Social Spammer Detection: A Multi-Relational Embedding Approach

Jun Yin[1,2], Zili Zhou[2,3], Shaowu Liu[2], Zhiang Wu[1⋆], and Guandong Xu[2]

[1] School of Info. Engineering, Nanjing Univ. of Finance and Economics, China.
[2] Advanced Analytics Institute, Univ. of Technology, Sydney, Australia.
[3] School of Computer Engineering and Science, Shanghai University.
{Jun.Yin-2, Zili.Zhou}@student.uts.edu.au, {Shaowu.Liu, Guandong.Xu}@uts.edu.au, zawuster@gmail.com

Abstract. Since the relation is the main data shape of social networks, social spammer detection desperately needs a relation-dependent but content-independent framework. Some recent detection method transforms the social relations into a set of topological features, such as degree, k-core, etc. However, the multiple heterogeneous relations and the direction within each relation have not been fully explored for identifying social spammers. In this paper, we make an attempt to adopt the *Multi-Relational Embedding* (*MRE*) approach for learning latent features of the social network. The *MRE* model is able to fuse multiple kinds of different relations and also learn two latent vectors for each relation indicating both sending role and receiving role of every user, respectively. Experimental results on a real-world multi-relational social network demonstrate the latent features extracted by our MRE model can improve the detection performance remarkably.

Keywords: Social Spammer, Social Networks, Heterogeneous Relations, Graph Embedding, Classification

1 Introduction

Social networks have played a huge role in information dissemination and communication. While the social media is favoring both organizations and individuals with great facilities, it has become an emerging and effective platform on which malicious users overwhelm other users with unwanted content [8]. It has been shown that around 83% of users have received more than one unwanted friend requests or messages in social networking platforms and one in 200 social messages contain spam [5, 26]. These spammers and the misleading contents released by them are seriously threatening the sustainable development of online social networks.

In the literature, an extensive body of research has been devoted to identify various kinds of spam, such as email spam [22], Web spam [4, 31], review/reviewer spam on e-commerce sites [6, 29], and consequently, social spam [5, 14]. The main research stream within spammer detection adopts the two-phase approach: constructing multi-fold features to indicate the abnormal behavior, and developing supervised classifiers or

⋆ Corresponding author.

unsupervised ranking algorithms. Finding right features largely determines the detection performance, and it is both *data-specific* and *task-specific*. That is, a right feature should be computable on the available data and it should also be qualified for the specific detection task. Along this line, feature construction towards identifying spam from online reviews in e-commerce has been widely studied. Researchers have designed a variety of features for reviews, users, or even user groups, by fully exploiting the metadata of the review such as rating, timestamps and review text [15, 16, 23, 29]. Nevertheless, spammer detection in social networks is much different from that in e-commerce. The metadata in social networks, especially the contents, is relatively scarce, because the whisper contents should not be exposed due to user privacy. By contrast, the topological relation becomes the inherent attribute of social networks, but it exhibits weakly in e-commerce platforms. Therefore, social spammer detection calls for the *relation-dependent* but *content-independent* framework.

There is limited research on spammer detection framework solely on social relations. Fakhraei *et al* [5] make a useful attempt in this area: for each relation, a topological graph is generated to describe the interactions among single relation in a topological way, with the underlying assumption that spammers are more important in the graph. Moreover, for each user, they use the sequence of relations based on the time it happened to partly disclose the relevance among relations. Then a framework is combined with these two aspects. However, both graph and sequence are extracted from single relation and individual user, the inter-activities between two users cross different relations have been neglected.

Graph structured embedding method has been widely used in the area like knowledge graph [18]. It excavates the latent information with the utilizing of both edges and vertices, which can exactly make up for the shortcoming of previous researches. Hence, we shall propose our *Multi-Relational Embedding* (*MRE*) model to trade on the preponderance of graph and remedy the limitation of it with graph-embedding method. The main contributions of this work are summarized as follows:

- To the best of our knowledge, this is the first attempt to model different types of relations among all users in a single model for multi-relation spammer detection.
- The *MRE* model is made scalable with the option to set the embedding space size, thus, both small and large number of relation types can be accommodated.
- We conducted empirical experiments on a large real-world social network dataset and provided interesting findings and discussions.

