Graph Convolutional Networks with Markov Random Field Reasoning for Social Spammer Detection

Yongji Wu,[1] Defu Lian,[1*] Yiheng Xu,[2] Le Wu,[3] Enhong Chen[1]
[1]University of Science and Technology of China
[2]Harbin Institute of Technology, [3]Hefei University of Technology
{wuyongji317, dove.ustc, hi.ranpox, lewu.ustc}@gmail.com, cheneh@ustc.edu.cn

Abstract

The recent growth of social networking platforms also led to the emergence of social spammers, who overwhelm legitimate users with unwanted content. The existing social spammer detection methods can be characterized into two categories: features based ones and propagation-based ones. Features based methods mainly rely on matrix factorization using tweet text features, and regularization using social graphs is incorporated. However, these methods are fully supervised and can only utilize labeled part of social graphs, which fail to work in a real-world semi-supervised setting. The propagation-based methods primarily employ Markov Random Fields (MRFs) to capture human intuitions in user following relations, which cannot take advantages of rich text features. In this paper, we propose a novel social spammer detection model based on Graph Convolutional Networks (GCNs) that operate on directed social graphs by explicitly considering three types of neighbors. Furthermore, inspired by the propagation-based methods, we propose a MRF layer with refining effects to encapsulate these human insights in social relations, which can be formulated as a RNN through mean-field approximate inference, and stack on top of GCN layers to enable end-to-end training. We evaluate our proposed method on two real-world social network datasets, and the results demonstrate that our method outperforms the state-of-the-art approaches.

Introduction

Online Social Networks (OSNs) such as Facebook and Twitter have gained increasing popularity in recent years for users to interact and communicate. Nowadays, they have become a universal platform for users to discuss events and share personal experience. However, with this growing popularity, a new kind of malicious users known as social spammers surface (Webb, Caverlee, and Pu 2008). These spammers launch various attacks on social networks with fake accounts. For instance, spreading advertisement to promote sales, posting tweets containing links to pornographic sites (Singh, Bansal, and Sofat 2016), or hijacking trend topics (VanDam and Tan 2016). A recent study (Varol et al. 2017) estimated that between 9% and 15% of active Twitter accounts are fake. The malicious behavior of social spammers poses a severe threat to the quality of user experience, hence effectively identifying these spammers is of great real-world importance in the development of OSNs.

A number of social spammer detection methods have been proposed, following spam detection in traditional environments like email (Blanzieri and Bryl 2008) and Web pages (Gyongyi and Garcia-Molina 2005). Most existing models can be divided into two categories: features based approaches and propagation-based ones. Features based methods (Zhu et al. 2012; Hu et al. 2013; 2014; Hu, Tang, and Liu 2014; Shen et al. 2017) generally exploit text features mined from tweets posted by users. Matrix factorization is performed on these features, and social graphs are used in regularization. However, these matrix factorization based methods are fully supervised; they can only utilize labeled part of the social graph, hence need a large number of labeled samples to work successfully. Recently, Li et al. (2018) proposed a semi-supervised model using an autoencoder framework. This model uses node2vec and doc2vec embeddings as input features for text view and social graph view. However, it fails to capture the interactions between users in social graphs explicitly. The propagation-based methods (Wang, Zhang, and Gong 2017; Wang, Gong, and Fu 2017), which are also called guilt-by-association methods, assume some sort of correlations between a pair of users and model these intuitions using a Markov Random Field (MRF). However, these methods simply perform the computation of posterior distributions of MRFs using a pre-defined pairwise influence weight, and cannot benefit from features in tweet text.

To this end, we propose a novel model for social spammer detection to take advantage of both features based and propagation-based methods. Since Graph Convolutional Networks (GCNs) which are developed in recent years (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016) can combine both graph structures and node features for semi-supervised learning, we use them as the building block of our model. Besides, the ability of GCNs to propagate information layer-wisely allows them to learn localized patterns at different scales. In our scenario, we consider that different directions in users' following relations entail differ-

[*]Corresponding author

ent underlying behaviors of users. Hence we assign an independent weight matrix for each different type of neighbor in the message-passing process of GCNs. We further propose to stack a MRF layer with refining effects on top of GCN. The MRF layer captures human insights of neighbors' influences on a user's identity (for instance, spammers tend to follow a large number of users). It is able to fix incorrect predictions made by GCN. We use the mean-field approximation to compute posterior distributions of the MRF and formulate it as a Recurrent Neural Network (RNN) which performs multi-step inference to ensure convergence.

The main contributions of this paper are listed as follows:

1. We propose a novel end-to-end deep learning model for social spammer detection based on GCNs that operate on directed social graphs, and a MRF layer that captures human insights in user following relations to refine predictions made by GCN. To the best of our knowledge, this is the first semi-supervised social spammer detection model that seamlessly integrates both features based methods and propagation-based ones.
2. We formulate the computation of the posterior distributions of MRF as a RNN which computes the result of each time step based on outputs from the previous time step using the same weight matrix, and stack it on top of GCN layers. We empirically investigate the indispensability of multi-step inference through RNN in the MRF layer.
3. We conduct extensive experiments on two real-world Twitter datasets and achieve superior performance. We illustrate the refining effects of the MRF layer by giving concrete examples. We also demonstrate the vital role of explicitly considering three kinds of neighbors in GCN, as well as the importance of jointly training GCN and MRF.

