Pattern Recognition 125 (2022) 108512

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

User-based network embedding for opinion spammer detection

Ziyang Wang[a], Wei Wei[a,a,*], Xian-Ling Mao[b], Guibing Guo[c], Pan Zhou[d], Sheng Jiang[a]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, China
[b] School of Computer Science and Technology, Beijing Institute of Technology, China
[c] Software College, Northeastern University, China
[d] School of Cyber Science and Engineering, Huazhong University of Science and Technology, China

ARTICLE INFO

Article history:
Received 18 March 2021
Revised 21 December 2021
Accepted 22 December 2021
Available online 29 December 2021

Keywords:
Spam detection
Collective spammer
Network embedding
Signed network

ABSTRACT

Due to the huge commercial interests behind online reviews, a tremendous amount of spammers manufacture spam reviews for product reputation manipulation. To further enhance the influence of spam reviews, spammers often collaboratively post spam reviews within a short period of time, the activities of whom are called *collective opinion spam campaign*. The goals and members of the spam campaign activities change frequently, and some spammers also imitate normal purchases to conceal the identity, which makes the spammer detection challenging. In this paper, we propose an unsupervised network embedding-based approach to jointly exploiting different types of relations, *e.g.,* direct common behavior relation, and indirect co-reviewed relation to effectively represent the relevances of users for detecting the collective opinion spammers. The average improvements of our method over the state-of-the-art solutions on dataset *AmazonCn* and *YelpHotel* are [14.09%,12.04%] and [16.25%,12.78%] in terms of AP and AUC, respectively.

1. Introduction

Online reviews are a valuable source of reference for decision-making, and thus the massive volume of spammers are attracted to post malicious generated reviews to promote/demote the target products. With popular crowdsourcing services, these spammers show great influence as they can easily propagate deceptive reviews for dominating the opinions of target products within a short period, and their activities are called *collective opinion spam campaign*. As such, it is highly valuable and desirable to develop an effective algorithm in accurately capturing collusion signals for collective opinion spammer detection. Although the opinion spam detection problem has attracted extensive attention and some proposed methods have greatly advanced the problem, they still easily fail due to the facts: *Firstly*, previous ***supervised***-based methods formulate it as a classification task, whose performance often heavily rely on the scale of labeled data [1–5]. Nevertheless, manual labeling spam reviews is extremely difficult and unreliable [6]. *Secondly*, there exist several attempts for the opinion spam detection problem in an ***unsupervised*** fashion: (i) Previous *individual spammer* detection methods focus on modeling highly-visible behaviors [7–9] or temporal factors [10–12] for detecting spammers. However, some spammers also make normal purchases to conceal the identity, and thus such features are not reliable for tackling the spammer detection problem. (ii) Many *spam group* detection methods employ co-behavior features [13,14] or clustering algorithm [15,16] to detect spam groups. However, they are based on the assumption that each user can only exist in one group, which is flawed as each user may participate in several different spam campaigns.

Recently, there exists several efforts that have been dedicated to research on spammer detection by modeling the user-user relations over behavior features [17–20]. These efforts usually focus on extracting discriminative features (*i.e.,* pointwise or pairwise features) from *direct* common behaviour relations (which are called direct relations). However, spammers are often required to balance workload within spam campaigns for evading detection, and thus it is insufficient to capture the collusion signals by modeling the direct relation alone. Intuitively, spammers are usually engaged repeatedly for different collective opinion spam campaigns and colluders are likely to share more common neighbors within the same spam-campaigns (even without direct collusion relation), and thus the combination of the *direct* and *indirect* collusion relation (which refers to indirect associations with multiple steps, *e.g., k*-steps, along with the neighborhood structures of colluders, which are called *indirect* relation) is a relatively stationary collusion signal, which is not easily manually manipulated.

[*] Corresponding author.
E-mail address: weiw@hust.edu.cn (W. Wei).

https://doi.org/10.1016/j.patcog.2021.108512
0031-3203/© 2022 Elsevier Ltd. All rights reserved.

To this end, in this work we propose a new *unsupervised* network embedding-based approach to learning the user embeddings by jointly exploiting the different types of relations (*i.e., direct* relation and *indirect* relation) between pairwise users for *opinion spammer detection*. The main contributions of our proposed method can be summarized as follows: i) We are the first to jointly learning **direct relevance** and **indirect relevance** for collective spammers detection. Direct relevance is captured by the pairwise behavior features between two users and indirect relevance is learned based on a **k-step co-rating neighborhood proximity**. By combining direct relevance and indirect relevance, we can capture both direct collusion signals and potential collusion signals. ii) Based on the pairwise features, we propose a novel **user-based signed network**, which includes both positive edges and negative edges. Through the signed network, we can more accurately capture the relevance between pairwise users. Extensive experiments conducted on two real-world datasets (*i.e.,* Amazon_cn and YelpHotel) verify the effectiveness of combining direct relevance and indirect relevance, which also show the superiority of user-based signed network.

2. Related work

Spam review detection. Spam review detection methods aim to identify fake reviews manufactured by spammers. Jindal *et al.* [8] find unexpected rules to represent suspicious behaviors of reviews based on the *unusual* review patterns. Ott *et al.* [6] create the first ground truth dataset by employing crowd-sourcing through the Amazon Mechanical Tuck, and they utilize psychological and linguistic clues on identifying review spam. Harris [21] explore several different kinds of assessment methods to spot deceptive opinion spam. Li *et al.* [22] propose a bayesian generative approach to find the general difference between deceptive and truthful reviews. Wang *et al.* [4] learn the user embeddings generated by tensor decomposition for training a spam review classifier. Li *et al.* [23] focus on the cold-start problem by using user behavior representation. Additionally, You *et al.* [24] develop a unified deep learning architecture to tackle the cold-start problem in spam review detection. However, most of spam review detection methods are based on linguistic clue [25,26] or document-level features [3], which have been shown ineffective for spam detection problem [13,15]. In contrast, our proposed model focuses on modeling spam-campaign characteristics within a unified ***unsupervised*** framework to explore the direct/indirect ***user-user*** relations of users to capture the collusion signals. **Individual spammer detection**. Individual spammer detection methods aim to detect individual spammers who participated in opinion spam campaigns and post spam reviews. Lim *et al.* [7] first study the rating behavioral characteristics for review spammer detection. Li *et al.* [9] propose a co-training method based on the extracted review-based/reviewer-based features for identifying spammers. Kumar *et al.* [20] propose three quality metrics, they detect spammers through computing the fairness of a user, reliability of a rating and goodness of a product iteratively. Kaghazgaran *et al.* [27] pre-train a spammer classifier by modeling structurally similar users within a three-phase framework. However, it is difficult to accurately determine a spammer using only the behavior and text characteristics of a single user. Recently, there have been numerous attempts to detect spammers by modeling the user-user relations over behavior features. Xu [17] extends the Markov Random Walk model to obtain a ranking of reviewers spamicity scores by exploring multiple heterogeneous pairwise features from reviewers rating behaviors and linguistic patterns. Rayana *et al.* [18] utilize a loopy belief propagation model to infer spammers by extracting relational features, which is then extended by introducing active inference[28]. Xu *et al.* [19] propose a regularized matrix factorization model to obtain reviewer behavior embeddings.

Our proposed method aims to capture spammers who participated in *collective opinion spam campaign*, which belongs to individual spammer detection. Different from existing methods which only focus on *shallow relational* information while neglecting ***indirect*** relations, we jointly learning direct relevance and indirect relevance for capturing the collusion signal. **Spam group detection**. Spam group detection methods are designed to detect fraudulent groups, where the users in these groups are regarded as spammers. Mukherjee *et al.* [13] rank collusion spam groups over user-group-product relations for detection. Ye *et al.* [15] estimate the likelihood of products being spam campaign targets for inferring spammers. Liu *et al.* [14] mainly focus on graph topology and temporal information for detecting fraud groups. Zheng *et al.* [16] propose FraudNE to jointly learn the user embedding and item embedding in a bipartite network, and then use a clustering algorithm to detect fraudulent groups. Wang *et al.* [29] design a markov random field-based model and use loopy belief propagation based algorithm to detect spammer groups. However, these methods are based on the assumption that each user can only exist in one group, which has obvious flaws as each user may participate in several different spam campaigns. In addition, the clustering algorithm they use is prone to misclassify normal users. **Network Embedding**. Network embedding learning is a subfield of representation learning [30–32]. Network embedding-based approaches [33–37] aim to produce low-dimensional representations for nodes while encapsulating the structure of the network, which have been highlighted in computer version [38–40], natural language processing [41,42] and recommendation systems [43–45]. To transform the nodes from original network space into embedding space, some early work [46–48] obtain the leading eigenvectors as the network representations. More recently, Perozzi *et al.* [49] propose Deepwalk to combine random walk and skip-gram to learn network representations. Grover *et al.* [50] design a biased random walk procedure to efficiently explore diverse neighborhoods and encode nodes to a low-dimensional space. To extend the representational power, Mousavi *et al.* [51] extract local and global features from different scales of the given graph. Bahonar *et al.* [52] use diffusion wavelet embedding to discover the inherent node clusters within the graphs. Taking into account high-level latent structures, Shi *et al.* [53] propose diffusion-based network embedding. To study the problem of network embedding with completely-imbalanced labels, Wang *et al.* [54] learn discriminative embeddings by approximately guaranteeing both intra-class similarity and inter-class dissimilarity.

