Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks

Tian Bian,[1,2] **Xi Xiao,**[1] **Tingyang Xu,**[2] **Peilin Zhao,**[2] **Wenbing Huang,**[2] **Yu Rong,**[2] **Junzhou Huang**[2]
[1]Tsinghua University
[2]Tencent AI Lab
bt18@mails.tsinghua.edu.cn, xiaox@sz.tsinghua.edu.cn, hwenbing@126.com, yu.rong@hotmail.com
{tingyangxu, masonzhao, joehhuang}@tencent.com

Abstract

Social media has been developing rapidly in public due to its nature of spreading new information, which leads to rumors being circulated. Meanwhile, detecting rumors from such massive information in social media is becoming an arduous challenge. Therefore, some deep learning methods are applied to discover rumors through the way they spread, such as Recursive Neural Network (RvNN) and so on. However, these deep learning methods only take into account the patterns of deep propagation but ignore the structures of wide dispersion in rumor detection. Actually, propagation and dispersion are two crucial characteristics of rumors. In this paper, we propose a novel bi-directional graph model, named *Bi-Directional Graph Convolutional Networks* (Bi-GCN), to explore both characteristics by operating on both top-down and bottom-up propagation of rumors. It leverages a GCN with a top-down directed graph of rumor spreading to learn the patterns of rumor propagation; and a GCN with an opposite directed graph of rumor diffusion to capture the structures of rumor dispersion. Moreover, the information from source post is involved in each layer of GCN to enhance the influences from the roots of rumors. Encouraging empirical results on several benchmarks confirm the superiority of the proposed method over the state-of-the-art approaches.

Introduction

With the rapid development of the Internet, social media has become a convenient online platform for users to obtain information, express opinions and communicate with each other. As more and more people are keen to participate in discussions about hot topics and exchange their opinions on social media, many rumors appear. Due to a large number of users and easy access to social media, rumors can spread widely and quickly on social media, bringing huge harm to society and causing a lot of economic losses. Therefore, regarding to the potential panic and threat caused by rumors, it is urgent to come up with a method to identify rumors on social media efficiently and as early as possible.

Conventional detection methods mainly adopt handcrafted features such as user characteristics, text contents and propagation patterns to train supervised classifiers, e.g., Decision Tree (Castillo, Mendoza, and Poblete 2011), Random Forest (Kwon et al. 2013), Support Vector Machine (SVM) (Yang et al. 2012). Some studies apply more effective features, such as user comments (Giudice 2010), temporal-structural features (Wu, Yang, and Zhu 2015), and the emotional attitude of posts (Liu et al. 2015). However, those methods mainly rely on feature engineering, which is very time-consuming and labor-intensive. Moreover, those handcrafted features are usually lack of high-level representations extracted from the propagation and the dispersion of rumors.

Recent studies have exploited deep learning methods that mine high-level representations from propagation path/trees or networks to identify rumors. Many deep learning models such as Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Recursive Neural Networks (RvNN) (Ma et al. 2016; Ma, Gao, and Wong 2018) are employed since they are capable to learn sequential features from rumor propagation along time. However, these approaches have a significant limitation on efficiency since temporal-structural features only pay attention to the sequential propagation of rumors but neglect the influences of rumor dispersion. The structures of rumor dispersion also indicate some spreading behaviors of rumors. Thus, some studies have tried to involve the information from the structures of rumor dispersion by invoking Convolutional Neural Network (CNN) based methods (Yu et al. 2017; 2019). CNN-based methods can obtain the correlation features within local neighbors but cannot handle the global structural relationships in graphs or trees (Bruna et al. 2014). Therefore, the global structural features of rumor dispersion are ignored in these approaches. Actually, CNN is not designed to learn high-level representations from structured data but Graph Convolutional Network (GCN) is (Kipf and Welling 2017).

So can we simply apply GCN to rumor detection since it has successfully made progress in various fields, such as social networks (Hamilton, Ying, and Leskovec 2017), physical systems (Battaglia et al. 2016), and chemical drug discovery (Defferrard, Bresson, and Vandergheynst 2016)? The answer is no. As shown in Figure 1(a), GCN, or called undirected GCN (UD-GCN), only aggregates information relied on the relationships among relevant posts but loses the sequential orders of follows. Although UD-GCN has the ability to handle the global structural features of rumor dispersion, it does not consider the direction of the rumor propagation, which however has been shown to be an important clue for rumor detection (Wu, Yang, and Zhu 2015). Specifically, deep propagation along a relationship chain (Han et al. 2014) and wide dispersion across a social community (Thomas 2007) are two major characteristics of rumors, which is eager for a method to serve both.

