propose a systematic approach with formal justifications to detect implementation bugs in GCN based node classifiers. Our approach is based on the idea of Metamorphic Testing, which does not check input-output relations for a single input, but for input-output pairs. To speed up our approach, we design a pipeline system, which synchronizes the workload on CPUs and GPUs adaptively and processes them simultaneously. Our empirical study shows that our approach is able to identify over 80% of the synthetic mutants and two real-world bugs in GCN implementations. In addition, our pipeline system can achieve more than 10× speedup over the sequential system that leaves the CPU/GPU idle when using the other.**

***Keywords*-Graph Convolutional Networks, Metamorphic Testing, Implementation Bugs, Node Classification**

I. Introduction

Node classification refers to the problem of classifying nodes (such as documents) in a graph (such as a citation network), where labels are only available for a subset of nodes. It is a fundamental task for evaluating the machine learning models on graphs, and meanwhile supports miscellaneous practical applications, e.g., learning molecular fingerprints [1] and predicting entity properties [2]. On this task, graph convolutional networks (GCNs) have made breakthrough advancements and become the default solution due to its state-of-the-art effectiveness [3], for which a comprehensive survey can be found in [4].

When a GCN based node classifier fails, for example in Fig. 1, people may train the model with more data since they believe that more training data can improve its performance [5]. However, this does not take effect if the implementation is wrong. To address this issue, in this paper, we focus on detecting implementation bugs of GCNs, which do not explicitly induce compile-time or run-time errors but heavily degrade their performance, in an effective and efficient manner.

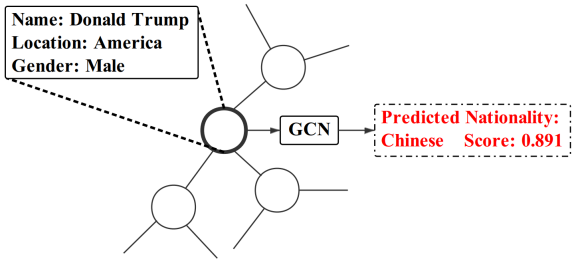

Figure 1: *A GCN based node classifier makes a wrong prediction on the entity property.* A possible fundamental cause is an implementation bug, which needs to be detected first. But due to the poor explainability of GCNs, it is hard to determine whether the error arises from bugs, or from other problems, such as insufficient training data.

To verify software, the traditional techniques build 'test cases' which contain '$input$'-'$output$' pairs. They supply the '$input$' to the program under test (PUT) and check whether the output matches their expectations. However, this mechanism is generally infeasible for GCNs, because finding one (or a few) instances that do not meet the expectations, such as the case in Fig. 1, does not necessarily indicate the presence of an implementation bug. In principle, GCNs are a form of neural network, which classifies different nodes by transforming the node attributes and graph structures via a multi-layer nonlinear function. This function is mathematically complicated, and how inputs contribute to the outputs is not traceable by humans. Many causes, e.g., adversarial examples and deficient data, can account for the failure as in Fig. 1; it is difficult to determine which is true, because GCNs lack complete interpretability and theoretical support [6].

Although significant effort has been devoted to developing GCN models, the approaches for testing them are under-explored [7]. In this field, we argue that Metamorphic Testing (MT) [8] is useful, since it does not need the correct

Optimization Algorithms for Graph Laplacian Estimation via ADMM and MM

Licheng Zhao, Yiwei Wang, Sandeep Kumar, and Daniel P. Palomar, *Fellow, IEEE*

***Abstract*—In this paper, we study the graph Laplacian estimation problem under a given connectivity topology. We aim at enriching the unified graph learning framework proposed by Egilmez *et al.* and improve the optimality performance of the combinatorial graph Laplacian (CGL) case. We apply the well-known alternating direction method of multipliers (ADMM) and majorization–minimization (MM) algorithmic frameworks and propose two algorithms, namely, GLE-ADMM and GLE-MM, for graph Laplacian estimation. Both algorithms can achieve an optimality gap as low as 10^{-4}, around three orders of magnitude more accurate than the benchmark. In addition, we find that GLE-ADMM is more computationally efficient in a dense topology (e.g., an almost complete graph), while GLE-MM is more suitable for sparse graphs (e.g., trees). Furthermore, we consider exploiting the leading eigenvectors of the sample covariance matrix as a nominal eigensubspace and propose a third algorithm, named GLENE, which is also based on ADMM. Numerical experiments show that the inclusion of a nominal eigensubspace significantly improves the estimation of the graph Laplacian, which is more evident when the sample size is smaller than or comparable to the problem dimension.**