The following sections will be organized as follows. In Section 2, we formulate the problem and outline the previous methodologies along with its limitations. We technically address details of our *MRE* model in Section 3. In Section 4, we exhibit experiment results, and present related work in Section 5. Finally, we conclude our work and give future plan in Section 6.

2 Preliminaries

In this section, we define the problem of identifying spammers from the multi-relational social network, and briefly summarize existing approaches as well as their limitations that motivated our research.

2.1 Formulating Multi-relational Spammer Detection

Let $\mathcal{U} = \{u_1, \cdots, u_n\}$ be the set of n users who are connected by m kinds of relations denoted as $\mathcal{R} = \{r_1, \cdots, r_m\}$. In this multi-relational network, assume the relation of type $r_k \in \mathcal{R}$ exists between two users u_i and u_j is encoded as π_{ijk}, where the first two subscripts indicate users and the third subscript tells the type of relation. Examples of relations include "add friend" and "block user". Note that the relation has a direction, therefore π_{ijk} and π_{jik} are different, where the first one is relation r_k from user u_i to user u_j and the second one is the same relation but from user u_j to user u_i. The collection of all relations π_{ijk} is denoted as $\mathit{\Pi}$.

The goal is to learn from provided relations $\mathit{\Pi}$ to predict the probability of being spammer for each user $u \in \mathcal{U}$. In practice, the probability is often unnormalized, thus the goal becomes ranking the users correctly instead of estimating the exact probability, i.e., spammers are ranked higher than normal users but the ordering among spammers does not matter. Let further divide the set of users $\mathcal{U}$ into a set $\mathcal{S}$ of spammers and a set $\mathcal{L}$ of legitimate users, i.e., $\mathcal{U} = \mathcal{S} \cup \mathcal{L}$. The ultimate goal of the spammer detection is to learn an order function for all users, denoted as $O(\mathcal{U})$. Then, we can define an indicator variable $I_{ij} = 1$ to represent that $u_i \in \mathcal{S}, u_j \in \mathcal{L}, O(u_i) > O(u_j)$, otherwise for $I_{ij} = 0$. Hence, one possible formulation of multi-relational spammer detection is:

$$\underset{O(\mathcal{U})}{\operatorname{argmax}} \sum_{u_i \in \mathcal{S}} \sum_{u_j \in \mathcal{L}} I_{ij}. \tag{1}$$

2.2 Feature Design from Multi-relational Data

While many quality classifiers are available, the main challenge is how to design effective features. Unlike traditional spammer detection models that make use of textual data, the multi-relational social network focuses on topological information. The main features design approaches to multi-relational data are graph-based and sequence-based approaches.

Graph-based Features Graph-based features are extracted by converting relations into a directed graph $\mathcal{G}$, where the vertices $\mathcal{V}$ represent the users and the edges $\mathcal{E}$ represent interactions among users. When there exist multiple types of relations, a graph is usually generated for each of them: $\{\mathcal{G}_1, \ldots, \mathcal{G}_m\}$ for m types of relations. Each graph is then feed into a feature extraction function $\mathbf{X}_m^{\text{graph}} = \psi(\mathcal{G}_m)$ to convert a directed graph into either numerical or categorical feature matrix $\mathbf{X}_m^{\text{graph}}$ for each type of relation. Existing literature has defined many feature extraction functions $\psi(\cdot)$, and we list a few popular choices:

- *Triangle Count* [24] computes how many times each vertex involves in subgraphs with three vertices, i.e., a triangle structure.
- *k-core* [1] measures the centrality of each vertex by gradually removing the least connected vertices. The earlier a vertex was removed the lower the centrality.
- *Graph Coloring* [9] assigns a set of colors to vertices with no adjacent vertices having the same colors, and the assigned colors are used as a categorical feature.

- *Page Rank* [19] similar to measuring the importance of Web page by counting the number of incoming links, the incoming edges are counted for each vertex.
- *Weakly Connected Components* [20] counts the number of subgraphs each vertex involves without considering the direction of edges.

Despite of their effectiveness, existing graph feature extraction techniques often assume a separated graph for each type of relation, or aggregation is performed by simple addition. The interactions among relations have been largely overlooked.