3. Problem statement and notations

Let $\mathbf{P} = \{p_j\}_{|\mathbf{P}|}$ be the set of items over a set of product categories $\mathbf{C} = \{c_i\}_{|\mathbf{C}|}$; $\mathbf{U} = \{u_i\}_{|\mathbf{U}|}$ and $\mathbf{X} = \{\mathbf{x}_{ij}\}_{|\mathbf{X}|}$ be the set of users and the records of their reviews, where $\mathbf{x}_{ij} \in \mathbf{X}$ denotes the reviews posted by u_i on product p_j, since a user u_i might post different reviews for the same product p_j owing to multiple purchases. As such, a 4-tuple (u_i, p_j, r, t) denotes the review generated by user u_i for product p_j with the rating r at time t. The rating and the time-stamp are denoted by x^r_{ijk} and x^t_{ijk} for simplicity, where k denotes user is k-th rating for the item j as the possibility of multiple reviews per user.

Then, given a set of users and the metadata of their posted reviews, *i.e.,* (u_i, p_j, r, t), the problem of *collective opinion spammer detection* is defined as identifying a set of spammers based on their participated ***spam campaigns***, namely, which is regarded as a task that requires a set of spammers to collectively post malicious opinions (*i.e.,* human-powered deceptive contents) on the

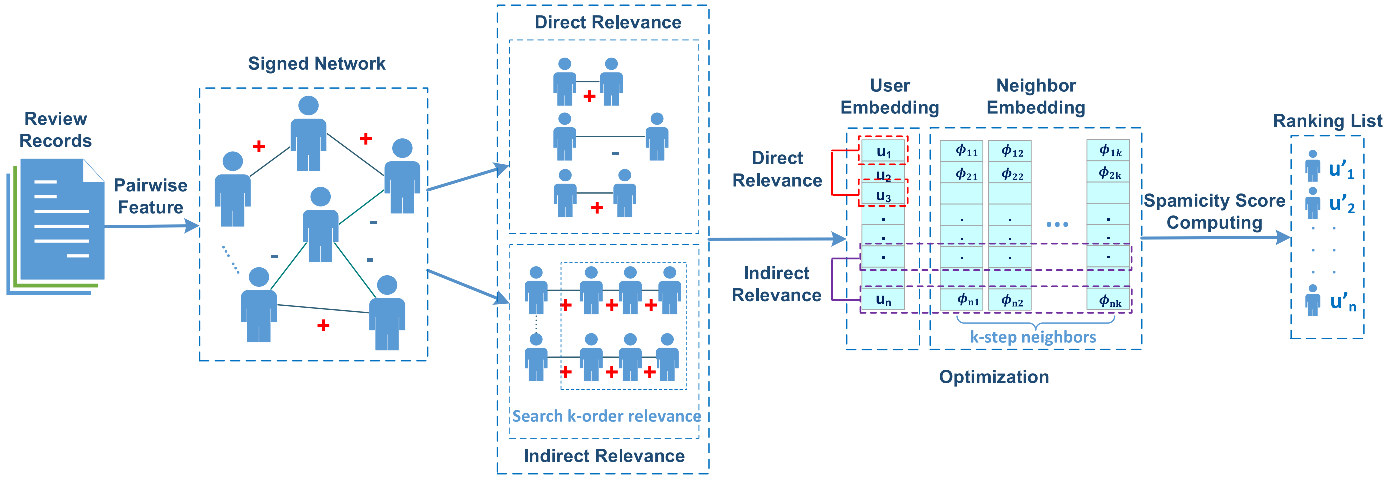


Fig. 1. Overview of proposed approach. First, a signed network is built based on the extraction of pairwise features from users' reviews, then we jointly optimize two types of relevances, *i.e., direct* relevance and *indirect* relevance for learning the user embedding in a low-dimensional space. Finally, the spamicity score of each user is calculated based on the learnt user embeddings for estimating the degree of being colluders.

target items. For clarity, frequently used abbreviations are summarized in Table 1.

In this section, we present our proposed **c**ollective **o**pinion **s**pammer **d**etection (COSD) approach within a unified architecture (shown in Fig. 1), the objective of which is to jointly combine *direct* and *indirect* neighborhood exploration for learning the embedding representation of each user for more accurately identifying spam reviewers. The rationale behind this is that *direct relevance embedding* controls the learning of user embeddings towards the pairwise users with strong intensity of the collusive characteristics, while *indirect relevance embedding* tends to make pairwise users sharing more commonly co-rating neighbors closer. Indeed, we are only interested in *direct relevance* embedding of users to identify spammers, however we also consider *indirect relevance* embedding in our framework because such two types of embeddings will reinforce each other mutually to make the relevant users close over *direct* and *indirect* relations of users (the details will be elaborated later on). Next, we detail how to model the *direct relevance embedding* and *indirect relevance embedding* for any pairwise users.

3.1. Modeling spam-campaign characteristics

In this section, we mainly focus on how to characterize spam-campaigns on different dimensions to build the user-user weighted matrix **W**, via capturing the intersections on the co-rated item sequences for each pairwise users. Many previous works are proposed for this task [4,13,17,19]. By following the work [17], we adopt four different types of heterogeneous pairwise features as follows. Note we do not consider *linguistic*-based features from reviews.

- **Product Rating Proximity (PR)**, it measures the intensity of spammers' agreements for pairwise users. Generally, spammers within a spam-campaign are instructed to post similar opinions with consistent ratings on target items. As such, given a pair of users (u_i, u_j), we measure the intensity of spammers' agreements as follows,

$$\psi_{PR}(i,j) = \frac{2}{1+\exp(\Gamma_{ij}^{p,r})} \tag{1}$$

where $\Gamma_{ij}^{p,r}$ denotes the average rating deviation of pairwise users (u_i, u_j) over their commonly reviewed items,

$$\Gamma_{ij}^{p,r} = \frac{1}{|\mathbf{P}_i \cap \mathbf{P}_j|} \sum_{p_k \in \mathbf{P}_i \cap \mathbf{P}_j} \left(\left| \frac{\sum_q x_{ikq}^r}{|x_{ik}|} - \frac{\sum_q x_{jkq}^r}{|x_{jk}|} \right| \right),$$

where $\mathbf{P}_{i(j)}$ refers to the set of items reviewed by user $u_{i(j)}$. Eq. (1) favors to find pairwise spammers who have more consistent ratings on co-rated items, especially $\psi_{PR}(i,j) = 1$ when $\Gamma_{ij}^{p,r} = 0$.

- **Product Time Proximity (PT)**, it captures the temporal consistency for pairwise users. Intuitively, colluders are often asked to complete the task within a short time frame (*e.g.,* less than a week) for maximizing the influence, and thus the temporal traces of their reviews tend to be more intensive than normal users'. Hence, we use PT to capture the temporal consistency of pairwise users (u_i, u_j),

$$\psi_{PT}(i,j) = \frac{1}{C + \gamma \Gamma_{ij}^{p,t}} \tag{2}$$

where $\Gamma_{ij}^{p,t}$ denotes the average time deviation of pairwise user (u_i, u_j) over their co-reviewed items,

$$\Gamma_{ij}^{p,t} = \frac{1}{|\mathbf{P}_i \cap \mathbf{P}_j|} \sum_{p_k \in \mathbf{P}_i \cap \mathbf{P}_j} \left(\left| \frac{\sum_q x_{ikq}^t}{|x_{ik}|} - \frac{\sum_q x_{jkq}^t}{|x_{jk}|} \right| \right);$$

C and γ denote the smoothing factor and the trade-off parameter, which are empirically set at 1 and 20, respectively.

- **Category Rating Proximity (CR)**, it measures the average category rating deviation between pairwise users. Intuitively, spammers from different spam-campaigns might have different rating distributions over reviewed categories, and thus the higher intersections of category rating distributions of two users are consistent, the more likely these two users are colluders. Hence, CR is computed based on the average category rating deviation $\Gamma_{ij}^{c,r}$ between pairwise users,

$$\psi_{CR}(i,j) = \frac{2}{1+\exp(\Gamma_{ij}^{c,r})} \tag{3}$$

where

$$\Gamma_{ij}^{c,r} = \frac{1}{|\mathbf{C}_i \cap \mathbf{C}_j|} \sum_{c_k \in \mathbf{C}_i \cap \mathbf{C}_j} \left(\left| \overline{c_{ik}^r} - \overline{c_{jk}^r} \right| \right),$$

and $\mathbf{C}_{i(j)}$ is the set of categories reviewed by user $u_{i(j)}$; $\overline{c_{ik}^r}$ denotes the average rating of user u_i in the k-th category, which is calculated by

$$\overline{c_{ik}^r} = \frac{1}{|c_k|} \sum_{p_j \in c_k} \frac{1}{|\mathbf{x}_{ij}|} \sum_q x_{ijq}^r.$$

- **Category Time Proximity (CT)**, it measures the consistency of time distributions between pairwise users over co-reviewed categories. Analogous to CR, it is also a strong indicator to measure the degree of the collusive characteristics of pairwise users, which is estimated by,

$$\psi_{CT}(i,j) = \frac{1}{C+\gamma\Gamma_{ij}^{c,t}} \tag{4}$$

where C and γ are empirically set at 1 and 20, respectively;

$$\Gamma_{ij}^{c,t} = \frac{1}{|\mathbf{C}_i\cap\mathbf{C}_j|}\sum_{c_k\in\mathbf{C}_i\cap\mathbf{C}_j}\left(\left|\overline{c_{ik}^t}-\overline{c_{jk}^t}\right|\right),$$

and

$$\overline{c_{ik}^t} = \frac{1}{|c_k|}\sum_{p_j\in c_k}\frac{1}{|\mathbf{x}_{ij}|}\sum_q x_{ijq}^t.$$

Pairwise Feature Combination. To estimate the intensity of spammer agreement for any pairwise users, we follow the work [17] to employ a convex combination of the mentioned pairwise proximities with a weighting vector α for calculating $\hbar_{ij}$,

$$\hbar_{ij} = \sum_k \alpha_k\psi_{(.)}(i,j), \tag{5}$$

where each feature $\psi_{(.)}(i,j)$ (*i.e.*, PR,PT,CR,CT) is normalized within [0,1], and $\sum_k\alpha_k = 1$ $(\alpha_k\geq 0)$, k is the index for each pairwise feature.