Related Work

Social spammer detection. There are many studies on social spammer detection. Zhu et al. (2012) proposed one of the earliest social spammer detection approach based on matrix factorization, where undirected graph Laplacian is used to incorporate the topology information and multi-label informed latent semantic indexing is used to model the context information. Hu, Tang, and Liu (2014) extended this method to exploit the direction information of user following relations. Sentiment information is considered to assist matrix factorization in (Hu et al. 2014). Fu et al. (2017) investigated the carefulness of users in social networks and how the robustness of the detection algorithms can be improved with the aid of user carefulness. Shen et al. (2017) considered matrix factorization by exploiting multi-view data. Li et al. (2018) further extended it to a semi-supervised setting by using a ladder-network-based autoencoder model. Wang, Zhang, and Gong (2017) proposed a guilt-by-association method using a MRF to propagate the given label information among the graph to predict labels of the remaining nodes. This method is extended to consider directed social graphs in (Wang, Gong, and Fu 2017). Our method also uses the intuitions in (Wang, Gong, and Fu 2017) but is quite different from it. We propose a GCN-based framework, along with a MRF layer formulated as a RNN stacked on top of GCN layers and can be jointly trained with GCN; while they simply compute the posterior distribution using pre-defined weights through loopy belief propagation, given a few labeled nodes.

Graph convolutional networks. In recent years, considerable efforts have been devoted to extending traditional convolutional neural networks (CNNs) which operate on Euclidean structures to arbitrary graphs. Bruna et al. (2014) first proposed the convolution operation on graphs based on spectral graph theory, which is extended in (Defferrard, Bresson, and Vandergheynst 2016) through using Chebyshev polynomials to approximate filters. Kipf and Welling (2016) further applied the first-order approximation to develop a layer-wise linear model for fast and scalable semi-supervised node classification. GCNs have since been utilized in many fields. (Wu et al. 2019) proposed GCNs on User Mobility Heterogeneous Graphs to infer social relations from trajectory data. (Wang, Lian, and Ge 2019) distilled the ranking information derived from GCN into binarized collaborative filtering to improve the efficiency of online recommendation. Jin et al. (2019) integrated MRF and GCN for semi-supervised community detection. However, our method is quite different from theirs. While they use a fully-connected MRF for community detection, we propose a novel sparse MRF for spammer detection by modeling intuitions about different types of neighbors' influences on a user's label, which can be effectively implemented using sparse-dense matrix multiplication with linear time complexity. Furthermore, (Jin et al. 2019) performs only one-step inference when computing posterior distributions of MRF, in which case convergence cannot be guaranteed and would result in poor performance. We fix this problem by formulating the MRF layer as a RNN and conduct multi-step inference.

Proposed Method

Social spammer detection is essentially a two-class classification problem. We aim to build a classifier to accurately assign identity labels for users in the test set, given a training user set, the social network, and/or features of each user. Our proposed social spammer detection model is built on the basis of GCN and MRF. First, graph convolution is performed on directed social graphs by explicitly considering different types of neighbors. Then we present three intuitions of neighbors' influences on a user's label. These intuitions are captured using a pairwise MRF. We formulate the MRF as a RNN to perform multi-step inference. Finally, we stack it on top of GCN layers and end-to-end train the whole model.

GCN on Directed Social Networks

In this section, we introduce the concept of directed social networks and how we perform graph convolution on them.

Social networks are inherently directed. In order to obtain decent performance, one must exploit the directional

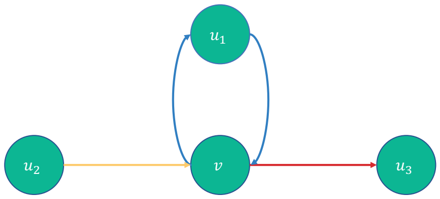


Figure 1: Illustration of three types of neighbors. Here u_1, u_2, u_3 are bidirectional, unidirectional incoming, unidirectional outgoing neighbors of user v, respectively.

information of edges in social graphs. Given a directed social graph $G = (V, E)$, we denote the set of unidirectional edges in the graph as E_{uni}, i.e., $E_{uni} = \{(u, v) \mid (u, v) \in E \text{ and } (v, u) \notin E\}$. The set of bidirectional edges is denoted as E_{bi}, i.e., $E_{bi} = \{(u, v) \mid u < v \text{ and } (u, v) \in E \text{ and } (v, u) \in E\}$. Notice that for a bidirectional edge (u, v), either (u, v) or (v, u) appears in E_{bi}, but not both. For each user u, we also have three types of neighbors, as illustrated in Figure 1. We denote by $\mathcal{N}_i(u), \mathcal{N}_o(u), \mathcal{N}_b(u)$ the set of unidirectional incoming, unidirectional outgoing, bidirectional neighbors of a user u. The adjacency matrices of the social graph formulated by the three types of neighbors are denoted as A_i, A_o, A_b, respectively.