Sequence-based Features Sequence-based features are extracted by converting relations into a user-wise sequence $\mathcal{T}_i = \{t_1, \cdots, t_q\}$ for each user u_i, where $t_j \in [1, m], 1 \leq j \leq q$, is the relation type and the length q of the sequence depends on the user. The sequence of each user is then fed into a feature extraction function $\mathbf{x}_i^{\text{seq}} = \psi(\mathcal{T}_i)$ to convert the sequence of user u_i into a feature vector $\mathbf{x}_i^{\text{seq}}$. Sequence-based feature extraction has also been used in spammer detection:

- *Sequential k-gram Features* [5] considers the activity order of users by counting the frequency of each length k sub-sequences for each user.
- *Mixture of Markov Models* [21] can be used to overcome the limitation of small k in k-gram models by identifying a small set of important and long sequence chains.

Unlike graph-based features, sequence-based feature can capture interactions among different types of relations to some extent. Nevertheless, user interactions are not captured properly as the sequence features are extracted independently for each user.

In this work, we take the graph-based approach, however, all types of relations are modeled simultaneously in a single graph instead of separated graphs for each type of relation. By embedding the users and relations at the same time, the proposed model overcomes the limitations of traditional graph-based and sequence-based feature extraction methods.

3 Methodology

In this section, we propose the *MRE* model to capture the interactions among different types of relations. The rest of this section defines the multi-relation learning problem, followed by a detailed description of the *MRE* model. In what follows, we shall use u and r as identity of user and type of relation, and use the bold-faced notation $\mathbf{u}$ and $\mathbf{r}$ to represent the latent vectors for user and relation respectively.

3.1 Multi-Relational Embedding

The prediction problem itself has been well-studied in literature, and mature classifiers are available in open source libraries. However, the main issue is that off-the-shelf classification algorithms expect numerical variables as input and do not accept input format such as relations defined in Section 2.1. Therefore, the main challenge is to learn a vector representation $\mathbf{u}$ for each user $u \in \mathcal{U}$ from relations such that the new representation

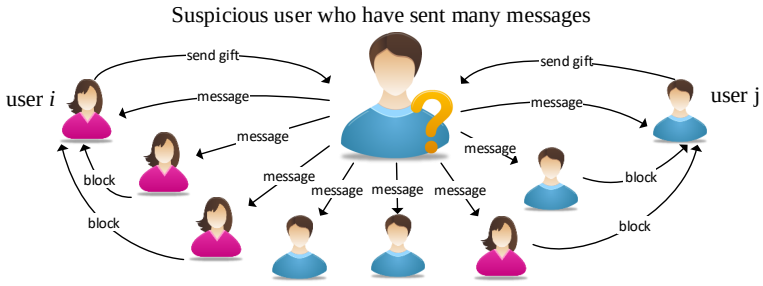


Fig. 1. The suspicious user in the middle who have sent messages to too many users looks like a spammer. However, he has received gifts from users i and j, which is a strong indicator of good user. But we realized that the users who sent gifts are actually low-credit users who have been blocked by others, thus the fact might be the spammer is trying to fool the detection system.

is in numerical format while discriminative information is preserved. Attempts [5] were made in literature to learn such representations, but all of them learn the representations for each type of relation independently. While informative interactions may exist among relations, we propose to learn from all types of relations simultaneously.

Learning from all types of relations at the same time provides more insights into user behaviors than looking at each individual relation type. For example, the simplest method of encoding relations into numerical representation is counting, i.e., how many times a user has sent/received each type of relation. Despite of its simplicity, this approach does encode important information such as "*users who have sent more messages are more likely to be spammers*". However, interactions among relations are ignored. A toy example of interactions among relations is shown in Fig. 1 for three types of relations: "*send message*", "*block user*", "*send gift*". In multi-relational embedding, the interactions among relations can be learned as latent factors.