Subsequently, following the observations in [17], we define Eq. (6) to measure the collusive characteristics of each pairwise users, which is different from [17],

$$w_{ij} = \hbar_{ij}*\eta_{PI}(i,j)-\zeta, \tag{6}$$

where ζ is a hyper-parameter that is a threshold to distinguish *high*-probability collusion signal or *low*-probability one, which can be set in different ways, for instance, $\zeta = \overline{\hbar_{(.)}*\eta_{PI}(i,j)}$, as most of pairwise users belong to **NC-NC** (non-colluder to non-colluder) and thus tend to be firstly filtered by our method; $\eta_{PI}(i,j) = \frac{|\mathbf{P}_i\cap\mathbf{P}_j|}{\sqrt{|\mathbf{P}_i|}\sqrt{|\mathbf{P}_j|}}$ is a confidence score induced by the proportion of items commonly reviewed by (u_i,u_j).

3.2. Modeling of direct relevance embedding

In this section, we mainly focus on how to learn the **direct relevance** of users over *direct* co-rating relations for making a pair of users with strong intensity of being colluders closer, otherwise the ones with a weak intensity of ones far away. Specifically, direct relevance is used to measure the degree of the spammer agreements based on their direct co-rating associations for any given pairwise users (u_i,u_j).

Now we define a *user-based signed network* built based on direct relevance of users for embedding.

Definition 1User-based Signed Network. (**USN**). *A user-based signed network [55] is defined by a 2-tuple,* i.e.,$\mathcal{G} = (\mathbf{U},\mathbf{E})$, *which consists of a set of users* $\mathbf{U} = \{u_i\}_{|\mathbf{U}|}$, *as well as a set of positive links* $\mathbf{E}^+$ *and a set of negative links* $\mathbf{E}^-$, *and* $\mathbf{E} = \mathbf{E}^+\cup\mathbf{E}^- = \{e_{ij}\}_{|\mathbf{E}|}$.

In **USN**, both positive and negative links are represented into a weighted matrix $\mathbf{W}\in\mathbb{R}^{|\mathbf{U}|\times|\mathbf{U}|}$, where each element $w_{ij}\in\mathbf{W}$ indicates the intensity of being colluders for a pair of users, especially $w_{ij}=0$ denotes the missing link between u_i and u_j. Then, the ***direct relevance*** can be estimated by employing a likelihood function to minimize the *negative* log-likelihood of collusion possibility for any pairwise users (u_i,u_j),

$$\mathcal{L}_d = -\sum_{e_{ij}\in\mathbf{E}}\log f(\mathbf{u}_i,\mathbf{u}_j;w_{ij}), \tag{7}$$

where $\mathbf{u}_{i(j)}$ denotes the K-dimensional user embedding vector. $f(.,.;.)$ is a likelihood function, and many approaches can be used to model it, here we define the function based on the principle of *ReLu* [56,57], and Eq. (7) can be rewritten as,

$$\mathcal{L}_d = \sum_{e_{ij}\in\mathbf{E}}\max\left(0, w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2+\delta\right), \tag{8}$$

where $\|.\|_F^2$ is the Frobenius norm of a vector, δ indicates a smoothing parameter. As mentioned, the learned user embeddings are expected to make a pair of users with strong intensity of being colluders closer, otherwise the ones with a weak intensity of ones far away.

More specifically, we consider the following cases regarding w_{ij}.

Case 1. *$w_{ij}>0$, which indicates $e_{ij}\in\mathbf{E}^+$ and the learnt user embeddings should be more closer, otherwise it should be penalized with a larger loss. As the smooth parameter δ is a positive value,*

$$\max\left(0, w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2+\delta\right) = w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2+\delta. \tag{9}$$

Case 2. *$w_{ij}<0$, which means $e_{ij}\in\mathbf{E}^-$ and the user embeddings should be discriminative in the learning space, otherwise it should be penalized, and a hyper-parameter is used to control the penalty,* i.e.,*the value of which should be set as 0 when $\|\mathbf{u}_i-\mathbf{u}_j\|_F^2 >= \frac{-\delta}{w_{ij}}$,*

$$\max\left(0, w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2+\delta\right) = \begin{cases} 0, & \|\mathbf{u}_i-\mathbf{u}_j\|_F^2 >= \dfrac{-\delta}{w_{ij}} \\ w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2+\delta, & \|\mathbf{u}_i-\mathbf{u}_j\|_F^2 < \dfrac{-\delta}{w_{ij}} \end{cases} \tag{10}$$

According to the discussion, Eq. (8) can be rewritten by combining the above two cases,

$$\mathcal{L}_d = \sum_{e_{ij}\in\mathbf{E}} I_{ij}\left(w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2+\delta\right), \tag{11}$$

where the indicator matrix I_{ij} is computed as follow:

$$I_{ij} = \begin{cases} 0, & w_{ij}<0\wedge\|\mathbf{u}_i-\mathbf{u}_j\|_F^2 >= \frac{-\delta}{w_{ij}} \\ 1, & otherwise. \end{cases} \tag{12}$$

After using the smoothing parameter δ to compute the value of indicator matrix I_{ij}, the parameter δ can be removed in Eq (11) as it has no influence to the gradient of $\mathbf{u}_i$ and $\mathbf{u}_j$,

$$\frac{\partial\left(I_{ij}w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2+I_{ij}\delta\right)}{\partial\mathbf{u}_i} = \frac{\partial\left(I_{ij}w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2\right)}{\partial\mathbf{u}_i}, \tag{13}$$

which means both the $max(\cdot,\cdot)$ function and parameter δ are substituted by the indicator matrix I_{ij},

$$\mathcal{L}_d = \sum_{e_{ij}\in\mathbf{E}} I_{ij}w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2. \tag{14}$$

Then, the remaining problem is how to estimate the intensity of being colluders, as well as model the indirect relevance embedding for each pairwise users. We will detail them in the following sections, respectively.

3.3. Modeling of indirect relevance embedding

As mentioned, spammers within the spam campaigns might balance their workload to evade detection, and thus explicitly modeling the *direct* relevance via *shallow relational* information (*i.e.*, direct co-rating relation) alone is insufficient for accurately capturing the collusion signals, which might make matrix **W** sparse and result in a poor performance in spammer detection, even worse when no direct correlations. Intuitively, users sharing more common neighbors may be a strong indicator that they are colluders, and thus it is a natural way to combine the direct relation and indirect relation, which is a relatively stationary collusion signal for collective review spammer detection, and is also not easily manual manipulation. Hence, here we mainly focus on how to measure the *indirect relevance* for user embedding, via modeling *indirect* common neighbors for a pair of users.

3.3.1. Indirect relevance proximity

To measure the *indirect* relevance for *pairwise* users, we employ a truncated *random walk* [49] to model the *indirect* neighborhood network structure of such users over *USN*. Note we solely consider to walk along with the positive links (*i.e.,* $\mathbf{E}^{+}$) for modeling the collusive characteristics of each pairwise users, and which is thus called ***positive-based randomwalk***. Specifically, we obtain r sequences with the maximum length of k steps for user u_i, and thus totally obtain $r \times |\mathbf{U}|$ sequences ($\mathbf{S}^{+}$), where the *indirect* neighbors of user u_i is collected from sequences containing the neighborhood structure of u_i, in which the interval between the points and u_i is within the window size ω.