Since different directions (follow / follower / reciprocal) of following relations capture different aspects of a user's behaviors, we should treat the three types of neighbors separately when performing graph convolution. Instead of using the original forward propagation rule of GCN on a single directed graph $H^{(l+1)} = \sigma(D^{-1}AH^{(l)}W^{(l)})$ in (Schlichtkrull et al. 2018), we assign separate weight matrices for each different type of neighbors. Hence we have the following layer-wise propagation rule:

$$\begin{aligned} H^{(l+1)} = \sigma(&D_i^{-1}A_iH^{(l)}W_i^{(l)} + D_o^{-1}A_oH^{(l)}W_o^{(l)} \\ &+ \tilde{D}_b^{-\frac{1}{2}}\tilde{A}_b\tilde{D}_b^{-\frac{1}{2}}H^{(l)}W_b^{(l)}) \end{aligned} \tag{1}$$

Here $\sigma(\cdot)$ denotes an activation function. D_i, D_o are the degree matrices of A_i, A_o (diagonal matrices of row sums). $\tilde{A}_b = A_b + I_N$ is the adjacency matrix of bidirectional relations with added self-connections, and $\tilde{D}_b$ is its degree matrix. The normalization trick $\tilde{D}_b^{-\frac{1}{2}}\tilde{A}_b\tilde{D}_b^{-\frac{1}{2}}$ in (Kipf and Welling 2016) is used for bidirectional relations since A_b is symmetric. The feature matrix of users forms the input to the first GCN layer as $H^{(0)} = X$. In this paper, we use bag-of-words (BoW) features extracted from each user's tweets. We will show the importance of assigning different weights for different types of neighbors in the experiments section.

Modeling Intuitions using Markov Random Fields

The above GCN model detects spammers through layer-wise neighbor message-passing. They learn how to aggregate information from each type of neighbor using weight matrices implicitly. However, there are several explicit natural patterns entailed in user following relations that we can take advantage of to enhance our GCN model further. Different types of neighbors a user would have different impacts on a user's identity. Concretely, we have the following intuitions of pairwise influences in the social network:

- **Intuition I**: Bidirectional neighbors of a user u tend to have the same label as u. This property is known as the homophily of social networks.
- **Intuition II**: If a user u has a lot of unidirectional incoming neighbors (i.e., u being followed by many users), u tend to be a legitimate user.
- **Intuition III**: If a user u has a lot of unidirectional outgoing neighbors (i.e., u follows many users), u tend to be a spammer.

We now capture the three intuitions using a pairwise Markov Random Field (pMRF) which models the joint probability distribution of all users' identities. We associate a binary random variable x_v for each user v where $x_v = 1$ and $x_v = 0$ denote the user is a spammer or a legitimate user, respectively. We denote x_V as the set of label assignments for all users. A pMRF can be formulated in this Gibbs distribution form: $P(x_V) = \frac{1}{Z}\exp\left(-E(x_V)\right)$ (here Z is a normalizing constant). The energy function E consists of unary potentials $\sum_v \phi(x_v)$ and pairwise potentials $\sum_{u,v} \varphi(x_u, x_v)$. Note that the lower the energy E (or potentials $\phi(x_v), \varphi(x_u, x_v)$) is, the higher the probability $P(x_V)$ becomes. By incorporating the three intuitions, we can obtain the following energy function:

$$\begin{aligned} E(x_V) = &\textstyle\sum_{v \in V} \phi_v(x_v) + \sum_{(u,v) \in E_{bi}} \varphi_{bi}(x_u, x_v) \\ &+ \textstyle\sum_{(u,v) \in E_{uni}} \varphi_{uni}(x_u, x_v) \end{aligned} \tag{2}$$

where unary potentials $\phi_v(x_v)$ measure the prior probabilities of label assignments, we let $\phi_v(x_v) = -\log p_v(x_v)$. $p_v(x_v)$ is the output probability that user v has label x_v from GCN. For pairwise potentials, $\varphi_{bi}(x_u, x_v) = -w$ when x_u and x_v are the same (both spammers or both legitimate users), w' otherwise. $\varphi_{uni}(x_u, x_v) = -w$ when $x_u = 1$ or $x_v = 0$, $\varphi_{uni}(x_u, x_v) = w'$ only if $x_u = 0$ and $x_v = 1$. Here $w, w' \geq 0$ are two learnable parameters measuring homophily and heterophily strength. The bidirectional pairwise potentials capture the first intuition: two users u, v connected by a bidirectional edge $u \leftrightarrow v$ tend to have the same label (they assign a higher probability if u, v have the same label, while penalize otherwise); the unidirectional pairwise potentials capture the second and third ones: for a unidirectional edge $u \rightarrow v$ connects two users u, v, u tend to be a spammer and v tend to be a legitimate user.

MRFasRNN

In this section, we introduce the computation of the MRF given by Eq. 2 through mean-field approximation and how we can formulate it as a RNN.

The exact posterior distribution of the MRF $P(x_V) = \frac{1}{Z}\exp\left(-E(x_V)\right)$ is infeasible to evaluate, so we use the mean-field theory to conduct approximate inference. We replace $P(x_V)$ with a factorizable distribution $Q(x_V) = \prod_{v \in V} Q_v(x_v)$ and approximate by minimizing the KL-divergence between the two distributions: $D(Q \parallel P) =$

$\mathbb{E}_{x_V\sim Q}[\log Q(x_V)] - \mathbb{E}_{x_V\sim Q}[\log P(x_V)]$. By substituting $P(x_V)$ and $Q(x_V)$, we get:

$$\begin{aligned} D(Q \parallel P) = {} & \mathbb{E}_{x_V\sim Q}[E(x_V)] \\ & + \textstyle\sum_{v\in V} \mathbb{E}_{x_v\sim Q_v}[\log Q_v(x_v)] + \log Z \end{aligned} \tag{3}$$