To deal with both propagation and dispersion of rumors, in this paper, we propose a novel *Bi-directional GCN* (Bi-GCN), which operates on both top-down and bottom-up propagation of rumors. The proposed method obtains the features of propagation and dispersion via two parts, the Top-Down graph convolutional Networks (TD-GCN) and Bottom-Up graph convolutional Networks (BU-GCN), respectively. As shown in Figure 1(b) and 1(c), TD-GCN forwards information from the parent node of a node in a rumor tree to formulate rumor propagation while BU-GCN aggregates information from the children nodes of a node in a rumor tree to represent rumor dispersion. Then, the representations of propagation and dispersion pooled from the embedding of TD-GCN and BU-GCN are merged together through full connections to make the final results. Meanwhile, we concatenate the features of the roots in rumor trees with the hidden features at each GCN layer to enhance the influences from the roots of rumors. Moreover, we employ DropEdge (Rong et al. 2019) in the training phase to avoid over-fitting issues of our model. The main contributions of this work are as follows:

- We leverage Graph Convolutional Networks to detect rumors. To the best of our knowledge, this is the first study of employing GCN in rumor detection of social media.
- We propose the Bi-GCN model that not only considers the causal features of rumor propagation along relationship chains from top to down but also obtains the structural features from rumor dispersion within communities through the bottom-up gathering.
- We concatenate the features of the source post with other posts at each graph convolutional layer to make a comprehensive use of the information from the root feature and achieve excellent performance in rumor detection.

Experimental results on three real-world datasets show that our Bi-GCN method outperforms several state-of-the-art approaches; and for the task of early detection of rumors, which is quite crucial to identify rumors in real time and prevent them from spreading, Bi-GCN also achieves much higher effectiveness.

Related Work

In recent years, automatic rumor detection on social media has attracted a lot of attention. Most previous work for rumor detection mainly focuses on extracting rumor features from the text contents, user profiles and propagation structures to learn a classifier from labeled data (Castillo, Mendoza, and Poblete 2011; Yang et al. 2012; Kwon et al. 2013; Liu et al. 2015; Zhao, Resnick, and Mei 2015). Ma et al. (Ma et al. 2015) classified the rumor by using the time-series to model the variation of handcrafted social context features. Wu et al. (Wu, Yang, and Zhu 2015) proposed a graph kernel-based hybrid SVM classifier by combining the RBF kernel with a random-walk-based graph kernel. Ma et al. (Ma, Gao, and Wong 2017) constructed a propagation tree kernel to detect rumors by evaluating the similarities between their propagation tree structures. These methods not only were ineffective but also heavily relied on handcrafted feature engineering to extract informative feature sets.

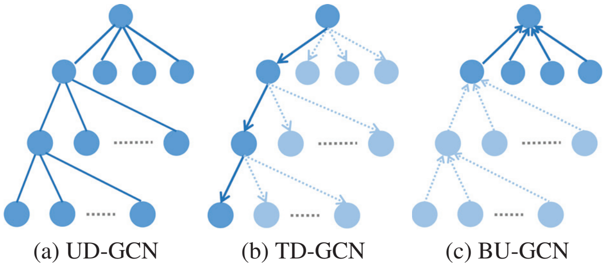


Figure 1: (a) the undirected graph with only node relationships; (b) the deep propagation along a relationship chain from top to down; (c) the aggregation of the wide dispersion within a community to an upper node.

In order to automatically learn high-level features, a number of recent methods were proposed to detect rumor based on deep learning models. Ma et al. utilized Recurrent Neural Networks (RNN) to capture the hidden representation from temporal content features (Ma et al. 2016). Chen et al. (Chen et al. 2018) improved this approach by combining attention mechanisms with RNN to focus on text features with different attentions. Yu et al. (Yu et al. 2017) proposed a method based on Convolutional Neural Network (CNN) to learn key features scattered among an input sequence and shape high-level interactions among significant features. Liu et al. (Liu and Wu 2018) incorporated both RNN and CNN to get the user features based on time series. Recently, Ma et al. (Ma, Gao, and Wong 2019) employed the adversarial learning method to improve the performance of rumor classifier, where the discriminator is used as a classifier and the corresponding generator improves the discriminator by generating conflicting noises. In addition, Ma et al. built a tree-structured Recursive Neural Networks (RvNN) to catch the hidden representation from both propagation structures and text contents (Ma, Gao, and Wong 2018). However, these methods are too inefficient to learn the features of the propagation structure, and they also ignore the global structural features of rumor dispersion.

Compared to the deep-learning models mentioned above, GCN is able to capture global structural features from graphs or trees better. Inspired by the success of CNN in the field of computer vision, GCN has demonstrated state-of-the-art performances in various tasks with graph data (Battaglia et al. 2016; Defferrard, Bresson, and Vandergheynst 2016; Hamilton, Ying, and Leskovec 2017). Scarselli et al. (Scarselli et al. 2008) firstly introduced GCN as a special massage-passing model for either undirected graphs or directed graphs. Later on, Bruna et al. (Bruna et al. 2014) theoretically analyzed graph convolutional methods for undirected graphs based on the spectral graph theory. Subsequently, Defferrard et al. (Defferrard, Bresson, and Vandergheynst 2016) developed a method named the Chebyshev Spectral CNN (ChebNet) and used the Chebyshev polynomials as the filter. After this work, Kipf et al. (Kipf and Welling 2017) presented a first-order approximation of ChebNet (1stChebNet), where the information of each node is aggregated from the node itself and its neighboring nodes. Our rumor detection model is inspired by the GCN.

Preliminaries

We introduce some fundamental concepts that are necessary for our method. First the notation used in this paper is as follows.