Then, we learn the *indirect relevance* embeddings over their common k-step neighbors. Hence, we adopt *skip-gram* [58] to compute the loss function of indirect relevance, which is defined by the sum of the negative log probability of any pairwise users based on the learned *indirect relevance* embeddings within a window size ω.

$$\mathcal{L}_{id} = \min\left(\sum_{i=1}^{|\mathbf{U}|}\sum_{\substack{s\in\mathbf{S}^{+}\\ u_i,u_j\in s}}\left(\sum_{i-\omega<j<i+\omega} -\log \Pr\big(\mathbf{u}_j|\mathbf{u}_i\big)\right)\right), \tag{15}$$

where $\Pr(\mathbf{u}_j|\mathbf{u}_i)$ is the co-occurrence probability parameterized using the inner product kernel with softmax, which is defined by

$$\Pr(\mathbf{u}_j|\mathbf{u}_i) = \frac{\exp(\mathbf{u}_i^T\mathbf{\Phi}_j)}{\sum_{k=1}^{|\mathbf{U}|}\exp(\mathbf{u}_i^T\mathbf{\Phi}_k)} \tag{16}$$

where $\mathbf{u}_i$ denotes the learnt *direct relevance* embedding of u_i; and $\mathbf{\Phi}_j$ is u_j's *indirect relevance* embedding of u_j when u_j is the *indirect* neighbor of u_i, which is based on the principle of making the users sharing more k neighbors close in the learnt vector space.

However, Eq. (16) cannot be scalable due to the expensive computation overheads, as we need to finish the updates of all users when computing $\Pr(\mathbf{u}_j|\mathbf{u}_i)$. To tackle the problem, we employ negative sampling for optimization [58], we formulate the negative sampling function for pairwise user (u_i, u_j) when computing Eq. (16),

$$\log\sigma(\mathbf{u}_i^T\mathbf{\Phi}_j) + \kappa\mathbb{E}_{u_n\sim P_n(u)}[\log\sigma(-\mathbf{u}_i^T\mathbf{\Phi}_n)] \tag{17}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function; κ is the number of negative samples. We empirically set $P_n(u_i) \propto d_{u_i}^{3/4}$ as [58], and d_{u_i} is the degree of user u_i.

3.4. The unified model

In this section, we present a unified model to optimize our *collective opinion spammer detection* problem by minimizing a combination loss which consists of three different loss terms, *i.e.,* two losses based on the two learned relevance embeddings, *i.e., direct relevance* embeddings and *indirect relevance* embeddings, as well as a regularization loss, which is formalized as,

$$\begin{aligned}\mathcal{L}_{mix} &= (1-\beta)\mathcal{L}_{id} + \beta\mathcal{L}_d + \psi\mathcal{L}_{reg}\\ &= -(1-\beta)\left(\sum_{i=1}^{|\mathbf{U}|}\sum_{\substack{s\in\mathbf{S}^{+}\\ u_i,u_j\in s}}\left(\sum_{i-\omega<j<i+\omega}\log\frac{\exp(\boldsymbol{u}_i^T\mathbf{\Phi}_j)}{\sum_{k=1}^{|V|}\exp(\boldsymbol{u}_i^T\mathbf{\Phi}_k)}\right)\right)\end{aligned}$$

$$+\beta\left(\sum_{e_{ij}\in\mathbf{E}} I_{ij}w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2\right) + \psi(\|\mathbf{U}\|_F^2), \tag{18}$$

where β is a trade-off parameter for controlling the contributions of the learnt *direct* embeddings and *indirect* embeddings; $\|\cdot\|_F^2$ is the Frobenius norm of a matrix; and ψ is a regularization parameter. Note that, for each user u, the direct user embedding $\mathbf{u}$ and the indirect user embedding $\mathbf{\Phi}$ are simultaneously learned during training. **Spamicity Score**. To measure the intensity of a pair of users (u_i, u_j) being colluders, we employ the Frobenius distance [19] *i.e.,*

$$Score_F(u_i, u_j) = \exp(-\|\mathbf{u}_i-\mathbf{u}_j\|_F^2)$$

. The higher the $Score_F(u_i, u_j)$, the smaller the distance between two users in the feature space, and the greater the possibility of collusion between the user i and user j.

For calculating the final spamicity score of user u_i, we first calculate the $Score_F(u_i, *)$ between user u_i and all other users. Then, we collect the top-n ranked values in $Score_F(u_i, *)$, namely R_i. The final spamicity score s_i for user u_i is computed by accumulating all the values in the R_i, where a high score means the corresponding user is highly likely to collude with n other users.

3.5. Optimization

In this section, we present the solution to the optimization problem stated in Eq. (18), which is optimized as follows, **Optimize** $\mathcal{L}_d$. We first focus on the loss function of $\mathcal{L}_d$ and it can be rephrased as follow:

$$\begin{aligned}\mathcal{L}_d &= \sum_{(i,j)\in E}\mathbf{I}_{ij}w_{ij}\|\mathbf{u}_i-\mathbf{u}_j\|_F^2\\ &= 2tr(\mathbf{U}^T\mathbf{L}\mathbf{U})\end{aligned} \tag{19}$$

where $\mathbf{L} = \mathbf{D} - \widetilde{\mathbf{W}}$, and $\widetilde{\mathbf{W}} = \mathbf{I}\odot\mathbf{W}$, $\mathbf{D}\in\mathbb{R}^{|\mathbf{U}|\times|\mathbf{U}|}$ is a diagonal matrix and $\mathbf{D}_{i,i} = \sum_j \widetilde{\mathbf{W}}_{i,j}$, $\mathbf{W}$ is the weighted user-user matrix. For each iteration, we first update $\mathbf{I}$ and then update $\mathbf{L}$.

According to the Eq. (19), we can utilize SGD to minimize $\mathcal{L}_d$, and the partial derivatives of $\mathcal{L}_d$ with respect to $\mathbf{U}$ can be computed by,

$$\frac{\partial\mathcal{L}_{low}}{\partial\mathbf{U}} = 2(\mathbf{L}+\mathbf{L}^T)\cdot\mathbf{U} \tag{20}$$

Optimize $\mathcal{L}_{id}$. Here, we first calculate the partial derivative of $\mathcal{L}_{id}$ with respect to user u_i, namely,

$$\frac{\partial\mathcal{L}_{id}}{\partial\mathbf{u}_i} = -\sum_{z\in u_c\cup N^{\kappa}(u_i)}(\mathbf{O}(z,u_i) - \sigma(\mathbf{u}_i^T\mathbf{\Phi}_z))\mathbf{\Phi}_z \tag{21}$$

where $\mathbf{O}(z, u_i)$ is an indicator function, $\mathbf{O}(z, u_i) = 1$ if z is the neighbor of u_i and 0 otherwise; $N^{\kappa}(u_i)$ is κ negative samples for u_i. Then the partial derivative of $\mathcal{L}_{id}$ with respect to user u_i is updated by

$$\frac{\partial\mathcal{L}_{id}}{\partial\mathbf{\Phi}_z} = -\sum_{z\in u_c\cup N^{\kappa}(u_i)}(\mathbf{O}(z,u_i) - \sigma(\mathbf{u}_i^T\mathbf{\Phi}_z))\mathbf{u}_i \tag{22}$$

Subsequently, the regularization term can be updated by,

$$\frac{\partial\mathcal{L}_{high}}{\partial\mathbf{U}} = 2\mathbf{U}. \tag{23}$$

We detail the proposed procedure of COSD in Algorithm 1 . We first construct a user-based signed network and obtain walk sequences as steps 1 and 2. Then we iteratively update the user embedding matrix $\boldsymbol{U}$ and $\mathbf{\Phi}$ using steps 8 to 11. After that, the spamicity scores are computed through frobenius distance in steps 12 to 19. Finally, we return the user embedding matrix $\boldsymbol{U}$ and spamicity

Table 1
Abbreviation and Interpretation.

Abbreviation	Interpretation
Pairwise feature	
PR	Product rating proximity
PT	Product time proximity
CR	Category rating proximity
CT	Category time proximity
Network symbol	
USN	User-based signed network
$\mathbf{U}$	A set of users in the network
$\mathbf{E}^{+}$	A set of positive links
$\mathbf{E}^{-}$	A set of negative links
Pairwise relation	
C-C	Colluder to colluder
NC-C	Non-colluder to colluder
NC-NC	Non-colluder to non-colluder
Proposed model	
COSD	Collective opinion spammer detection approach
COSD-D	A variant of COSD which only optimizes direct relevance loss
COSD-ID	A variant of COSD which only optimize indirect relevance loss

Table 2
Statistics of the used datasets.

		AmazonCn	YelpHotel
# reviews		1,205,125	688,313
# reviewers		645,072	5122
# products		136,785	282,974
# labeled data	**# colluders**	1937	450
	# non-colluders	3118	672

Algorithm 1: The Learning Process of COSD.

```
Input: $\forall < u_i, p_j, r_{ij}, t_{ij}, c_j >$
Output: $\boldsymbol{U}$, $S$: User embedding matrix and spamicity scores
        list
Construct Signed Network based on Eq. (6)(7)
Obtain walk sequences by RandomWalk.
while not converged do
    Compute $\nabla\mathcal{L}_{low}$ based on Eq. (20)
    Compute $\nabla\mathcal{L}_{reg}$ based on Eq. (23)
    for $u_i \in U$ do
        Sample $\kappa$ negative nodes
        Compute $\nabla\mathcal{L}_{high}$based on Eq. (21) (22)
    end
    Update $\boldsymbol{U}$ and $\boldsymbol{\Phi}$
end
for $u_i \in U$ do
    for $u_j \in U$ do
        $r_{i,j} = \exp(-\|u_i - u_j\|_F^2)$
        Insert $r_{i,j}$ into $Score_i$
    end
    Accumulate top k maximum values in $Score_i$ to $s_i$
    Insert $(s_i, u_i)$ into S
end
RETURN $\boldsymbol{U}$, S
```

score list S. **Time Complexity**. Firstly, it requires $O(|\mathbf{U}|^2)$ to calculate the pairwise features for constructing a user-based signed network. Then, the cost of iteratively updating the user embedding matrix is $O(\mathbf{I} \times |\mathbf{U}| \times \kappa \times K)$, where I is the number of iterations, κ is the number of negative sampling, K is the dimension of the representation vector. Furthermore, the method requires $O(|\mathbf{U}|^2 \times \log(|\mathbf{U}|))$ complexity to calculate spamicity scores list. As $O(I \times \kappa \times K)$ always greater than $O(|\mathbf{U}| \times \log(|\mathbf{U}|))$ during the iterative process, the final time complexity is $O(I \times |\mathbf{U}| \times \kappa \times K)$.