We define a Lagrangian composed of all terms involving $Q_v(x_v)$ in $D(Q \parallel P)$:

$$\begin{aligned} L_v(Q) = {} & \mathbb{E}_{x_V\sim Q}[E(x_V)] + Q_v(x_v)\log Q_v(x_v) \\ & + \lambda(\textstyle\sum_{x_v} Q_v(x_v) - 1) \end{aligned} \tag{4}$$

Here the term involving Lagrange multiplier λ assures that Q_v is a proper probability distribution. Now we take derivatives of Eq. 4 with respect to $Q_v(x_v)$:

$$\begin{aligned} \frac{\partial L_v(Q)}{\partial Q_v(x_v)} = {} & \phi_v(x_v) + \textstyle\sum_{u\in\mathcal{N}_b(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{bi}(x_u, x_v)] \\ & + \textstyle\sum_{u\in\mathcal{N}_i(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{uni}(x_u, x_v)] \\ & + \textstyle\sum_{u\in\mathcal{N}_o(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{uni}(x_v, x_u)] \\ & + \log Q_v(x_v) + 1 + \lambda \end{aligned} \tag{5}$$

By setting the derivative to 0, reorder the terms, we get:

$$\begin{aligned} \log Q_v(x_v) = {} & -\phi_v(x_v) - \textstyle\sum_{u\in\mathcal{N}_b(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{bi}(x_u, x_v)] \\ & - \textstyle\sum_{u\in\mathcal{N}_i(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{uni}(x_u, x_v)] \\ & - \textstyle\sum_{u\in\mathcal{N}_o(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{uni}(x_v, x_u)] - (1+\lambda) \end{aligned} \tag{6}$$

Taking exponent of both sides, and absorb the constant term $\lambda + 1$ into a normalizing constant z, we have:

$$\begin{aligned} Q_v(x_v) = {} & \frac{1}{z}\exp\{-\phi_v(x_v) \\ & - \textstyle\sum_{u\in\mathcal{N}_b(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{bi}(x_u, x_v)] \\ & - \textstyle\sum_{u\in\mathcal{N}_i(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{uni}(x_u, x_v)] \\ & - \textstyle\sum_{u\in\mathcal{N}_o(v)} \mathbb{E}_{x_u\sim Q_u}[\varphi_{uni}(x_v, x_u)]\} \end{aligned} \tag{7}$$

We now write Eq. 7 in the matrix form, where each row of Q corresponds to a user, and the two columns correspond to identity labels (spammer and legitimate user). This conversion to the matrix form can be achieved by decomposing the computation of Eq. 7 into three steps: message passing, unary potentials addition, and normalization. The message passing step consists of that between users, and the message passing between identity labels. Both can be implemented using matrix multiplication. More details on this conversion can be found in the supplementary material. Then we obtain the following update rule which can be implemented using sparse-dense matrix multiplication (note that here we use $\phi_v(x_v) = -\log p_v(x_v)$):

$$\begin{aligned} Q = {} & softmax(\log H^{(K)} - A_i Q \begin{bmatrix} -w & w' \\ -w & -w \end{bmatrix} \\ & - A_o Q \begin{bmatrix} -w & -w \\ w' & -w \end{bmatrix} - A_b Q \begin{bmatrix} -w & w' \\ w' & -w \end{bmatrix}) \end{aligned} \tag{8}$$

Here $H^{(K)}$ is the softmax output from the last layer of GCN with the same shape as Q, which is the predicted probabilities of user identities. We notice from Eq. 8 that the computation of Q relies on Q itself; hence iterative computation is required. As RNNs compute the outputs of each time step based on the results from the previous time step, we use them as the building block to conduct this iterative computation using a fixed number of steps. The difference between our model and the RNNs used in natural language tasks is that there is no input in our model, only the cell state describing posterior probabilities Q, which is initialized with GCN outputs $H^{(K)}$. Using this RNN framework enables us to implement the iterative computation of Eq.8 through multi-step inference. We notice that multi-step inference by RNN is essential for convergence of MRF posteriors, as we will demonstrate in the experiments section.

Algorithm 1 Forward propagation of GCNwithMRF

Input: Adjacency matrix A of the directed social network; input features matrix X; GCN depth K; # of MRF inference steps T; weight matrices $W_i^{(l)}, W_o^{(l)}, W_b^{(l)}, \forall l \in \{1, \ldots, K\}$; MRF weights w, w'; non-linearity σ

Output: Predicted probability matrix Q, where the first column represents the probability of being a spammer, the second column represents that of being a legitimate user.

```
Construct $A_i, A_o, A_b$ from $A$.
$H^{(0)} \leftarrow X$
for $l = 1 \ldots K-1$ do
    Compute $H^{(l)}$ from $H^{(l-1)}$ using Eq. 1 with non-linearity $\sigma$.
end for
Compute $H^{(K)}$ from $H^{(K-1)}$ using Eq. 1 with softmax nonlinearity.
$Q \leftarrow H^{(K)}$    ▷ Initialize posterior probabilities of the MRF layer with GCN outputs
for $i = 1 \ldots T$ do
    Update $Q$ according to Eq. 8.
end for
return $Q$
```

The MRF layer is then stacked on top of GCN, and the whole model can be trained in an end-to-end manner. The cross-entropy loss is used for training: $\mathcal{L} = -\sum_{u\in\mathcal{Y}_L}\sum_{l\in\{0,1\}} Y_{ul}\ln Q_{ul}$, where $\mathcal{Y}_L$ is the set of labeled nodes and Y is the one-hot label matrix. We denote our model as GCNwithMRF, and the complete forward propagation procedure of GCNwithMRF is described in Algorithm 1. In training time, we infer for 5 steps (i.e., $T = 5$) and during testing, we compute for 10 steps (i.e., $T = 10$). The proposed model has a computational complexity linear in the number of edges in the social network.