Notation

Let $C = \{c_1, c_2, ..., c_m\}$ be the rumor detection dataset, where c_i is the i-th event and m is the number of events. $c_i = \{r_i, w_1^i, w_2^i, ..., w_{n_i-1}^i, G_i\}$, where n_i refers to the number of posts in c_i, r_i is the source post, each w_j^i represents the j-th relevant responsive post, and G_i refers to the propagation structure. Specifically, G_i is defined as a graph $\langle V_i, E_i \rangle$ with r_i being the root node (Wu, Yang, and Zhu 2015; Ma, Gao, and Wong 2017), where $V_i = \{r_i, w_1^i, \dots, w_{n_i-1}^i\}$, and $E_i = \{e_{st}^i | s, t = 0, \dots, n_i - 1\}$ that represents the set of edge from responded posts to the retweeted posts or responsive posts, as shown in Figure 1(b). For example, if w_2^i has a response to w_1^i, there will be an directed edge $w_1^i \rightarrow w_2^i$, i.e., e_{12}^i. If w_1^i has a response to r_i, there will be an directed edge $r_i \rightarrow w_1^i$, i.e., e_{01}^i. Denote $\mathbf{A}_i \in \{0, 1\}^{n_i \times n_i}$ as an adjacency matrix where

$$a_{ts}^i = \begin{cases} 1, & \text{if } e_{st}^i \in E_i \\ 0, & \text{otherwise} \end{cases}.$$

Denote $\mathbf{X}_i = [\mathbf{x}_0^{i\top}, \mathbf{x}_1^{i\top}, ..., \mathbf{x}_{n_i-1}^{i\top}]^\top$ as a feature matrix extracted from the posts in c_i, where $\mathbf{x}_0^i$ represents the feature vector of r_i and each other row feature $\mathbf{x}_j^i$ represents the feature vector of w_j^i.

Moreover, each event c_i is associated with a ground-truth label $y_i \in \{F, T\}$ (i.e., False Rumor or True Rumor). In some cases, the label y_i is one of the four finer-grained classes $\{N, F, T, U\}$ (i.e., Non-rumor, False Rumor, True Rumor, and Unverified Rumor) (Ma, Gao, and Wong 2017; Zubiaga et al. 2018). Given the dataset, the goal of rumor detection is to learn a classifier

$$f : C \rightarrow Y,$$

where C and Y are the sets of events and labels respectively, to predict the label of an event based on text contents, user information and propagation structure constructed by the related posts from that event.

Graph Convolutional Networks

Recently, there is an increasing interest in generalizing convolutions to the graph domain. Among all the existing works, GCN is one of the most effective convolution models, whose convolution operation is considered as a general "message-passing" architecture as follows:

$$\mathbf{H}_k = M(\mathbf{A}, \mathbf{H}_{k-1}; \boldsymbol{W}_{k-1}), \quad (1)$$

where $\mathbf{H}_k \in \mathbb{R}^{n \times v_k}$ is the hidden feature matrix computed by the $k - th$ Graph Conventional Layer (GCL) and M is the message propagation function, which depends on the adjacency matrix $\mathbf{A}$, the hidden feature matrix $\mathbf{H}_{k-1}$ and the trainable parameters $\boldsymbol{W}_{k-1}$.

There are many kinds of message propagation functions M for GCN (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016). Among them, the message propagation function defined in the first-order approximation of ChebNet (1stChebNet) (Kipf and Welling 2017) is as follows:

$$\mathbf{H}_k = M(\mathbf{A}, \mathbf{H}_{k-1}; \boldsymbol{W}_{k-1}) = \sigma(\hat{\mathbf{A}}\mathbf{H}_{k-1}\boldsymbol{W}_{k-1}). \quad (2)$$

In the above equation $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is the normalized adjacency matrix, where $\tilde{\mathbf{A}}$=$\mathbf{A} + \mathbf{I}_N$ (i.e., adding self-connection), $\tilde{\mathbf{D}}_{ii}$=$\Sigma_j \tilde{\mathbf{A}}_{ij}$ that represents the degree of the $i - th$ node; $\boldsymbol{W}_{k-1} \in \mathbb{R}^{v_{k-1} \times v_k}$; and $\sigma(\cdot)$ is an activation function, e.g., the ReLU function.

DropEdge

DropEdge is a novel method to reduce over-fitting for GCN-based models (Rong et al. 2019). In each training epoch, it randomly drops out edges from the input graphs to generate different deformed copies with certain rate. As a result, this method augments the randomness and the diversity of input data, just like rotating or flapping images at random. Formally, suppose the total number of edges in the graph $\mathbf{A}$ is N_e and the dropping rate is p, then the adjacency matrix after DropEdge, $\boldsymbol{A}'$, is computed as below:

$$\mathbf{A}' = \mathbf{A} - \mathbf{A}_{drop} \quad (3)$$

where $\mathbf{A}_{drop}$ is the matrix constructed using $N_e \times p$ edges randomly sampled from the original edge set.

Bi-GCN Rumor Detection Model

In this section, we propose an effective GCN-based method for rumor detection based on the rumor propagation and the rumor dispersion, named as *Bi-directional Graph Convolutional Networks* (Bi-GCN). The core idea of Bi-GCN is to learn suitable high-level representations from both rumor propagation and rumor dispersion. In our Bi-GCN model, two-layer 1stChebNet are adopted as the fundamental GCN components. As shown in Figure 2, we elaborate the rumor detection process using Bi-GCN in 4 steps.