4. Experiments

4.1. Data sets

In this section, follow [13,19,59,60] we use two public real-world datasets for evaluation, *i.e., AmazonCn, YelpHotel*, as compared to the state-of-the-art methods. The statistics of the datasets are summarized in Table 2: (i) **AmazonCn** is a collection of real consumers' reviews from *Amazon.cn* [59], which already has *gold* standard of collective spammers whose reviews are filtered by *AmazonCn*, and then gather those tagged users with their co-rating behaviors; and (ii) **YelpHotel** contains real reviews about hotels on *Yelp.com*, which is collected by Yelp's own filtering mechanism [60]. Note that there are no labels of collective spammer in YelpHotel, and thus we follow the work [13] to form spammer groups via mining user common behaviors with *Frequent Itemset Mining* (FIM).

4.2. Evaluation metrics

As mentioned, the output of collective opinion spammer detection problem is a ranking list of candidates with the spamicity scores, which can be deemed to the likelihood of candidates participating in opinion spam. As such, to evaluate the detection performance of different approaches, we follow the work in [19] to adopt four well-known metrics,

i.e., (i) **Average Precision (AP)**, which measures the average precision of collective spammer retrieving over the interval of r (recall value) from 0 to 1; (ii) **Area Under ROC Curve (AUC)**, which measures the accuracy based on *False Positive Ratio* (FPR) against *True Positive Ratio* (TPR) for binary classification; (iii) **Precision@k (P@k)**, which measures the percentage of true spammers in the top-K returned candidates; and (iv) **Normalized Discounted Cumulative Gain@k (NDCG@k)**, the performance of the detection model based on the ground-truth (*i.e.,* spammer (1)/non-spammer (0)) of the returned spammers, which is the normalization of Discounted Cumulative Gain (DCG) at each position for a chosen value of k.

According to previous studies [17,29], we set the range of k as [300, 1800] and [50, 350] for AmazonCn and YelpHotel, respectively. The setting of k is based on the number of colluders on the datasets. And in this way the metrics can fully evaluate the model's ability to distinguish spammers from normal users.

Table 3
The hyper-parameters tried for each model.

Model	Hyper-parameters
GsRank	scale factor τ : $\{1, 2, 3\}$, β : $\{6, 8, 10\}$
FraudInformer	learning rate : $\{0.1, 0.01, 0.001, 0.0001\}$
FraudScan	α_1 : $\{0.1, 0.5, 1\}$, α_2 : $\{0.2, 0.4, 0.6, 0.8\}$, learning rate: $\{0.1, 0.01, 0.001, 0.0001\}$
FraudNE	trade-off parameter α : $\{0.1, 0.5, 1, 2\}$, scale factor β : $\{5, 10, 15, 20\}$
ColluEagle	the variance σ_1, σ_2 : $\{1, 10, 30, 60, 90, 120\}$, prior rate: $\{0, 0.2, 0.4, 0.6, 0.8\}$
Our method	trade-off parameter β : $\{0, 0.2, \ldots, 0.8, 1\}$, dimention K : $\{16, 32, 64, 96, 128\}$

4.3. Baseline methods

As the proposed COSD is unsupervised, we compare our model with the following state-of-the-art unsupervised spammer detection methods: **GsRank**. [13]: This method employs a ranking model for spotting spammer groups by extracting group behavior features and individual behavior features over user-product-group relationships, which are iteratively computed for the spamicity scores of users in the reviewer group. **FraudInformer**. [17]: This method extends the Markov Random Walk model to obtain a ranking of reviewers' spamicity scores by exploring multiple heterogeneous pairwise features from reviewers' rating behaviors and linguistic patterns. **HoloScope**. [14]: This method detects fraud users by using a scalable dense block detection method to explore graph topology and temporal spikes. **FRAUDSCAN**. [19]: This method utilizes the regularized matrix factorization to learn the fraud campaign embedding for detecting fraud campaign. **FraudNE**. [16]: A state-of-the-art network embedding based method that jointly learns the user embedding and item embedding in a bipartite network, the spammers are detected by the average degree of each cluster in the network. **ColluEagle**. [29]: A powerful Markov random field (MRF)-based model which detects collusive review by considering both network effects and time effects. **Our method**. Our proposed collective opinion spammer detection method is called *COSD*, which simultaneously learn the *direct relevance embeddings* and *indirect relevance embeddings* over the co-reviewed correlations for detecting spammers. To analyze the different impacts of the two types of relevance embeddings, we also evaluate two variants of our model, that is, (i) ***COSD-D***, this is a variant of our COSD model and we only optimize *direct* relevance loss; and (ii) ***COSD-ID***, this is a variant of our COSD model that we only optimize *indirect* relevance loss.

4.4. Implementation details

For *GsRank*, we utilize FIM to extract candidate spammer groups as mentioned in [13]. For *FraudInformer*, we treat all features fairly as mentioned in [17]. For *HoloScope*, we use the implementation provided by the shared source code[1]. For *FraudInformer*, we keep the hyper-parameters (*i.e.,* $\alpha_1, \alpha_2, \alpha_3, k$) consistent with the original paper. For a fair comparison, we employ a widely-used and effective hyper-parameter search approach, namely **grid search** to select the optimal value of the hyper-parameters for all baselines and our method. The search space for each hyper-parameter of each model is shown in Table 3.

The parameters for our proposed method are empirically set as follows: The embedding size K is set as 64 on AmazonCn and 128 on YelpHotel, respectively. For the randomwalk, $\gamma = 30$, $t = 8$, $\omega = 5$, $\kappa = 8$ for two datasets; and for $\{\beta, n\}$, we set is as $\{0.6, 25\}$ for *AmazonCn*, and $\{0.4, 40\}$ for *YelpHotel*.

4.5. Evaluation results and analysis

Comparison of the Spammer detection Performance. This experiment is to evaluate the effectiveness of identifying the spammers by our approach with the baseline methods on *AmazonCn* and *YelpHotel*. Figure (2 a) and (2 b) shows *AP* and *AUC* of each method on such two datasets. Each method is performed 10 times, respectively.

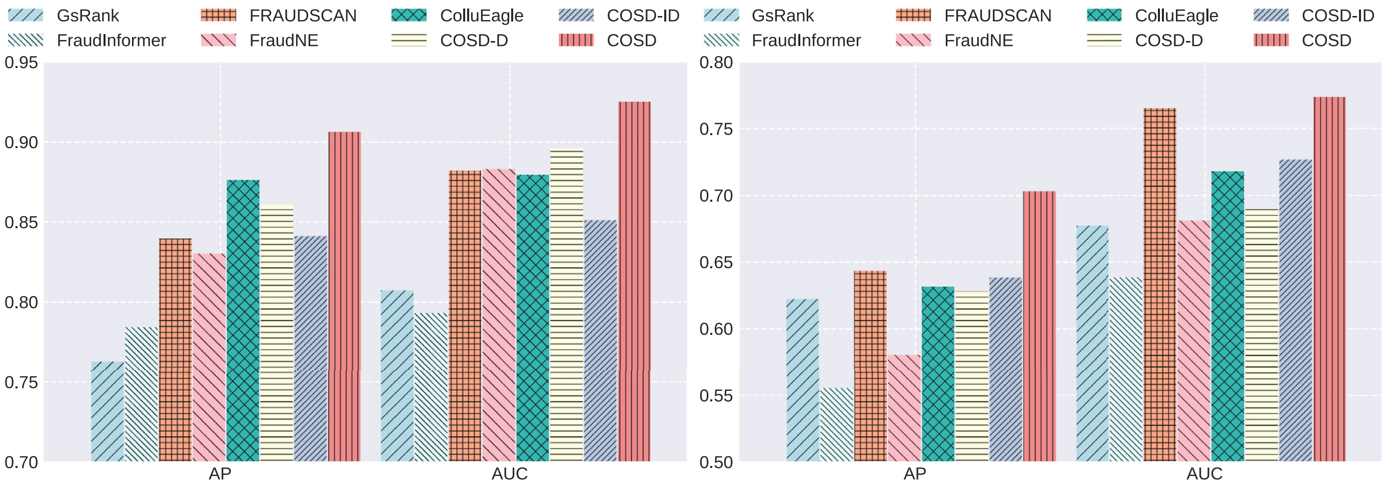
From Figure (2 a) and (2 b), we can observe that: *COSD* consistently outperforms all baselines on all metrics. For example, *COSD* outperforms *GsRank* by $[18.8\%, 14.5\%]$ in terms of *AP* and *AUC* on Amazon and $[12.9\%, 1.6\%]$ on YelpHotel, respectively. This is because *COSD* is capable of learning the information from pairwise user relations with behavioral information, while GsRank is based on group information, which might omit several subtle effective information from users. In addition, *COSD* outperforms *FraudInformer* by $[17.9\%, 16.6\%]$ on Amazon and $[26.5\%, 21.8\%]$ on YelpHotel in terms of AP and AUC, respectively. However, *COSD* and *FraudInformer* both make use of pairwise user features for collective spammers detection, which demonstrates that *COSD* is more effective and robust in exploiting the relationships between users from different-levels, *i.e., direct relevance* and *indirect relevance*, which can effectively distinguish users with a low likelihood of collusion and thereby avoiding heaps of noise. Compared with *FraudNE*, our method outperforms it on both datasets, as *FraudNE* neglects the rating information and time information in the datasets, besides it is hard for *FraudNE* to capture the potential collusion between pair of users.