To be specific, we model all users and all types of relations in a shared embedding space. Given the set of all relations Π, we can construct a graph $\mathcal{G}$ where users are the vertices and relations are the edges. Then each user $u_i \in \mathcal{U}$ is represented as a numerical vector $\mathbf{u}_i \in \mathbf{R}^z$ and each type of relation $\mathbf{r}_k \in \mathbf{R}^z$ is represented as a numerical vector $\mathbf{r}_k \in \mathbf{R}^z$. The shared embedding space has a user-defined dimension z. Unlike traditional matrix factorization, the multi-relational embedding has a graph structure, and representation of type of edges (relations) must be learned. Formally, we aim to learn all $\mathbf{u} \in \mathbf{R}^z$ and $\mathbf{r} \in \mathbf{R}^z$ such that

$$\mathbf{u}_i \cdot \mathbf{r}_k + \mathbf{u}_j \cdot \mathbf{r}_k \approx \pi_{ijk}. \tag{2}$$

The above model has not considered the direction of relations yet. For the same type of relation, the sending node (*src*) and the receiving node (*dest*) often delivery different semantic meanings. For instance, the spammer tends to propagate the unwanted content to a large number of users, where the user who usually acts as the sending node should be embedded as the spam user. Therefore, it is a good idea to model them separately. To do so, we define two vectors $\mathbf{r}_k^{\text{src}}$ and $\mathbf{r}_k^{\text{dest}}$ for each type of relation $r_k \in \mathcal{R}$. Similarly, we define $\mathbf{u}_i^{\text{src}}$ and $\mathbf{u}_i^{\text{dest}}$ for each user $u_i \in \mathcal{U}$. Then jointly, we aim to learn $\mathbf{r}^{\text{src}}$, $\mathbf{r}^{\text{dest}}$,

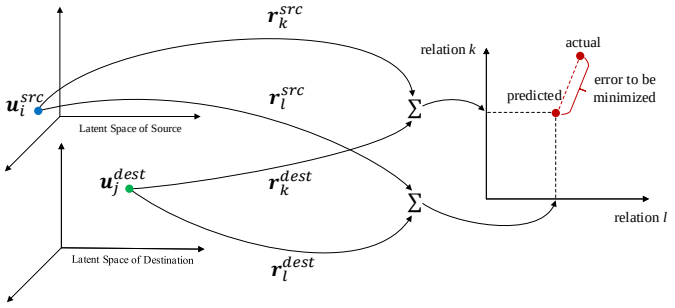


Fig. 2. Illustration of the MRE model on two relations.

$\mathbf{u}^{\text{src}}$, and $\mathbf{u}^{\text{dest}}$ for all types of relations and all users such that

$$\mathbf{u}_i^{\text{src}} \cdot \mathbf{r}_k^{\text{src}} + \mathbf{u}_j^{\text{dest}} \cdot \mathbf{r}_k^{\text{dest}} \approx \pi_{ijk}. \tag{3}$$

Fig. 2 illustrates the proposed *MRE* model considering both sending node and receiving node with two relations. As can be seen, we have a source user vector $\mathbf{u}_i^{src}$ and a destination user vector $\mathbf{u}_j^{dest}$, which are mapped to the shared embedding space of two types of relations r_k and r_l. The learning task is to estimate the latent vectors of users and relations such that the prediction error is minimized.

Algorithm 1 *Multi-Relational Embedding* Algorithm

Input: List of triples (source user, destination user, relation type).
Preparing:
Step 1: Draw pairwise user pair set $\mathcal{S}$, each pair contains a relation sender user and a relation receiver user.
Step 2: Collect all relation types as $\mathcal{R}$.
Embedding Model Training:
Step 3: Repeat
for each user pair $(u_i, u_j) \in \mathcal{U}$ **do**
  Draw relation frequency vector $\mathbf{r}$ for (u_i, u_j),
  Each value of $\mathbf{r}$ counts the frequency of a type relation sent from u_i to u_j.
  for each $r_k \in \mathcal{R}$ **do**
    Measure the real frequency value with the value predicted by user embedding and transfer matrices with relation k.
    error $= \|\pi_{ijk} - (\mathbf{u}_i^{\text{src}} \cdot \mathbf{r}_k^{\text{src}} + \mathbf{u}_j^{\text{dest}} \cdot \mathbf{r}_k^{\text{dest}})\|_2^2$
    minimize the error between real value and predict value $error$ by updating the parameters $\mathbf{u}_i^{\text{src}}, \mathbf{r}_k^{\text{src}}, \mathbf{u}_j^{\text{dest}}, \mathbf{r}_k^{\text{dest}}$
Until stopping criteria met