We first discuss how to apply the Bi-GCN model to one event, i.e., $c_i \rightarrow y_i$ for the i-th event. The other events are calculated in the same manner. To better present our method, we omit the subscript $_i$ in the following content.

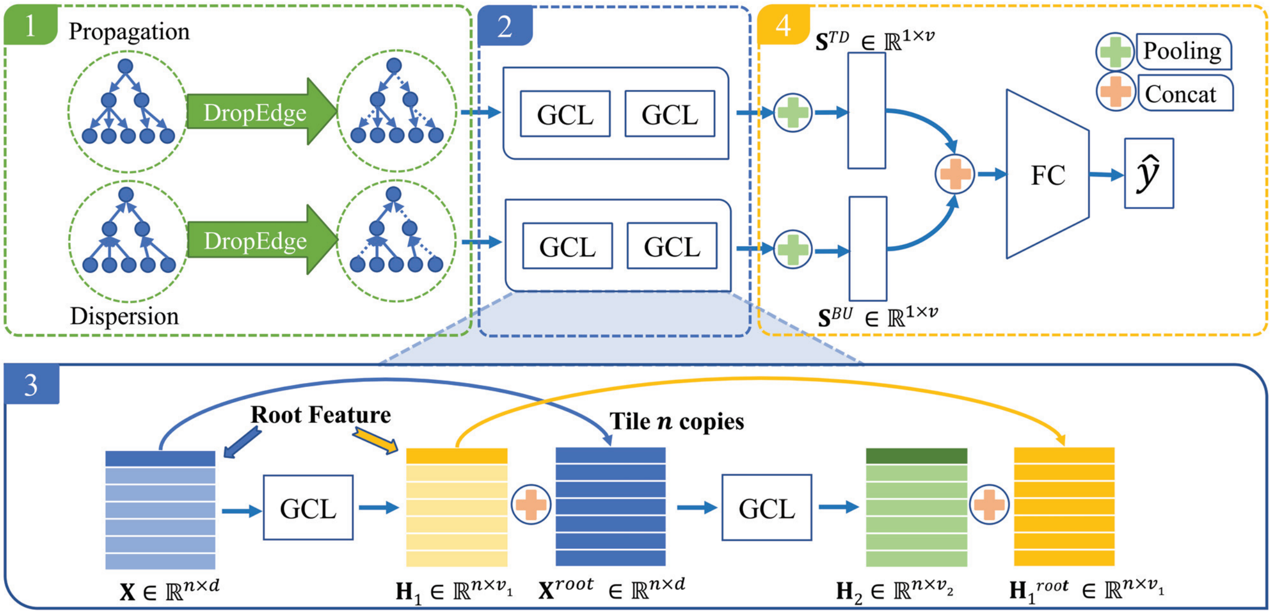

1
Propagation
DropEdge
DropEdge
Dispersion
2
GCL
GCL
GCL
GCL
4
$\mathbf{S}^{TD} \in \mathbb{R}^{1\times v}$
$\mathbf{S}^{BU} \in \mathbb{R}^{1\times v}$
Pooling
Concat
FC
$\hat{y}$
3
Root Feature
Tile n copies
GCL
GCL
$\mathbf{X} \in \mathbb{R}^{n\times d}$
$\mathbf{H}_1 \in \mathbb{R}^{n\times v_1}$
$\mathbf{X}^{root} \in \mathbb{R}^{n\times d}$
$\mathbf{H}_2 \in \mathbb{R}^{n\times v_2}$
$\mathbf{H}_1{}^{root} \in \mathbb{R}^{n\times v_1}$

Figure 2: Our Bi-GCN rumor detection model. $\mathbf{X}$ denotes the original feature matrix input to the Bi-GCN model, and $\mathbf{H}_k$ is the hidden features matrix generated from the $k-th$ GCL. $\mathbf{X}^{root}$ and $\mathbf{H}_1^{root}$ represents the matrix extended by the features of source post.

1 Construct Propagation and Dispersion Graphs

Based on the retweet and response relationships, we construct the propagate structure $\langle V, E\rangle$ for a rumor event c_i. Then, let $\mathbf{A} \in \mathbb{R}^{n_i\times n_i}$ and $\mathbf{X}$ be its corresponding adjacency matrix and feature matrix of c_i based on the spreading tree of rumors, respectively. $\mathbf{A}$ only contains the edges from the upper nodes to the lower nodes as illustrated in Figure 1(b). At each training epoch, p percentage of edges are dropped via Eq. (3) to form $\mathbf{A}'$, which avoid penitential overfitting issues (Rong et al. 2019). Based on $\mathbf{A}'$ and $\mathbf{X}$, we can build our Bi-GCN model. Our Bi-GCN consists of two components: a Top-Down Graph Convolutional Network (TD-GCN) and a Bottom-Up Graph Convolutional Network (BU-GCN). The adjacency matrices of two components are different. For TD-GCN, the adjacency matrix is represented as $\mathbf{A}^{TD} = \mathbf{A}'$. Meanwhile, for BU-GCN, the adjacency matrix is $\mathbf{A}^{BU} = \mathbf{A}'^{\top}$. TD-GCN and BU-GCN adopt the same feature matrix $\mathbf{X}$.

2 Calculate the High-level Node Representations

After the DropEdge operation, the top-down propagation features and the bottom-up propagation features are obtained by TD-GCN and BU-GCN, respectively.