Additionally, we also observe that *COSD* outperforms *FRAUDSCAN* by $[7.9\%, 9.2\%]$ on *AmazonCn*, and $[4.8\%, 1.6\%]$ on *YelpHotel* in terms of *AUC* and *AP*, respectively. And comparing with state-of-the-art method ColluEage, the performance is improved by $[3.4\%, 5.1\%]$ on *AmazonCn*, and $[11.3\%, 7.7\%]$ on *YelpHotel* in terms of *AUC* and *AP*, respectively. The reason might be *COSD* effectively captures the possibility of collusion for pairwise users who do not have common co-reviewed records, and *COSD* can model users' neighbor structure over the *indirect relations* for detection.

From Fig. 3[2] we can observe that most of the methods perform worse when rank k increases, and our method outperforms other methods consistently on AmazonCn and Yelp datasets in terms of Precision@k and NDCG@k, which also demonstrates the effectiveness of our method. We can also observe that there is a drop with ranks of 600 for FraudInformer. This is because that FraudInformer explores multiple heterogeneous pairwise features from the reviewers rating behaviors and linguistic patterns. However, linguistic patterns are not reliable in the most real-world scenario as reviewers can modify their comments to prevent them from being detected and FraudInformer do not consider the potential relation between users. In addition, due to the small scale of AmazonCn dataset, the sample distribution is prone to be uneven. Different from the baseline methods, *COSD* can combine the*direct relevance* and *indirect relevance* over user behavioral features for jointly op-

[1] https://github.com/shenghua-liu/HoloScope

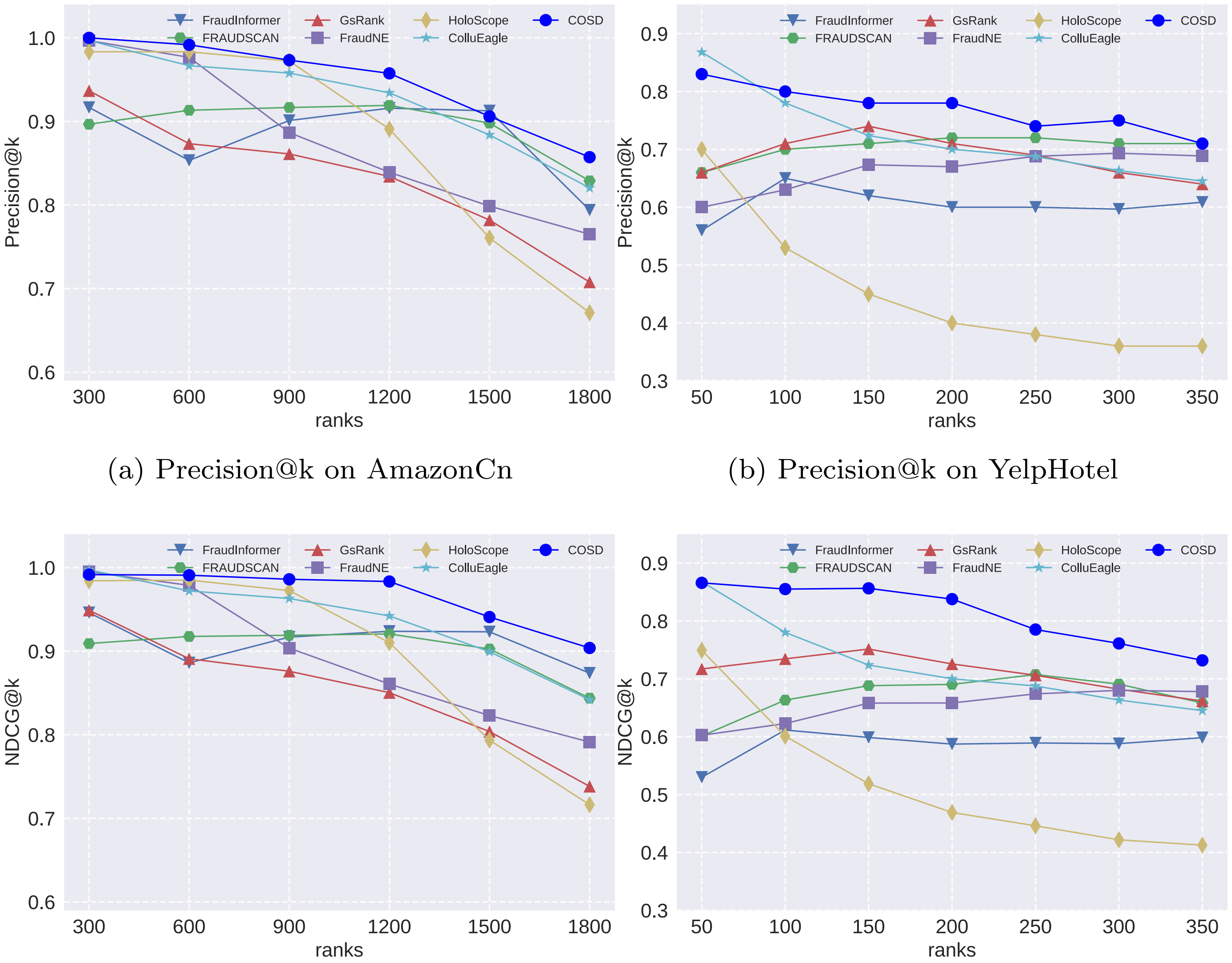
[2] We select different scales and ranges of X-axis for two datasets because of the difference of scales between Amazon_cn and YelpHotel.



(a) AmazonCn

(b) YelpHotel

Fig. 2. Effectiveness comparison between COSD and state-of-the-art approaches on the two datasets.