Table 1. Statistics of Dataset

Dataset	#User	#Spammer	#Legitimate	#Relations
Tagged.com	$4,111,179$	$182,939$	$3,928,240$	$85,470,637$

3.2 Parameter Estimation

In general, Multi-Relational Embedding models cannot be determined by convex optimization, instead, approximation techniques are often used in practice. In this work, we adopt the Adam [12] optimizer, the parameters are learned by minimizing the loss function as follows:

$$\underset{(\mathbf{u}^{\text{src}},\mathbf{u}^{\text{dest}},\mathbf{r}^{\text{src}},\mathbf{r}^{\text{dest}})}{\operatorname{argmin}} \sum_{(u_i,u_j)\in\mathcal{U}} \sum_{r_k\in\mathcal{R}} \|\pi_{ijk} - (\mathbf{u}_i^{\text{src}} \cdot \mathbf{r}_k^{\text{src}} + \mathbf{u}_j^{\text{dest}} \cdot \mathbf{r}_k^{\text{dest}})\|_2^2, \tag{4}$$

where $||\cdot||_2^2$ is the L2 norm. The overall learning algorithm is summarized in Algorithm 1. We follow common practice by setting the stopping criteria as $error \leq 10^{-4}$.

4 Experimental Results

To evaluate the effectiveness of the proposed Multi-Relational Embedding model, experiments were conducted on a large real-world dataset from *Tagged.com*. Comparisons were made against several graph-based and sequence-based methods. Our algorithm was implemented on TensorFlow and experiment was conducted on a computer with 28 CPU cores and 256GB of memory.

4.1 Experimental Setup

Dataset The dataset used in this experiment was from *Tagged.com*, which is a website for people to meet and socialize with new friends. The dataset contains 7 types of directed relations, including *Message*, *Pet Game*, *Meet-Me Game Add Friend*, *Give a Gift*, *Report Abuse*, and *View Profile* However, the semantic meaning of each relation is not utilized as multi-relational spammer detection models should learn the importance of each relation from training data. The ground truth label is provided by domain experts to mark each user as legitimate or spam. The data is stored as quad-tuples: $\langle \text{timestamp}, u_i^{\text{src}}, u_j^{\text{dest}}, r_k \rangle$, where user u_i^{src} performs action (relation) r_k on user u_j^{dest}. We extracted all relations of a day, resulted in a dataset containing 85M interactions among 4M users. Out of these users, 182K of them are labeled as spammers, i.e., 4.45%. Statistics of the dataset is shown in Table 1.

Test Data Among the 7 types of relations, there exists a reporting relation that is provided by the *Report Abuse* mechanism. In this reporting relation, the user u_i^{src} reports user u_j^{dest} for violating the terms of conditions. However, a user who has been reported may or may not be a spammer. The collective detection framework [5] combines the

classification results with the report relation using the probabilistic soft logic (PSL) rule, in order to improve the security team's efficiency. Two important PSL rules proposed are:

$$\begin{aligned} &\text{Legitimate}(u_i^{\text{src}}) \wedge \text{Report}(u_i^{\text{src}}, u_j^{\text{dest}}) \rightarrow \text{Spammer}(u_j^{\text{src}}), \\ &\text{Spammer}(u_j^{\text{dest}}) \wedge \text{Report}(u_i^{\text{src}}, u_j^{\text{dest}}) \rightarrow \text{Legitimate}(u_i^{\text{src}}). \end{aligned} \tag{5}$$

The PSL rules limit the evaluation to users who appear in the reporting relation. To be consistent with related research, we adopted the same testing scheme by extracting users appeared in the reporting relation as our test data.