By substituting $\mathbf{A}^{TD}$ and $\mathbf{X}$ to Eq. (2) over two layers, we write the equations for TD-GCN as below:

$$\mathbf{H}_1^{TD} = \sigma\left(\hat{\mathbf{A}}^{TD}\mathbf{X}\boldsymbol{W}_0^{TD}\right), \tag{4}$$

$$\mathbf{H}_2^{TD} = \sigma\left(\hat{\mathbf{A}}^{TD}\mathbf{H}_1^{TD}\boldsymbol{W}_1^{TD}\right), \tag{5}$$

where $\mathbf{H}_1^{TD} \in \mathbb{R}^{n\times v_1}$ and $\mathbf{H}_2^{TD} \in \mathbb{R}^{n\times v_2}$ represent the hidden features of two layer TD-GCN. $\boldsymbol{W}_0^{TD} \in \mathbb{R}^{d\times v_1}$ and $\boldsymbol{W}_1^{TD} \in \mathbb{R}^{v_1\times v_2}$ are the filter parameter matrices of TD-GCN. Here we adopt ReLU function as the activation function, $\sigma(\cdot)$. Dropout (Srivastava et al. 2014) is applied on GCN Layers (GCLs) to avoid over-fitting. Similar to Eqs. (4) and (5), we calculate the bottom-up hidden features $\mathbf{H}_1^{BU}$ and $\mathbf{H}_2^{BU}$ for BU-GCN in the same manner as Eq. (4) and Eq. (5).

3 Root Feature Enhancement

As we know, the source post of a rumor event always has abundant information to make a wide impact. It is necessary to better make use of the information from the source post, and learn more accurate node representations from the relationship between nodes and the source post.

Consequently, besides the hidden features from TD-GCN and BU-GCN, we propose an operation of root feature enhancement to improve the performance of rumor detection as shown in Figure 2. Specifically, for TD-GCN at the k-th GCL, we concatenate the hidden feature vectors of every nodes with the hidden feature vector of the root node from the $(k-1)$-th GCL to construct a new feature matrix as

$$\tilde{\mathbf{H}}_k^{TD} = \text{concat}(\mathbf{H}_k^{TD}, (\mathbf{H}_{k-1}^{TD})^{root}) \tag{6}$$

with $\mathbf{H}_0^{TD} = \mathbf{X}$. Therefore, we express TD-GCN with the root feature enhancement by replacing $\mathbf{H}_1^{TD}$ in Eq. (5) with $\tilde{\mathbf{H}}_1^{TD} = \text{concat}(\mathbf{H}_1^{TD}, \mathbf{X}^{root})$, and then get $\tilde{\mathbf{H}}_2^{TD}$ as follows:

$$\mathbf{H}_2^{TD} = \sigma\left(\hat{\mathbf{A}}^{TD}\tilde{\mathbf{H}}_1^{TD}\boldsymbol{W}_1^{TD}\right), \tag{7}$$

$$\tilde{\mathbf{H}}_2^{TD} = \text{concat}(\mathbf{H}_2^{TD}, (\mathbf{H}_1^{TD})^{root}). \tag{8}$$

Similarly, the hidden feature metrics of BU-GCN with root feature enhancement, $\tilde{\mathbf{H}}_1^{BU}$ and $\tilde{\mathbf{H}}_2^{BU}$, are obtained in the same manner as Eq. (7) and Eq. (8).

4 Representations of Propagation and Dispersion for Rumor Classification

The representations of propagation and dispersion are the aggregations from the node representations of TD-GCN and BU-GCN, respectively. Here we employ mean-pooling operators to aggregate information from these two sets of the node representations. It is formulated as

$$\mathbf{S}^{TD} = \text{MEAN}(\tilde{\mathbf{H}}_2^{TD}), \tag{9}$$

$$\mathbf{S}^{BU} = \text{MEAN}(\tilde{\mathbf{H}}_2^{BU}). \tag{10}$$

Then, we concatenate the representations of propagation and the representation of dispersion to merge the information as

$$\mathbf{S} = \text{concat}(\mathbf{S}^{TD}, \mathbf{S}^{BU}). \tag{11}$$

Finally, the label of the event $\hat{\mathbf{y}}$ is calculated via several full connection layers and a softmax layer:

$$\hat{\mathbf{y}} = Softmax(FC(\mathbf{S})). \tag{12}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{1\times C}$ is a vector of probabilities for all the classes used to predict the label of the event.

We train all the parameters in the Bi-GCN model by minimizing the cross-entropy of the predictions and ground truth distributions, Y, over all events, C. L_2 regularizer is applied in the loss function over all the model parameters.

Experiments

In this section, we first evaluate the empirical performance of our proposed Bi-GCN method in comparison with several baseline models. Then, we investigate the effect of each variant of the proposed method. Finally, we also examine the capability of early rumor detection for both the proposed method and the compared methods.