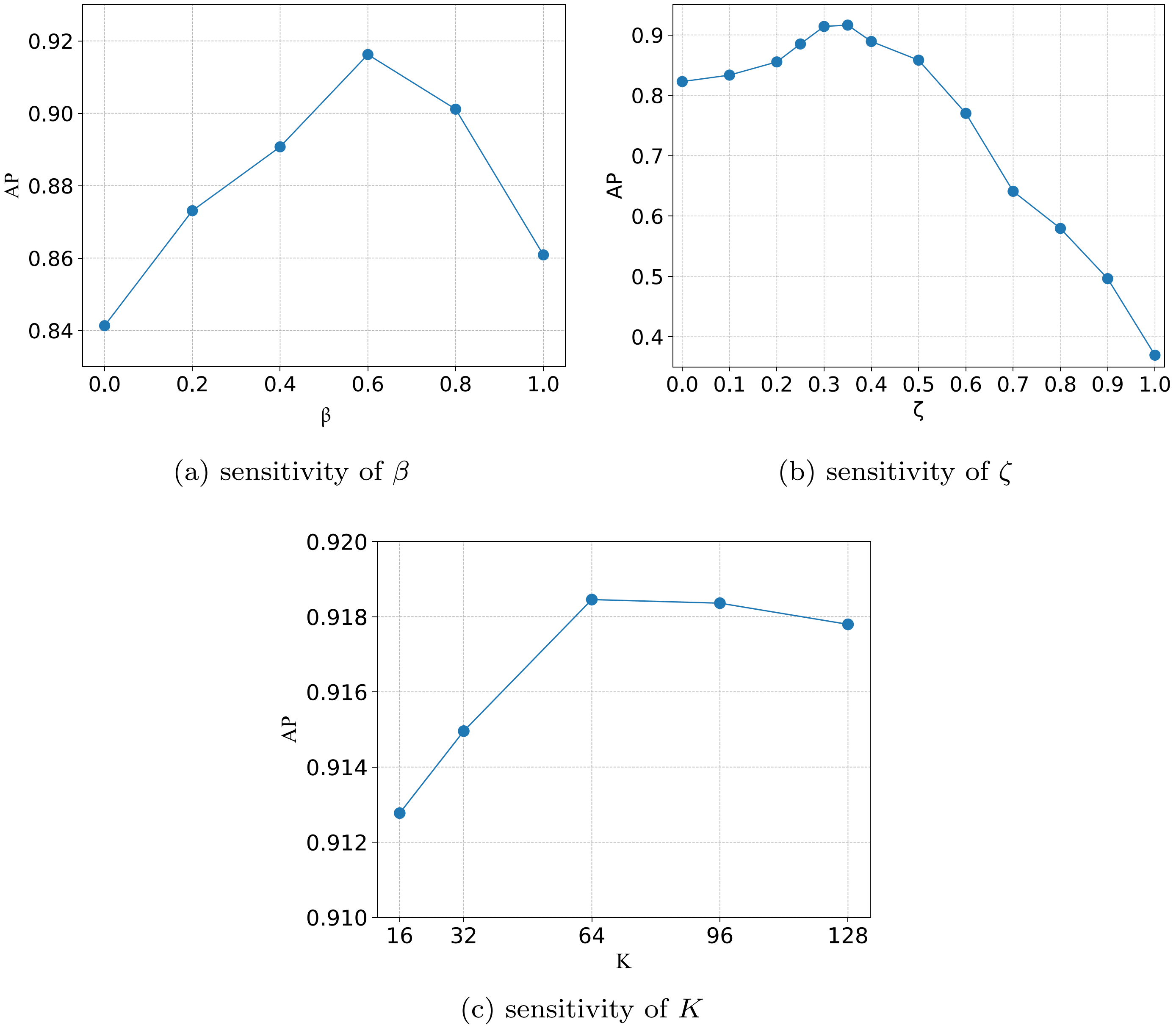
(a) Precision@k on AmazonCn

(b) Precision@k on YelpHotel

(c) NDCG@k on AmazonCn

(d) NDCG@k on YelpHotel

Fig. 3. Model performance in terms of Precision@k.



(a) sensitivity of β

(b) sensitivity of ζ

(c) sensitivity of K

Fig. 4. Illustration of the effect of different parameters, *i.e.*, α, ζ and K on *AmazonCn*.

timizing the losses of learning such two relevance embedding, and thus the learned user embeddings might preserve not only the behavioral information but also the topology information, which makes the users with *direct* and *indirect* collusive connections close in the learned low-dimensional vector space. **The Impact of Direct Relevance Embedding**. From Figure (2), we can observe that *COSD-D* and *COSD-ID* achieve good results. *COSD-D* outperforms all baselines on *AmazonCn* in terms of *AP*, which demonstrate the effectiveness of using a signed network with *positive* and *negative* links, and it also shows that optimizing the *direct relevance* is capable of more accurately reflecting the realistic relationships in low-dimensional space. **The Impact of indirect Relevance**. From Figure (2), we can observe that *COSD-ID* does not perform as well as *COSD-D*, however it still outperforms other baselines on *AmazonCn* in terms of *AP*, which demonstrates that the effectiveness of performing a random walk along with the positive links on a built user-based signed network while discarding the negative links with weak collusion signals (even noise) of being colluders, and thus it can focus on the analysis of the filtered neighbor structure for more accurately capturing the collusion signals, and it also shows that using the indirect relations is useful for exploring the implicit associations for any pairwise uses in these fraud campaigns.

4.6. On the sensitivity of parameters

In this section, we study the impact of three parameters, *i.e.*, β, ζ, K, in our method[3]

For parameter β (used in Eq. (18)), which controls the contributions of *direct relevance embeddings* and *indirect relevance embeddings* in our model, and As shown in Figure (4 a), our proposed method achieves the best result when $\beta = 0.6$, and simultaneously making use of both *direct relevance embedding* and *indirect relevance embedding* outperforms the extreme cases when only considering the *indirect relevance embedding* ($\beta = 0$) or the *direct rele-*

[3] We only perform parameter sensitivity experiments on AmazonCn dataset as the scale of the YelpHotel dataset is too small to observe significant changes.

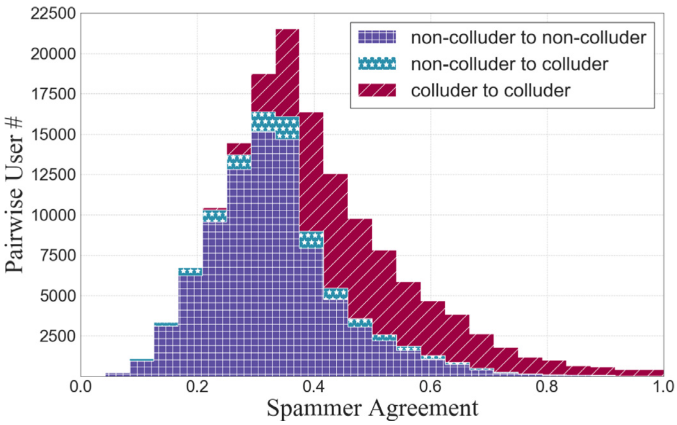


Fig. 5. Empirically data analysis. We mainly focus on three different types of pairwise user relations collected from *AmonzonCn, i.e.,* **C-C, NC-C, NC-NC**. X-axis denotes the intensity of spammers' agreements for measuring the collusive characteristics between each pairwise users, and Y-axis indicates the corresponeding number of pairwise users.

vance embedding ($\beta = 1$), which is consistent with the former analysis, *i.e., direct relevance embedding* is more important than *indirect relevance embedding*.

For parameter ζ (used in Eq. 6), which is a threshold to distinguish *high*-probability collusion signal or *low*-probability one. As shown in Figure (4 b), our proposed method achieves the best result within the range of [0.3,0.4], which demonstrate that when the value of ζ is small, it might result in more noise involved, while a large one would lead to the insufficiently learning of *indirect relevance embedding* due to neglecting more indirect collusive relationships.

For parameter K, which is the dimensional (same for *direct* and *indirect*) of the representation embedding vector. As shown in Figure (4 c), when varying K, the performance first increases and then decreases, and the best performance is achieved when $K = 64$, which demonstrates that our proposed method can well work with a low dimensional vector space.

4.7. Empirical study of different relations

In this section, we present an empirical study on a real-world dataset, namely Amazon_cn [59], where more than 100,000 pairs of users are used to explore the distribution of different co-review relations on spammer-agreement ($\hbar_{ij}$). The spammer-agreement $\hbar_{ij}$ is calculated based on the sum of four extracted spam-campaign characteristics (*i.e.,* $\psi_{(PR)}(i, j)$, $\psi_{(PT)}(i, j)$, $\psi_{(CR)}(i, j)$ and $\psi_{(CT)}(i, j)$) over the pairwise users from Amazon_cn. Figure (5) presents the distribution of the spammer-agreement ($\hbar_{ij}$) to the number of pairwise users over three different co-reviewed relations, *i.e., colluder to colluder* (**C-C**), *non-colluder to colluder* (**NC-C**), and*non-colluder to non-colluder* (**NC-NC**). For the sake of simplicity, we mainly focus on the discussion of **C-C** and **NC-NC**, as the remaining discussions are similar. From Figure (5), we can observe that the change ratio of **C-C** and **NC-NC** is different when the *spammer agreement* increases, *i.e.,* the number of **NC-NC** considerably decreases occurred within the range of [0.4,0.8]. As opposed to **NC-NC**, the number of **C-C** gradually increases within such range. Actually, similar trends are also reported in [17].

5. Conclusion

In this paper, we propose a novel *unsupervised* network embedding-based approach to jointly combine direct and indirect neighborhood exploration for learning the user embeddings for more accurately identifying spam reviewers. Experiments on two real-world datasets demonstrate that our proposed approach significantly outperforms all of baselines on all metrics, by learning the user representation via jointly optimizing the *direct relevance* and *indirect relevance*.

For future work, more robust pairwise features and prior knowledge information can be incorporated to enhance the ability of direct relevance modeling. And a promising direction is to design an unsupervised graph neural network (based) algorithm for indirect relevance modeling. With the powerful feature extraction capability of the graph neural network, the embeddings of users will be more robust and accurate in the learning space. Moreover, it is significant to consider the efficiency of the spammer detection algorithm (*i.e.,* to implement an online detection system).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No.L1924068, Grant No.61602197, Grant No.61772076 and Grant No.61972448, and in part by CCF-AFSG Research Fund under Grant No.RF20210005.