Evaluation Metrics Since the ground-truth label of each user is provided by the dataset, we adopt standard metrics (P-R-F), including precision (P), recall (R) and F-measure (F) to do evaluate the effectiveness the models. Furthermore, all metrics are computed on the class of spammers:

$$R = \frac{TP}{TP + FN},\ P = \frac{TP}{TP + FP},\ F = \frac{2PR}{P + R}, \tag{6}$$

where TP is the number of spammers that have been identified correctly, on contrast, FP is the number of spammers that have been identified mistakenly, and FN is the number of spammers that have been missed by the model. Depending on the application scenario, a trade-off can be made on these metrics. High precision represents for catching more spammers. Meanwhile, it will do harm to legitimates, as it takes more users as spammers. While, high recall represents for higher confidence on detected spammers, but may lead to more missing of some spammers. F-measure balances between precision and recall, and is suitable for general scenarios. As the main focus is to evaluate the quality of features extracted from multi-relational data instead of new classification algorithm, two classic but simple supervised models are selected: Logistic Regression (*LR*) and Gaussian Naive Bayes (*GNB*).

4.2 Performance Comparison

Several state-of-the-art graph-based and sequence-based features are chosen as the baselines, including *k-core* [1], *Graph Coloring* [9], *Page Rank* [19], *Weakly Connected Components* [20], *Degree* [5], and *Sequential k-gram Features* [5].

Graph-based features are computed using *Graphlab Create* [4] on each type of relation and resulted in a total of 56 graph-based features, i.e., 8 for each type of relation. For sequence-based features, we compute them using bigram sequence. With 7 relations in the dataset, we ended up with 49 bigram sequence-based features. In our multi-relational embedding features, we generated 30 features to do the overall comparison. Other scale of multi-relational embedding features will discuss later in this section.

After getting the baseline features, we split train and test dataset with 10 different random seeds for evaluation on *LR* and *GNB* classifiers. First, we compare our multi-relational embedding features with them separately. Then, we combine the baseline methods together to show the effectiveness of our proposed method.

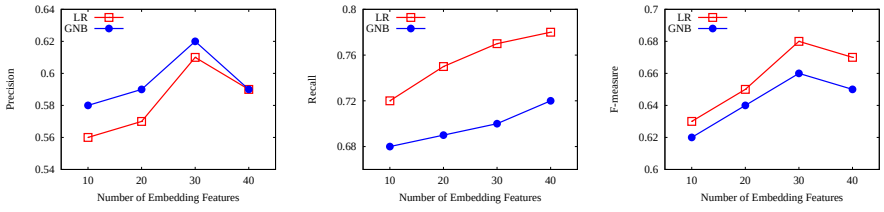
[4] `https://turi.com/`

Table 2. Comparison of Two Classifiers with Different Kinds of Features

	Logistic Regression			Gaussian Naive Bayes		
Features	Precision	Recall	F-measure	Precision	Recall	F-measure
Graph	0.4537	0.6390	0.5308	0.5978	0.3840	0.4675
Sequential	0.4907	0.8620	0.6253	0.4168	0.9320	0.5759
Graph+Sequential	0.5316	0.8600	0.6570	0.4571	0.9260	0.6120
MRE ($z = 30$)	**0.6138**	0.7730	**0.6844**	**0.6165**	0.7020	**0.6566**

Table 2 shows the comparison performance of different kinds of features. As can be seen, our multi-relational embedding features have shown a significant performance advantage over other features on F-measure both with *LR* and *GNB*, which means we can catch the spammer more accurately with the least harm to legitimates. Encouragingly, the precisions of embedding features consistently are the highest ones, giving the proof that the proposed features can reveal the most of spammers with a little loss in recalls. In terms of recall, although sequential features enjoy the highest position, they show the worst performance on precision as the price, which means they treat more users as spammers and greatly affect the legitimates.

To throughly examine the performance of the *Multi-Relational Embedding* model, we analyze its performance by varying the size of embedding space from 10 to 40. Fig. 3 shows the performance of each embedding features on precision, recall and F-measure separately. Obviously, the recall rate increases with the raise of dimension, giving the sign that more spammers will be disclosed when increasing the dimension of our *MRE* model. While, the precision and the F-measure reach their peaks at the dimension of 30, followed by a decline. That is to say, if the dimension keeps growing after reaching 30, the *MRE* model will lose its preciseness by listing more users as spammers. In general, it shows that the most effective performance has been achieved on 30 embedding features. Nevertheless, the number of embedding features depends on the dataset. One recommendation is that the number of embedding features should be increased alongside the number of types of relations, because more type of relations implies more complex interactions.