Experiments

In this section, we evaluate our proposed method with the aim to respond to the following questions:

1. Does our proposed GCNwithMRF method outperform current state-of-the-art spammer detection approaches? How robust is our method?
2. Will treating the three types of neighbors in the social network separately in the GCN layers help im-

	TwitterSH	1KS-10KN
# of spammers	3,579	487
# of legitimate users	5,050	8,770
# of social relations	609,746	3,024,744
# of tweets	1,265,860	950,342

Table 1: Statistics of processed datasets.

prove the performance of our method? Is jointly training GCN and MRF layers indispensable for satisfying results?

3. How can the MRF layer correct the incorrect predictions generated by GCN? How important is the multi-step inference in the MRF layer to ensure convergence and obtain better performance?
4. How well do the bag-of-words features contribute to the overall performance of our model?

Experiment Setup

Datasets. We use two public datasets to evaluate our method: *Twitter Social Honeypot Dataset (TwitterSH)* (Lee, Eoff, and Caverlee 2011) and *Twitter 1KS-10KN Dataset (1KS-10KN)* (Yang et al. 2012). The two datasets contain labeled spammers and legitimate users collected on Twitter, along with their corresponding tweets. The 1KS-10KN dataset also contains the social network information, but the TwitterSH dataset does not. Hence we use an external Twitter social graph dataset (Kwak et al. 2010) to extract social relations of the users in TwitterSH dataset. We filtered all non-English tweets and pre-process each tweet by removing URLs, numbers, mentions, stop words, etc. Then lemmatization is performed. Users with less than two tweets or two social relations are filtered. Table 1 summarizes the processed datasets. We notice that the 1KS-10KN dataset is a highly imbalanced dataset, with a spammers to legitimate users ratio of $1 : 18$. We divide each dataset into three part: a 30% semi-supervised training set, a 10% validation set, and the remaining 60% as the test set.

Evaluation metrics. We use Area Under the Precision-Recall Curve (PRAUC) to evaluate the performance of spammer detection systems. We choose PRAUC instead of ROCAUC because precision-recall curves are better in evaluating performance with class imbalance datasets, while ROC curves can be deceptive in this circumstance (Davis and Goadrich 2006). For the TwitterSH dataset, we also use accuracy as an additional metric. However, we do not use accuracy on the 1KS-10KN dataset since all methods achieve $\approx 95\%$ accuracy on this highly imbalanced dataset.

Compared baselines. We compare our proposed GCNwithMRF method with the following baselines, including state-of-the-art spammer detection approaches and variants of GCNwithMRF:

- **SMSFR** (Zhu et al. 2012) is a supervised social spammer detection method based on matrix factorization to learn latent user and text features representations. Undirected social graphs are used as a regularization mechanism.
- **OSSD** (Hu, Tang, and Liu 2014) further extends SMFSR to consider directed social networks.
- **SSDMV** (Li et al. 2018) is a semi-supervised spammer detection model built on an autoencoder framework. Multi-view data are fused to get joint representations of users using correlated ladder networks.
- **SybilSCAR** (Wang, Zhang, and Gong 2017) is a structure based semi-supervised spammer detection method using a pairwise Markov Random Field, based on the intuition that neighbor users tend to have the same labels.
- **GANG** (Wang, Gong, and Fu 2017) extends SybilSCAR to consider directed social graph composed of three types of neighbors.
- **RF** is a random forest baseline using graph-based features introduced in (Fu et al. 2017).
- **GCN** is a variant of our GCNwithMRF model which excludes the MRF layer.
- **GCNsg** is a variant of the GCN model. It operates on a single directed social network instead of modeling three types of neighbors separately.
- **GCN+GANG** is a refined version of GANG which uses the outputs of our GCN model as node priors.

Parameter settings. As previously mentioned, we use bag-of-words features for our models and the matrix factorization based models. We select the most frequent 5000 words and 1000 hashtags to construct the vocabulary tables. The obtained text and hashtag BoW features are then concatenated to form the input features for each user.

We follow the two-layer GCN setup in (Kipf and Welling 2016). We optimize hyperparameters on the validation set, the settings after hyperparameter tuning are reported. We train all models using Adam optimizer with a learning rate of 0.01 for a maximum of 200 epochs, early stopping with a window of 10 epochs is adopted. We also use dropout with a ratio of 0.5. The number of hidden units is set to 32 on the TwitterSH dataset and 64 on the 1KS-10KN dataset. For the compared state-of-the-art methods, we use the parameter setups in the original papers.

Comparison with Baselines

In this part, we answer the first and second questions. We evaluate the performance of our model and the baselines listed above on the two datasets using $3\% \sim 100\%$ of data from the 30% semi-supervised training set (so the models will only see $0.9\% \sim 30\%$ data of the entire dataset). The results are shown in Table 2 and Table 3.