Settings and Datasets

Datasets We evaluate our proposed method on three real-world datasets: *Weibo* (Ma et al. 2016), *Twitter15* (Ma, Gao, and Wong 2017), and *Twitter16* (Ma, Gao, and Wong 2017). *Weibo* and *Twitter* are the most popular social media sites in China and the U.S., respectively. In all the three datasets, nodes refer to users, edges represent retweet or response relationships, and features are the extracted top-5000 words in terms of the TF-IDF values as mentioned in the Bi-GCN Rumor Detection Model Section. The *Weibo* dataset contains two binary labels: False Rumor (F) and True Rumor (T), while *Twitter15* and *Twitter16* datasets contains four labels: Non-rumor (N), False Rumor (F), True Rumor (T), and Unverified Rumor (U). The label of each event in *Weibo* is annotated according to Sina community management center, which reports various misinformation (Ma et al. 2016). And the label of each event in *Twitter15* and *Twitter16* is annotated according to the veracity tag of the article in rumor debunking websites (e.g., snopes.com, Emergent.info, etc) (Ma, Gao, and Wong 2017). The statistics of the three datasets are shown in Table 1.

Table 1: Statistics of the datasets

Statistic	*Weibo*	*Twitter15*	*Twitter16*
# of posts	3,805,656	331,612	204,820
# of Users	2,746,818	276,663	173,487
# of events	4664	1490	818
# of True rumors	2351	374	205
# of False rumors	2313	370	205
# of Unverified rumors	0	374	203
# of Non-rumors	0	372	205
Avg. time length / event	2,460.7 Hours	1,337 Hours	848 Hours
Avg. # of posts / event	816	223	251
Max # of posts / event	59,318	1,768	2,765
Min # of posts / event	10	55	81

Experimental Setup We compare the proposed method with some state-of-the-art baselines, including:

- DTC (Castillo, Mendoza, and Poblete 2011): A rumor detection method using a Decision Tree classifier based on various handcrafted features to obtain information credibility.
- SVM-RBF (Yang et al. 2012): A SVM-based model with RBF kernel, using handcrafted features based on the overall statistics of the posts.
- SVM-TS (Ma et al. 2015): A linear SVM classifier that leverages handcrafted features to construct time-series model.
- SVM-TK (Ma, Gao, and Wong 2017): A SVM classifier with a propagation Tree Kernel on the basis of the propagation structures of rumors.
- RvNN (Ma, Gao, and Wong 2018): A rumor detection approach based on tree-structured recursive neural networks with GRU units that learn rumor representations via the propagation structure.
- PPC_RNN+CNN (Liu and Wu 2018): A rumor detection model combining RNN and CNN, which learns the rumor representations through the characteristics of users in the rumor propagation path.
- Bi-GCN: Our GCN-based rumor detection model utilizing the Bi-directional propagation structure.

We implement DTC and SVM-based models with scikit-learn[1]; PPC_RNN+CNN with Keras[2]; RvNN and our method with Pytorch[3]. To make a fair comparison, we randomly split the datasets into five parts, and conduct 5-fold cross-validation to obtain robust results. For the *Weibo* dataset, we evaluate the Accuracy (Acc.) over the two categories and Precision (Prec.), Recall (Rec.), F1 measure (F_1) on each class. For the two *Twiter* datasets, we evaluate Acc. over the four categories and F_1 on each class. The

[1] https://scikit-learn.org

[2] https://keras.io/

[3] https://pytorch.org/

Table 2: Rumor detection results on *Weibo* dataset (F: False Rumor; T: True Rumor)

Method	Class	Acc.	Prec.	Rec.	F_1
DTC	F	0.831	0.847	0.815	0.831
	T		0.815	0.824	0.819
SVM-RBF	F	0.879	0.777	0.656	0.708
	T		0.579	0.708	0.615
SVM-TS	F	0.885	0.950	0.932	0.938
	T		0.124	0.047	0.059
RvNN	F	0.908	0.912	0.897	0.905
	T		0.904	0.918	0.911
PPC_RNN+CNN	F	0.916	0.884	0.957	0.919
	T		0.955	0.876	0.913
Bi-GCN	F	**0.961**	**0.961**	**0.964**	**0.961**
	T		**0.962**	**0.962**	**0.960**

Table 3: Rumor detection results on *Twitter15* and *Twitter16* datasets (N: Non-Rumor; F: False Rumor; T: True Rumor; U: Unverified Rumor)

Twitter15

Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.454	0.415	0.355	0.733	0.317
SVM-RBF	0.318	0.225	0.082	0.455	0.218
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-TK	0.750	0.804	0.698	0.765	0.733
RvNN	0.723	0.682	0.758	0.821	0.654
PPC_RNN+CNN	0.477	0.359	0.507	0.300	0.640
Bi-GCN	**0.886**	**0.891**	**0.860**	**0.930**	**0.864**

Twitter16

Method	Acc.	N	F	T	U
		F_1	F_1	F_1	F_1
DTC	0.473	0.254	0.080	0.190	0.482
SVM-RBF	0.553	0.670	0.085	0.117	0.361
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.732	0.740	0.709	0.836	0.686
RvNN	0.737	0.662	0.743	0.835	0.708
PPC_RNN+CNN	0.564	0.591	0.543	0.394	0.674
Bi-GCN	**0.880**	**0.847**	**0.869**	**0.937**	**0.865**

parameters of Bi-GCN are updated using stochastic gradient descent, and we optimize the model by Adam algorithm (Kingma and Ba 2014). The dimension of each node's hidden feature vectors are 64. The dropping rate in DropEdge is 0.2 and the rate of dropout is 0.5. The training process is iterated upon 200 epochs, and early stopping (Yao, Rosasco, and Caponnetto 2007) is applied when the validation loss stops decreasing by 10 epochs. Note that we do not employ SVM-TK on the *Weibo* dataset due to its exponential complexity on large datasets.