***Index Terms*—Graph learning, Laplacian estimation, nominal eigensubspace, ADMM, Majorization-Minimization.**

I. Introduction

GRAPH signal processing has been a rapidly developing field in recent years, with a wide range of applications such as social, energy, transportation, sensor, and neuronal networks [2]. Its popularity results from the revolutionary way it models data points and their pairwise interconnections. When a collection of data samples are modeled as a graph signal, each sample is treated as a vertex and their pairwise interconnections are represented by a number of edges. Every edge is associated with a weight, and the weight value often reflects the similarity between the connecting vertices. We define a weighted graph as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, where $\mathcal{V}$ denotes the vertex set with $\text{card}(\mathcal{V}) = N$ (N vertices), $\mathcal{E}$ denotes the edge set with $\text{card}(\mathcal{E}) = M$ (M edges), and $\mathbf{W} \in \mathbb{R}^{N\times N}$ is the weight matrix. We will focus on a specific type of graph which is undirected and connected (i.e., one connected component only) with no self-loops, so the corresponding weight matrix is symmetric and elementwisely non-negative, with its diagonal elements all being zero. The graph Laplacian, also known as a combinatorial graph Laplacian (see [1, Definition 2]), is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \in \mathbb{R}^{N\times N}, \tag{1}$$

where $\mathbf{D}$ is the degree matrix, which is diagonal in structure with $D_{ii} = \sum_{j=1}^{N} W_{ij}$. The adjacency matrix $\mathbf{A}$ is defined as

$$\mathbf{A} = \text{sgn}(\mathbf{W}) \in \mathbb{R}^{N\times N}, \tag{2}$$

which implies $A_{ij} = 1$ if $W_{ij} > 0$, $A_{ij} = 0$ if $W_{ij} = 0$, and $A_{ii} = 0$.

In most practical scenarios, it is straightforward to derive the vertex set, but the edge set and the associated weight matrix are not readily available. This is either because no reasonable initial graph exists, or only a vague prior is given [3]. Under these circumstances, it is of great significance to learn the graph structure through statistical methods from the available finite data samples. In this paper, we specifically assume the data samples are drawn from a Gaussian Markov Random Field (GMRF) [4]. GMRFs are powerful tools and can be applied to such areas as structural time-series analysis (e.g., autoregressive models), graphical models, semiparametric regression and splines, image analysis, and spatial statistics [4]. The graph structure estimation of a GMRF model naturally amounts to the estimation of the precision matrix (inverse covariance matrix) by means of maximum likelihood estimation. As it is pointed out in the literature, the precision matrix is popularly structured as a graph Laplacian [5], [6] and the corresponding GMRF models are named Laplacian GMRF models. A graph Laplacian is a positive semidefinite (PSD) matrix with non-positive off-diagonal entries and a zero row-sum [7]:

$$\mathcal{L} = \left\{\mathbf{L} \succeq \mathbf{0} \middle| \mathbf{L}\mathbf{1} = \mathbf{0},\ L_{ij} \leq 0,\ i \neq j\right\}, \tag{3}$$

which always corresponds to a graph with non-negative weighted edges [6]. As is mentioned in [6], the significance of the Laplacian GMRF model has been recognized in image reconstruction [8], image segmentation [9], and texture modeling and discrimination [10], [11]. With the aforementioned definitions for $\mathbf{L}$ and $\mathbf{A}$, we can describe the constraint set for graph Laplacians under a given connectivity topology:

$$\mathcal{L}(\mathbf{A}) = \left\{\mathbf{\Theta} \succeq \mathbf{0} \middle| \mathbf{\Theta}\mathbf{1} = \mathbf{0},\ \begin{array}{l}\Theta_{ij} \leq 0 \text{ if } A_{ij} = 1\\ \Theta_{ij} = 0 \text{ if } A_{ij} = 0\end{array} \text{ for } i \neq j\right\}, \tag{4}$$

which is a subset of $\mathcal{L}$. The graph Laplacian notation is changed to $\mathbf{\Theta}$ so as to align with the majority of the existing works.