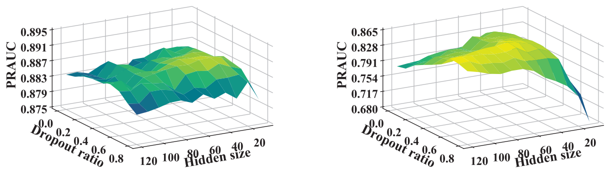
From the results, we can see that GCNwithMRF consistently outperforms all the baselines compared with the increasing training data used, especially on the extremely class-imbalanced 1KS-10KN dataset where other methods yield significantly lower PRAUC values. The 1KS-10KN is also a more realistic case where the majority of users on social networks are legitimate users.

	3%		5%		10%		20%		60%		100%	
	ACC	PRAUC	ACC	PRAUC	ACC	PRAUC	ACC	PRAUC	ACC	PRAUC	ACC	PRAUC
SMFSR	0.613	0.533	0.645	0.559	0.661	0.584	0.678	0.620	0.695	0.640	0.701	0.646
OSSD	0.670	0.599	0.683	0.616	0.691	0.624	0.696	0.647	0.710	0.690	0.717	0.697
SSDMV	0.737	0.703	0.743	0.724	0.746	0.730	0.749	0.744	0.790	0.805	0.796	0.813
SybilSCAR	0.417	0.537	0.420	0.537	0.425	0.538	0.434	0.544	0.455	0.556	0.479	0.570
GANG	0.433	0.733	0.440	0.735	0.454	0.735	0.477	0.736	0.520	0.736	0.545	0.738
RF	0.762	0.745	0.766	0.757	0.772	0.760	0.774	0.766	0.776	0.767	0.779	0.771
GCNsg	0.737	0.805	0.763	0.811	0.788	0.826	0.780	0.834	0.784	0.834	0.786	0.835
GCN	0.777	0.821	0.787	0.829	0.805	0.845	0.814	0.850	0.820	0.859	0.825	0.868
GCN+GANG	0.779	0.669	0.791	0.675	0.810	0.700	0.822	0.730	0.828	0.742	0.834	0.747
GCNwithMRF	**0.792**	**0.849**	**0.805**	**0.860**	**0.820**	**0.876**	**0.824**	**0.880**	**0.833**	**0.887**	**0.839**	**0.890**

Table 2: Results on the TwitterSH dataset.

	3%	5%	10%	20%	60%	100%
SMFSR	0.124	0.131	0.140	0.147	0.154	0.162
OSSD	0.141	0.149	0.156	0.164	0.173	0.200
SSDMV	0.191	0.202	0.219	0.238	0.267	0.278
SybilSCAR	0.053	0.053	0.053	0.053	0.053	0.053
GANG	0.076	0.076	0.076	0.076	0.077	0.078
RF	0.164	0.249	0.293	0.377	0.573	0.653
GCNsg	0.595	0.602	0.617	0.629	0.650	0.706
GCN	0.665	0.677	0.695	0.726	0.811	0.833
GCN+GANG	0.114	0.136	0.170	0.177	0.226	0.245
GCNwithMRF	**0.676**	**0.687**	**0.713**	**0.741**	**0.839**	**0.865**

Table 3: Results on the 1KS-10KN dataset.



(a) TwitterSH

(b) 1KS-10KN

Figure 2: Parameter sensitivity of GCNwithMRF.

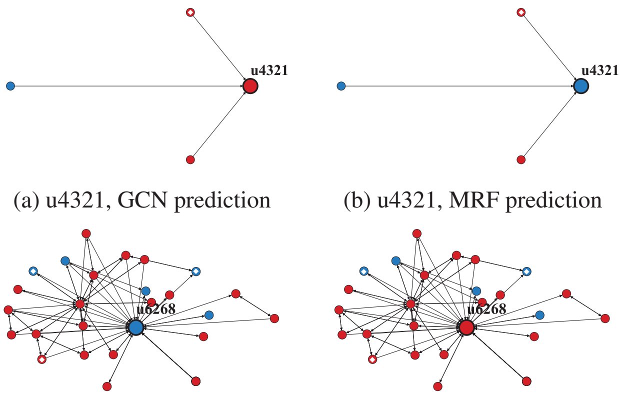
Comparison with state-of-the-art methods. The matrix factorization based models SMFSR and OSSD are fully supervised models, which can only exploit labeled part of the social networks, hence perform poorly in the real-world semi-supervised setting. Still, we can observe that OSSD is relatively better than SMFSR since it considers directed social networks. SSDMV takes node2vec and doc2vec embeddings as input features, which cannot explicitly model the interactions in text and social networks, hence its performance is limited. Furthermore, since SSDMV can only operate on undirected social graphs composed of reciprocal relations, its performance is severely restricted on the 1KS-10KN dataset which contains far more unidirectional edges (40.43% edges are unidirectional) than the TwitterSH dataset (only 12.66%). The MRF based methods SbyilSCAR and GANG perform poorly on both datasets. On the TwitterSH datasets they produce low accuracy values since the social network is quite sparse, the posterior probabilities of the majority of nodes remain unaffected in the belief propagation process (20.5% users have the same prior and posterior probabilities of 0.5 given 100% training data using SybilSCAR). On the 1KS-10KN dataset, most users' posterior probabilities of being legitimate users collapse to 1.0 (97.8% users given 100% training data using GANG). Clearly, these two methods are not suitable for such highly imbalanced datasets.