Overall Performance

Table 2 and Table 3 show the performance of the proposed method and all the compared methods on the *Weibo* and *Twitter* datasets, respectively.

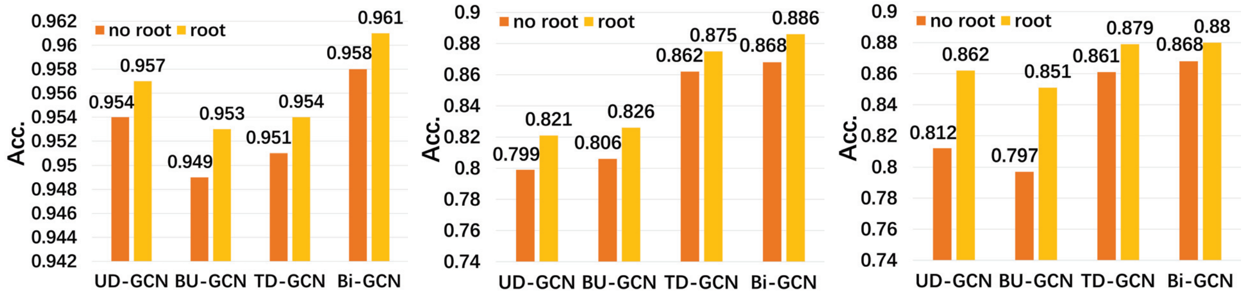
First, among the baseline algorithms, we observe that the deep learning methods performs significantly better than those using hand-crafted features. It is not surprising, since the deep learning methods are able to learn high-level representations of rumors to capture valid features. This demonstrates the importance and necessity of studying deep learning for rumor detection.

Second, the proposed method outperforms the PPC_RNN+CNN method in terms of all the performance measures, which indicates the effectiveness of incorporating the dispersion structure for rumor detection. Since RNN and CNN cannot process data with the graph structure, PPC_RNN+CNN ignores important structural features of rumor dispersion. This prevents it from obtaining efficient high-level representations of rumors, resulting in worse performance on rumor detection.

Finally, Bi-GCN is significantly superior to the RvNN method. Since RvNN only uses the hidden feature vector of all the leaf nodes so that it is heavily impacted by the information of the latest posts. However, the latest posts are always lack of information such as comments, and just follow the former posts. Unlike RvNN, the root feature enhancement allows the proposed method to pay more attention to the information of the source posts, which helps improve our models much more.

Ablation Study

To analyze the effect of each variant of Bi-GCN, we compare the proposed method with TD-GCN, BU-GCN, UD-GCN and their variants without the root feature enhancement. The empirical results are summarized in Figure 3. UD-GCN, TD-GCN, and BU-GCN represent our GCN-based rumor detection models utilize the UnDirected, Top-Down and Bottom-Up structures, respectively. Meanwhile, ”root” refers to the GCN-based model with concatenating root features in the networks while ”no root” represents the GCN-based model without concatenating root features in the networks. Some conclusions are drawn from Figure 3. First, Bi-GCN, TD-GCN, BU-GCN, and UD-GCN outperforms their variants without the root feature enhancement, respectively. This indicates that the source posts plays an important role in rumor detection. Second, TD-GCN and BU-GCN can not always achieve better results than UD-GCN, but Bi-GCN is always superior to UD-GCN, TD-GCN and BU-GCN. This implies the importance to simultaneously consider both top-down representations from the ancestor nodes, and bottom-up representations from the children nodes. Finally, even the worst results in Figures 3 are better than those of other baseline methods in Table 2 and 3 by a large gap, which again verifies the effectiveness of graph convolution for rumor de-

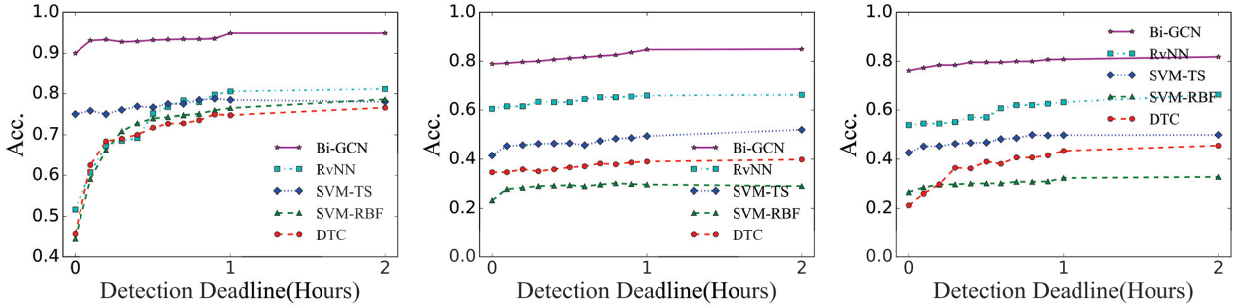


(a) *Weibo* dataset

(b) *Twitter15* dataset

(c) *Twitter16* dataset

Figure 3: The rumor detection performance of the GCN-based methods on three datasets



(a) *Weibo* dataset

(b) *Twitter15* dataset

(c) *Twitter16* dataset

Figure 4: Result of rumor early detection on three datasets

tection.