Manuscript received April 23, 2018; revised October 23, 2018, January 27, 2019, May 3, 2019, and May 29, 2019; accepted June 13, 2019. Date of publication June 27, 2019; date of current version July 23, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pierre Borgnat. This work was supported by the Hong Kong RGC 16208917 research grant. *(Corresponding author: Yiwei Wang.)*

The authors are with the Hong Kong University of Science and Technology, Hong Kong (e-mail: lzhaoai@ust.hk; ywanggp@ust.hk; eesandeep@ust.hk; palomar@ust.hk).

Digital Object Identifier 10.1109/TSP.2019.2925602

Using Knowledge Graphs to Explain Entity Co-occurrence in Twitter

Yiwei Wang*
Hong Kong University of Science and Technology
Hong Kong, China
wangyw_seu@foxmail.com

Mark James Carman
Monash University
Caulfield, VIC, Australia
mark.carman@monash.edu

Yuan-Fang Li
Monash University
Clayton, VIC, Australia
yuanfang.li@monash.edu

ABSTRACT

Modern Knowledge Graphs such as DBPedia contain significant information regarding Named Entities and the logical relationships which exist between them. Twitter on the other hand, contains important information on the popularity and frequency with which these entities are mentioned and discussed in combination with one another. In this paper we investigate whether these two sources of information can be used to complement and explain one another. In particular, we would like to know whether the logical relationships (a.k.a. semantic paths) which exist between pairs of known entities can help to explain the frequency with which those entities co-occur with one another in Twitter. To do this we train a ranking function over semantic paths between pairs of entities. The aim of the ranker is to identify the path that most likely explains why a particular pair of entities have appeared together in a particular tweet. We train the ranking model using a number of lexical, graph-embedding and popularity-based features over semantic paths containing a single intermediate entity and demonstrate the efficacy of the model for determining why pairs of entities occur together in tweets.

KEYWORDS

Microblog; Information Retrieval; Importance Ranking; Machine Learning; DBPedia; Knowledge Graphs; Twitter

1 INTRODUCTION

On-line social networks have become an inalienable part of many people's lives allowing them to communicate effectively with friends and colleagues. Currently about 500 million tweets are posted on Twitter per day[1]. This mountain of data provides useful information about the popularity of various named entities (people, places, products, etc.) which are also described in knowledge graphs such as DBPedia [2]. Many tweets contain more than one named entity and knowledge graphs can provide semantic relations (paths) between the entities to explain their co-occurrence. For example, Fig. 1 shows a tweet referring to two entities, Adidas and Nike Inc., and the semantic relations containing one intermediate entity between them present in DBPedia, as shown by RelFinder[2].

Figure 1: A tweet referring to the entities *Nike* and *Adidas* (above), and semantic relations linking them (below).

For any given pair of named entities, many possible paths may link them in the knowledge graph. The issue we investigate in this paper is how best to rank these relations, as represented by the intermediate entities that lie along the path between the entities, for the purpose of explaining their co-occurrence. To the best of our knowledge, this is the first research work aimed at ranking the semantic relations between popular entities in Twitter. Our method can be described as follows:

- We propose an approach for automatically labelling semantic paths for building a large training corpus of labelled semantic paths, alleviating the need for manual labelling and allowing us to scale to larger training quantities: a dataset of approx. 10 million tweets with 4 hundred thousand pairs of co-occurring entities.
- We propose several features for predicting the importance of different paths based on lexical information, knowledge graph embeddings and on-line popularity information.
- We cast the problem as a rank learning problem and train a RankSVM model to rank paths.
- Our preliminary evaluation using a human-labelled dataset in terms of NDCG@k shows promising results. Our analysis also identified features most important to the ranking algorithm.

2 METHOD

We now describe methods used for data collection, labelling and feature extraction.

*Work was conducted while on placement at Monash University, and was supported partly by the China Scholarship Council.

[1] http://www.internetlivestats.com/twitter-statistics/

[2] http://www.visualdataweb.org/relfinder/

CIKM'17, November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133161