(a) Precision (b) Recall (c) F-Measure

Fig. 3. Impact of the number of dimensions (z) in our *MRE* model.

5 Related Work

In the literature, an extensive researches have been developed to extract abnormal behavior as features in social media, including e-commerce sites [6, 29] and social networks sites [15, 16]. The indicating features of spammers are depending on the available metadata, e.g., timestamps, text content, ratings, etc. Generally, it can be categorized into three parts: *content-based*, *behavior-based* and *topological* features. In early studies of email spams and e-commence spams, reviews/emails containing similar content have a high probability to be spams [10, 11]. Various of content-based features are designed to detect such spams in e-commerce and emails, e.g. average length in number of words [17], ratio of objective words [13]. In addition, behavior-based features are mainly generated considering the timestamps, sequence of time, ranks, distributions, etc. For instance, Fei *et al* [7] suggest that the ratio of Amazon verified purchases will somehow track spammers. Arjun *et al* [15, 16] proposed other behavior-based features focusing on timestamps and ranks. Fakhraei *et al* [5] raised a k-gram sequential feature with the help of Mixture Markov Model. Beside of individual spammers, groups of spammers also attract researchers' attention [16, 27, 28, 30], with the assumption that spammers within a group are more likely to attack legitimates together, which indicates that the relationships in social media might be useful to detect spammers. Along such mentality, topological features have been proposed in recent literature [5, 6], which usually consists of *degree*, *score of Page Rank*, *k-core* etc. However, the existing topological feature extraction methods often assume the data to be homogeneous, i.e., different types of user relations need to be modeled separately. This assumption limits the potential of topological methods as the interactions among different types of relations are not captured.

Graph structured embedding can help with the utilization of interactions among different relations, as it leverages relational learning methods [18] to extract the latent information of graph elements including both vertices and edges. Depending on the assumptions, each relational learning method proposes a different model to represent graph triple: two vertices and one edge. The models can be categorized into three categories: direct vector space translating, vector space translating with relation subspace, and tensor factorization. Considering of graph is a multi-relational heterogeneous network, Bordes et al. [3] proposed a bi-directed relation subspace mapping based model, which maps head vertex and tail vertex by two different matrices of one relation. Bordes et al. [2] proposed another model using direct vector space translating model, which ignore multi-relation problem but make the model much more efficient in training speed. Nickel et al. [18] and Socher et al. [25] proposed a new type of relational learning methods based on tensor factorization, which is efficient in both speed and accuracy. In present work, we extended the graph structured embedding method to our special case of a small number of relation types.

6 Conclusions

In this work we tacked the multi-relational spammer detection problem from graph perspective of view by proposing the *Multi-Relational Embedding* model. The *MRE*

model takes advantages of both the representational power of graph and the ease of modeling higher order interactions of embedding. Experiment results on public dataset have demonstrated the effectiveness of the *MRE* model by achieving improved spammer detection performance. For future work, the computational efficiency of *MRE* can be further improved by parallelization. This is feasible due to the fact that the full graph consists of many isolated subgraphs, i.e., the graph is not fully connected.

Acknowledgment

This work was supported in part by National Key Research and Development Program of China under Grant 2016YFB1000901, the National Natural Science Foundation of China (NSFC) under Grant 71571093, Grant 91646204, Grant 71372188, Grant 71701089, and the National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035.