Comparison with variants of GCNwithMRF. By comparing GCNsg and GCN, We observe that treating three types of neighbors separately in GCN is essential for better performance as each type of neighbors implies different kind of information and different weight matrices should be learned. We find that the MRF layer which models the three intuitions indeed helps improve the performance of GCN, by comparing GCN and GCNwithMRF. We also notice that jointly training GCN and MRF layers is crucial for better performance. If we simply feed GCN's outputs as priors to GANG, the accuracy will slightly increase, but the PRAUC will tremendously decrease. This is because the posterior probabilities generated by GANG fail to capture the confidence of GANG predictions, as 96.2% users' posterior probabilities collapse to 0.0 or 1.0 (using 100% training data on the TwitterSH dataset).

Parameter sensitivity. To evaluate the robustness of our model, we vary the number of neurons in the hidden layer and the dropout ratio. All models here are trained using 100% training data. The results are shown in Figure 2.

The Effectiveness of the MRF Layer

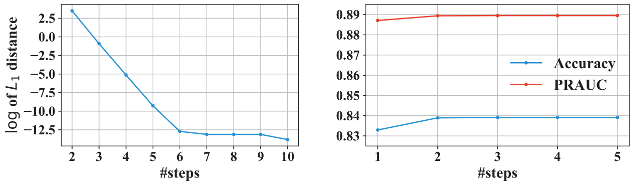
To further demonstrate the refining effect of the MRF layer, we show two examples from the TwitterSH dataset, where the incorrect predictions made by GCN are corrected in the MRF layer in Figure 3. The first example is user 4321, where the GCN layers predict this user as a spammer. However, all neighbors of u4321 are unidirectional incoming ones; this indicates u4321 tend to be a legitimate user according to our second intuition, and the MRF layer successfully remedies this mistake. This incorrect prediction of GCN is attributed to the fact that one of the u4321's neighbors is labeled, and this labeled spammer directly affects GCN's predicted label of u4321. The second example is user 6268, where GCN



(a) u4321, GCN prediction (b) u4321, MRF prediction

(c) u6268, GCN prediction (d) u6268, MRF prediction

Figure 3: Examples of users which are wrongly classified by GCN but are corrected in the MRF layer, nodes with the white diamond symbol are labeled (which GCN correctly predicted). Each node's color denotes its predicted label: red nodes are spammers, blue ones are legitimate users.

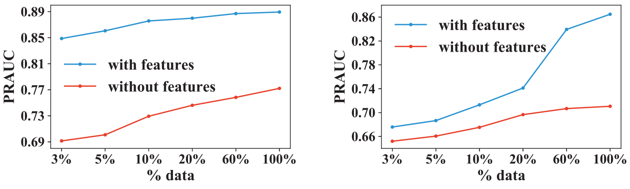


(a) Convergence (b) Performance on test set

Figure 4: (a) shows the convergence of output probabilities by varying # of inference steps during testing; (b) shows the performance on test set by varying #steps during training.

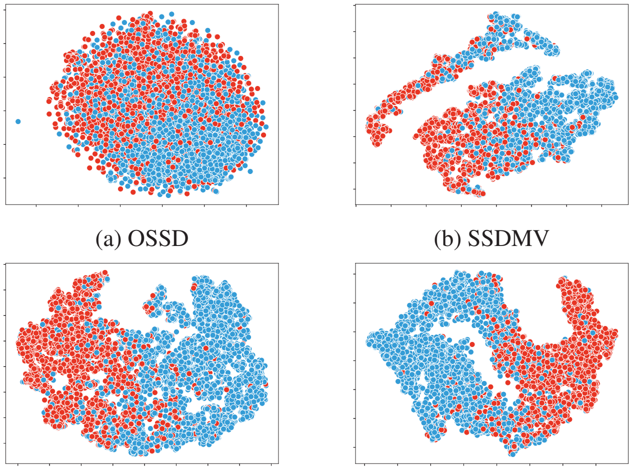
incorrectly predicts that it is a legitimate user due to the misguidance of the two labeled legitimate neighbors. However, u6268 has a lot of bidirectional spammers neighbors (predicted by GCN). According to our first intuition, a user and its bidirectional neighbors tend to have the same label; the MRF layer again corrects this mistake by predicting u6268 as a spammer.

The importance of multi-step inference. We conduct multi-step inference by formulating MRF as a RNN, in order for the posterior probabilities to converge, as opposed to the one-step computation in (Jin et al. 2019). Here we illustrate the importance of this multi-step inference to convergence and final performance. We first keep the number of inference steps during training to 5 and vary #steps during testing. We then compute the log of L_1 distance between the output probabilities of two consecutive steps, as shown in Figure 4a. We also keep #steps during testing to 10 and change #steps during training. We show the performance on the test set versus #steps in Figure 4b. We can see that multi-step inference is crucial for both convergence and performance. There is a significant gap in performance between one-step and two-step inference, though the differences become slight when we compute for more than two steps. Here all models are trained using 100% training data on the TwitterSH dataset.



(a) TwitterSH (b) 1KS-10KN

Figure 5: Performance comparison of GCNwithMRF model with/without BoW features.