Early Rumor Detection

Early detection aims to detect rumor at the early stage of propagation, which is another important metric to evaluate the quality of the method. To construct an early detection task, we set up a series of detection deadlines and only use the posts released before the deadlines to evaluate the accuracy of the proposed method and baseline methods. Since it is difficult for the PPC_RNN+CNN method to process the data of variational lengths, we cannot get the accurate results of PPC_RNN+CNN at each deadline in this task, so it is not compared in this experiment.

Figure 4 shows the performances of our Bi-GCN method versus RvNN, SVM-TS, SVM-RBF and DTC at various deadlines for the *Weibo* and *Twitter* datasets. From the figure, it can be seen that the proposed Bi-GCN method reaches relatively high accuracy at a very early period after the source post initial broadcast. Besides, the performance of Bi-GCN is remarkably superior to other models at each deadline, which demonstrates that structural features are not only beneficial to long-term rumor detection, but also helpful to the early detection of rumors.

Conclusions

In this paper, we propose a GCN-based model for rumor detection on social media, called Bi-GCN. Its inherent GCN model gives the proposed method the ability of processing graph/tree structures and learning higher-level representations more conducive to rumor detection. In addition, we also improve the effectiveness of the model by concatenating the features of the source post after each GCL of GCN. Meanwhile, we construct several variants of Bi-GCN to model the propagation patterns, i.e., UD-GCN, TD-GCN and BU-GCN. The experimental results on three real-world datasets demonstrate that the GCN-based approaches outperform state-of-the-art baselines in very large margins in terms of both accuracy and efficiency. In particular, the Bi-GCN model achieves the best performance by considering both the causal features of rumor propagation along relationship chains from top to down propagation pattern and the structural features from rumor dispersion within communities through the bottom-up gathering.

Acknowledgments

The authors would like to thank the support of Tencent AI Lab and Tencent Rhino-Bird Elite Training Program. This work is supported by the National Natural Science Foundation of Guangdong Province (2018A030313422), National Natural Science Foundation of China (Grant No. 61773229, No. 61972219) and Overseas Cooperation Research Fund of Graduate School at Shenzhen, Tsinghua University (Grant No. HW2018002).

References

Battaglia, P.; Pascanu, R.; Lai, M.; Rezende, D. J.; et al. 2016. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, 4502–4510.

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, http–openreview.

Castillo, C.; Mendoza, M.; and Poblete, B. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, 675–684. ACM.

Chen, T.; Li, X.; Yin, H.; and Zhang, J. 2018. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 40–52. Springer.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, 3844–3852.

Giudice, K. D. 2010. Crowdsourcing credibility: The impact of audience feedback on web page credibility. In *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem-Volume 47*, 59. American Society for Information Science.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 1024–1034.

Han, S.; Zhuang, F.; He, Q.; Shi, Z.; and Ao, X. 2014. Energy model for rumor propagation on social networks. *Physica A: Statistical Mechanics and its Applications* 394:99–109.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, N., T., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

Kwon, S.; Cha, M.; Jung, K.; Chen, W.; and Wang, Y. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th International Conference on Data Mining*, 1103–1108. IEEE.

Liu, Y., and Wu, Y.-F. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 354–361. AAAI press.

Liu, X.; Nourbakhsh, A.; Li, Q.; Fang, R.; and Shah, S. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1867–1870. ACM.

Ma, J.; Gao, W.; Wei, Z.; Lu, Y.; and Wong, K.-F. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1751–1754. ACM.

Ma, J.; Gao, W.; Mitra, P.; Kwon, S.; Jansen, B. J.; Wong, K.-F.; and Cha, M. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Ijcai*, 3818–3824.

Ma, J.; Gao, W.; and Wong, K.-F. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 708–717.

Ma, J.; Gao, W.; and Wong, K.-F. 2018. Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1980–1989.

Ma, J.; Gao, W.; and Wong, K.-F. 2019. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *The World Wide Web Conference*, 3049–3055. ACM.

Rong, Y.; Huang, W.; Xu, T.; and Junzhou, H. 2019. The truly deep graph convolutional networks for node classification. *arXiv preprint arXiv:1907.10903*.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20(1):61–80.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Thomas, S. A. 2007. Lies, damn lies, and rumors: an analysis of collective efficacy, rumors, and fear in the wake of katrina. *Sociological Spectrum* 27(6):679–703.

Wu, K.; Yang, S.; and Zhu, K. Q. 2015. False rumors detection on sina weibo by propagation structures. In *2015 IEEE 31st international conference on data engineering*, 651–662. IEEE.

Yang, F.; Liu, Y.; Yu, X.; and Yang, M. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, 13. ACM.

Yao, Y.; Rosasco, L.; and Caponnetto, A. 2007. On early stopping in gradient descent learning. *Constructive Approximation* 26(2):289–315.

Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2017. A convolutional approach for misinformation identification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 3901–3907. AAAI Press.

Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2019. Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. *Computers & Security* 83:106–121.

Zhao, Z.; Resnick, P.; and Mei, Q. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, 1395–1405. International World Wide Web Conferences Steering Committee.

Zubiaga, A.; Aker, A.; Bontcheva, K.; Liakata, M.; and Procter, R. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51(2):32.