References

1. Alvarez-Hamelin, J.I., Dall'Asta, L., Barrat, A., Vespignani, A.: Large scale networks fingerprinting and visualization using the k-core decomposition. In: Advances in Neural Information Processing Systems. pp. 41–50 (2006)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: International Conference on Neural Information Processing Systems. pp. 2787–2795 (2013)
3. Bordes, A., Weston, J., Collobert, R., Bengio, Y., et al.: Learning structured embeddings of knowledge bases. In: AAAI. vol. 6, p. 6 (2011)
4. Cheng, Z., Gao, B., Sun, C., Jiang, Y., Liu, T.Y.: Let web spammers expose themselves. In: ACM International Conference on Web Search and Data Mining. pp. 525–534 (2011)
5. Fakhraei, S., Foulds, J., Shashanka, M., Getoor, L.: Collective spammer detection in evolving multi-relational social networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1769–1778. ACM (2015)
6. Fayazi, A., Lee, K., Caverlee, J., Squicciarini, A.: Uncovering crowdsourced manipulation of online reviews. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 233–242 (2015)
7. Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: Exploiting burstiness in reviews for review spammer detection. ICWSM pp. 175–184 (2013)
8. Hu, X., Tang, J., Liu, H.: Online social spammer detection. In: AAAI. pp. 59–65 (2014)
9. Jensen, T.R., Toft, B.: Graph coloring problems. Wiley (1995)
10. Jindal, N., Liu, B.: Analyzing and detecting review spam. In: International Conference on Data Mining (ICDM). pp. 547–552. IEEE (2007)
11. Jindal, N., Liu, B.: Review spam detection. In: International Conference on World Wide Web. pp. 1189–1190. ACM (2007)
12. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. Computer Science (2014)
13. Li, F., Huang, M., Yang, Y., Zhu, X.: Learning to identify review spam. In: International Joint Conference on Artificial Intelligence (IJCAI). vol. 22, pp. 2488–2493 (2011)
14. Liu, H., Zhang, Y., Lin, H., Wu, J., Wu, Z., Zhang, X.: How many zombies around you? In: International Conference on Data Mining (ICDM). pp. 1133–1138. IEEE (2013)

15. Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., Ghosh, R.: Spotting opinion spammers using behavioral footprints. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 632–640 (2013)
16. Mukherjee, A., Liu, B., Glance, N.: Spotting fake reviewer groups in consumer reviews. In: International Conference on World Wide Web. pp. 191–200. ACM (2012)
17. Mukherjee, A., Venkataraman, V., Liu, B., Glance, N.S.: What yelp fake review filter might be doing? In: ICWSM (2013)
18. Nickel, M.: Tensor factorization for relational learning. Ludwig-Maximilians-Universitt Mnchen (2013)
19. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking : Bringing order to the web. Stanford Digital Libraries Working Paper pp. 1–14 (1998)
20. Pemmaraju, S.V., Skiena, S.S.: Computational discrete mathematics : combinatorics and graph theory with mathematica. Cambridge University Press (2009)
21. Peng, F., Schuurmans, D., Wang, S.: Augmenting naive bayes classifiers with statistical language models. Information Retrieval pp. 317–345 (2004)
22. Pitsillidis, A., Levchenko, K., Kreibich, C., Kanich, C., Voelker, G.M., Paxson, V., Weaver, N., Savage, S.: Botnet judo: Fighting spam with itself. In: Network and Distributed System Security Symposium (2010)
23. Rayana, S., Akoglu, L.: Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 985–994. ACM (2015)
24. Schank, T.: Algorithmic aspects of triangle-based network analysis. Phd in Computer Science, University Karlsruhe (2007)
25. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: International Conference on Neural Information Processing Systems. pp. 926–934 (2013)
26. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Computer Security Applications Conference. pp. 1–9. ACM (2010)
27. Wang, Y., Wu, Z., Bu, Z., Cao, J., Yang, D.: Discovering shilling groups in a real e-commerce platform. Online Information Review pp. 62–78 (2016)
28. Wu, L., Hu, X., Morstatter, F., Liu, H.: Adaptive spammer detection with sparse group modeling. In: The International AAAI Conference on Web and Social Media. pp. 319–326 (2017)
29. Wu, Z., Wang, Y., Wang, Y., Wu, J., Cao, J., Zhang, L.: Spammers detection from product reviews: A hybrid model. In: IEEE International Conference on Data Mining. pp. 1039–1044. IEEE (2016)
30. Yu, R., He, X., Liu, Y.: Glad: group anomaly detection in social media analysis. In: International Conference on Knowledge Discovery and Data Mining. pp. 372–381. ACM (2014)
31. Zhou, B., Pei, J.: Link spam target detection using page farms. ACM Transactions on Knowledge Discovery from Data (TKDD) pp. 1–38 (2009)