(a) OSSD (b) SSDMV

(c) GCN (d) GCNwithMRF

Figure 6: The t-SNE visualization of latent representations.

The Contribution of BoW Features

Our GCN-based models obtain superior performance because they can combine both graph structures and external semantic information in the text domain. Here we investigate the role that BoW features play in our GCNwithMRF model. We implement a featureless model where we simply feed an identity matrix as input features to GCNwithMRF. The results are shown in Figure 5. We can observe that the BoW features are vital to the high performance of GCNwithMRF.

Visualization

To intuitively demonstrate the quality of the user embeddings, we use the t-SNE tool (Maaten and Hinton 2008) to visualize the learned latent user representations of different models on the TwitterSH dataset. All models are trained using 100% training data. For GCN and GCNwithMRF, we use the hidden layer to perform the t-SNE visualization, while for SSDMV, we use concatenated final representations of the encoders of each view. The results are shown in Figure 6. We can clearly observe that GCNwithMRF generates notably better embeddings that can separate legitimate users and spammers well; while using only GCN results in slightly worse embeddings. OSSD, however, produce embeddings that fail to differentiate spammers and legitimate users. We have also conducted K-means clustering (K=2) on the results of t-SNE and measured the adjusted Rand index (ARI). We obtained the following results: OSSD: 0.1465, SSDMV:0.2857, GCN:0.4132, GCNwithMRF: 0.5173. The results have verified our conclusions.

Conclusion

In this paper, we developed a model incorporating both GCN and MRF, which operate on directed social graphs for semi-supervised social spammer detection. Extensive experiments demonstrate the superiority of our method and the refining effect of the MRF layer. We also find that multi-step reasoning in the MRF layer is essential to ensure convergence. However, we simply use BoW features. We may incorporate state-of-the-art language models in future work.

Acknowledgments

The work was partially supported by grants from the National Natural Science Foundation of China (Grant No. 61976198, U1605251, 61832017 and 61972125).

References

Blanzieri, E., and Bryl, A. 2008. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review* 29(1):63–92.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*.

Davis, J., and Goadrich, M. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*.

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 29*.

Fu, H.; Xie, X.; Rui, Y.; Gong, N. Z.; Sun, G.; and Chen, E. 2017. Robust spammer detection in microblogs: Leveraging user carefulness. *ACM Trans. Intell. Syst. Technol.* 8(6):83:1–83:31.

Gyongyi, Z., and Garcia-Molina, H. 2005. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*.

Hu, X.; Tang, J.; Zhang, Y.; and Liu, H. 2013. Social spammer detection in microblogging. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Hu, X.; Tang, J.; Gao, H.; and Liu, H. 2014. Social spammer detection with sentiment information. In *2014 IEEE International Conference on Data Mining*.

Hu, X.; Tang, J.; and Liu, H. 2014. Online social spammer detection. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Jin, D.; Liu, Z.; Li, W.; He, D.; and Zhang, W. 2019. Graph convolutional networks meet markov random fields: Semi-supervised community detection in attribute networks. In *Thirty-Third AAAI Conference on Artifical Intelligence*.

Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*.

Lee, K.; Eoff, B. D.; and Caverlee, J. 2011. Seven months with the devils: A long-term study of content polluters on twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*.

Li, C.; Wang, S.; He, L.; Philip, S. Y.; Liang, Y.; and Li, Z. 2018. Ssdmv: Semi-supervised deep social spammer detection by multi-view data fusion. In *2018 IEEE International Conference on Data Mining)*.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* 9(Nov):2579–2605.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*.

Shen, H.; Ma, F.; Zhang, X.; Zong, L.; Liu, X.; and Liang, W. 2017. Discovering social spammers from multiple views. *Neurocomput.* 225(C):49–57.

Singh, M.; Bansal, D.; and Sofat, S. 2016. Behavioral analysis and classification of spammers distributing pornographic content in social media. *Social Network Analysis and Mining* 6(1):41.

VanDam, C., and Tan, P.-N. 2016. Detecting hashtag hijacking from twitter. In *Proceedings of the 8th ACM Conference on Web Science*.

Varol, O.; Ferrara, E.; Davis, C. A.; Menczer, F.; and Flammini, A. 2017. Online human-bot interactions: Detection, estimation, and characterization. In *Eleventh International AAAI Conference on Web and Social Media*.

Wang, B.; Gong, N. Z.; and Fu, H. 2017. Gang: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs. In *2017 IEEE International Conference on Data Mining*.

Wang, H.; Lian, D.; and Ge, Y. 2019. Binarized collaborative filtering with distilling graph convolutional networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

Wang, B.; Zhang, L.; and Gong, N. Z. 2017. Sybilscar: Sybil detection in online social networks via local rule based propagation. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*.

Webb, S.; Caverlee, J.; and Pu, C. 2008. Social honeypots: Making friends with a spammer near you. In *Proceedings of the Fifth Conference on Email and Anti-Spam*.

Wu, Y.; Lian, D.; Jin, S.; and Chen, E. 2019. Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

Yang, C.; Harkreader, R.; Zhang, J.; Shin, S.; and Gu, G. 2012. Analyzing spammer's social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st International Conference on World Wide Web*.

Zhu, Y.; Wang, X.; Zhong, E.; Liu, N. N.; Li, H.; and Yang, Q. 2012. Discovering spammers in social networks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.