References

[1] T. Fornaciari, M. Poesio, Identifying fake amazon reviews as learning from crowds (2014).
[2] H. Li, Z. Chen, A. Mukherjee, B. Liu, J. Shao, Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns, in: Ninth International AAAI Conference on Web and Social Media, 2015.
[3] Y. Ren, Y. Zhang, Deceptive opinion spam detection using neural network, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 140–150.
[4] X. Wang, K. Liu, S. He, J. Zhao, Learning to represent review with tensor decomposition for spam detection, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 866–875.
[5] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, J. Shao, Bimodal distribution and co-bursting in review spam detection, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1063–1072.
[6] M. Ott, Y. Choi, C. Cardie, J.T. Hancock, Finding deceptive opinion spam by any stretch of the imagination, arXiv preprint arXiv:1107.4557 (2011).
[7] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, H.W. Lauw, Detecting product review spammers using rating behaviors, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 939–948.
[8] N. Jindal, B. Liu, E.-P. Lim, Finding unusual review patterns using unexpected rules, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 1549–1552.
[9] F.H. Li, M. Huang, Y. Yang, X. Zhu, Learning to identify review spam, in: Twenty-second International Joint Conference on Artificial Intelligence, 2011.
[10] S. Günnemann, N. Günnemann, C. Faloutsos, Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 841–850.
[11] N. Günnemann, S. Günnemann, C. Faloutsos, Robust multivariate autoregression for anomaly detection in dynamic product ratings, in: Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 361–372.
[12] J. Ye, S. Kumar, L. Akoglu, Temporal opinion spam detection by multivariate indicative signals, in: Tenth International AAAI Conference on Web and Social Media, 2016.
[13] A. Mukherjee, B. Liu, N. Glance, Spotting fake reviewer groups in consumer reviews, in: Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 191–200.
[14] S. Liu, B. Hooi, C. Faloutsos, Holoscope: Topology-and-spike aware fraud detection, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1539–1548.
[15] J. Ye, L. Akoglu, Discovering opinion spammer groups by network footprints, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 267–282.
[16] M. Zheng, C. Zhou, J. Wu, S. Pan, J. Shi, L. Guo, Fraudne: a joint embedding approach for fraud detection, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[17] C. Xu, J. Zhang, Combating product review spam campaigns via multiple heterogeneous pairwise features, in: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, 2015, pp. 172–180.
[18] S. Rayana, L. Akoglu, Collective opinion spam detection: Bridging review networks and metadata, in: Proceedings of the 21th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, 2015, pp. 985–994.
[19] C. Xu, J. Zhang, Z. Sun, Online reputation fraud campaign detection in user ratings, in: IJCAI, 2017, pp. 3873–3879.
[20] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, V. Subrahmanian, Rev2: Fraudulent user prediction in rating platforms, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 333–341.
[21] C.G. Harris, Detecting deceptive opinion spam using human computation, in: Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
[22] J. Li, M. Ott, C. Cardie, E. Hovy, Towards a general rule for identifying deceptive opinion spam, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 1566–1576.
[23] Q. Li, Q. Wu, C. Zhu, J. Zhang, W. Zhao, Unsupervised user behavior representation for fraud review detection with cold-start problem, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 222–236.
[24] Z. You, T. Qian, B. Liu, An attribute enhanced domain adaptive model for cold-start spam review detection, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1884–1895.
[25] S. Feng, R. Banerjee, Y. Choi, Syntactic stylometry for deception detection, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2012, pp. 171–175.
[26] Y. Liu, B. Pang, X. Wang, Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph, Neurocomputing 366 (2019) 276–283.
[27] P. Kaghazgaran, J. Caverlee, A. Squicciarini, Combating crowdsourced review manipulators: A neighborhood-based approach, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 306–314.
[28] S. Rayana, L. Akoglu, Collective opinion spam detection using active inference, in: Proceedings of the 2016 SIAM International Conference on Data Mining, SIAM, 2016, pp. 630–638.
[29] Z. Wang, R. Hu, Q. Chen, P. Gao, X. Xu, Collueagle: collusive review spammer detection using markov random fields, Data Min Knowl Discov 34 (6) (2020) 1621–1641.
[30] I.E. Aguerri, A. Zaidi, Distributed variational representation learning, IEEE Trans Pattern Anal Mach Intell 43 (1) (2019) 120–138.
[31] X. Jia, X.-Y. Jing, X. Zhu, S. Chen, B. Du, Z. Cai, Z. He, D. Yue, Semi-supervised multi-view deep discriminant representation learning, IEEE Trans Pattern Anal Mach Intell 43 (7) (2020) 2496–2509.
[32] Y. Xie, B. Yu, S. Lv, C. Zhang, G. Wang, M. Gong, A survey on heterogeneous network representation learning, Pattern Recognit 116 (2021) 107936.
[33] L. Zhang, X. Li, J. Xiang, Y. Qi, Lhone: Label homophily oriented network embedding, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 665–670.
[34] M.A. Lozano, F. Escolano, M. Curado, E.R. Hancock, Network embedding from the line graph: random walkers and boosted classification, Pattern Recognit Lett 143 (2021) 36–42.
[35] W. Zhao, J. Luo, T. Fan, Y. Ren, Y. Xia, Analyzing and visualizing scientific research collaboration network with core node evaluation and community detection based on network embedding, Pattern Recognit Lett 144 (2021) 54–60.
[36] Z.-Y. Ran, W. Wang, B.-G. Hu, On connections between rényi entropy principal component analysis, kernel learning and graph embedding, Pattern Recognit Lett 112 (2018) 125–130.
[37] J. Lu, H. Wang, J. Zhou, Y. Chen, Z. Lai, Q. Hu, Low-rank adaptive graph embedding for unsupervised feature extraction, Pattern Recognit 113 (2021) 107758.
[38] W. Hu, J. Gao, J. Xing, C. Zhang, S. Maybank, Semi-supervised tensor-based graph embedding learning and its application to visual discriminant tracking, IEEE Trans Pattern Anal Mach Intell 39 (1) (2016) 172–188.
[39] R. Jiang, A.T. Ho, I. Cheheb, N. Al-Maadeed, S. Al-Maadeed, A. Bouridane, Emotion recognition from scrambled facial images via many graph embedding, Pattern Recognit 67 (2017) 245–251.
[40] W. Zheng, L. Yin, X. Chen, Z. Ma, S. Liu, B. Yang, Knowledge base graph embedding module design for visual question answering model, Pattern Recognit 120 (2021) 108153.
[41] S. Yang, B. Yang, Enhanced network embedding with text information, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 326–331.
[42] W. Wei, Z. Wang, X. Mao, G. Zhou, P. Zhou, S. Jiang, Position-aware self-attention based neural sequence labeling, Pattern Recognit 110 (2021) 107636.
[43] C. Shi, B. Hu, W.X. Zhao, S.Y. Philip, Heterogeneous information network embedding for recommendation, IEEE Trans Knowl Data Eng 31 (2) (2018) 357–370.
[44] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, M. Qiu, Global context enhanced graph neural networks for session-based recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 169–178.
[45] S. Zhao, W. Wei, D. Zou, X.-L. Mao, Multi-view intent disentangle graph networks for bundle recommendation, Thirty-sixth AAAI conference on artificial intelligence, 2022.
[46] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[47] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Nips, volume 14, 2001, pp. 585–591.
[48] R.C. Wilson, E.R. Hancock, E. Pekalska, R.P. Duin, Spherical and hyperbolic embeddings of data, IEEE Trans Pattern Anal Mach Intell 36 (11) (2014) 2255–2269.
[49] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[50] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.
[51] S.F. Mousavi, M. Safayani, A. Mirzaei, H. Bahonar, Hierarchical graph embedding in vector space by graph pyramid, Pattern Recognit 61 (2017) 245–254.
[52] H. Bahonar, A. Mirzaei, R.C. Wilson, Diffusion wavelet embedding: a multi-resolution approach for graph embedding in vector space, Pattern Recognit 74 (2018) 518–530.
[53] Y. Shi, M. Lei, H. Yang, L. Niu, Diffusion network embedding, Pattern Recognit 88 (2019) 518–531.
[54] Z. Wang, X. Ye, C. Wang, J. Cui, P. Yu, Network embedding with completely-imbalanced labels, IEEE Trans Knowl Data Eng (2020).
[55] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 641–650.
[56] R.H. Hahnloser, H.S. Seung, Permitted and forbidden sets in symmetric threshold-linear networks, in: Advances in Neural Information Processing Systems, 2001, pp. 217–223.
[57] R.H. Hahnloser, R. Sarpeshkar, M.A. Mahowald, R.J. Douglas, H.S. Seung, Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit, Nature 405 (6789) (2000) 947–951.
[58] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).
[59] C. Xu, J. Zhang, K. Chang, C. Long, Uncovering collusive spammers in chinese review websites, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013, pp. 979–988.
[60] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance, What yelp fake review filter might be doing? in: Seventh International AAAI Conference on Weblogs and Social Media, 2013.

Ziyang Wang received the bachelor degree from Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently a master at Huazhong University of Science and Technology, China. His research interests include information retrieval, data mining and social computing.

Wei Wei received the Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China, in 2012. He is currently an associate professor with the School of Computer Science and Technology, Huazhong University of Science and Technology. He was a research fellow with Nanyang Technological University, Singapore, and Singapore Management University, Singapore. His current research interests include artificial intelligence, natural language processing, information retrieval, data mining, machine learning, and social computing and recommender system.

Xian-Ling Mao received the Ph.D. degree from Peking University, in 2013. He is currently an associate professor of computer science with the Beijing Institute of Technology. He works in the fields of machine learning and information retrieval. His current research interests include topic modeling, learning to hashing, and question answering. Dr. Mao is a member of the IEEE Computer Society and a member of the Association for Computing Machinery (ACM).

Guibing Guo received the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2015. He is currently an Associate Professor with the Software College, Northeastern University, China. His research interests include recommender systems, deep learning, natural language processing, and data mining.

Pan Zhou received the Ph.D. degree from the Georgia Institute of Technology, America, in 2011. He is currently an associate professor with the School of Cyber Science and Engineering, Huazhong University of Science and Technology. His current research interests include big data analysis, data and network security and privacy, and wireless scheduling algorithms. Dr. Zhou is a member of the IEEE Computer Society.

Sheng Jiang received the Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China, in 2009. He is currently a lecture with the School of Computer Science and Technology, Huazhong University of Science and Technology. His current research interests include data mining and natural language processing.