Spam2Vec: Learning Biased Embeddings for Spam Detection in Twitter

†Suman Kalyan Maity, ‡Santosh K C, ‡Arjun Mukherjee
†Dept. of CSE, IIT Kharagpur, India ‡Dept. of CS, University of Houston, TX, USA
sumankalyan.maity@cse.iitkgp.ernet.in,syantokc@gmail.com,arjun@cs.uh.edu

ABSTRACT

In this paper, we propose a semi-supervised framework *Spam2Vec* to identify spammers in Twitter. This algorithmic framework learns the spam representations of the node in the network by leveraging biased random walks. Our spammer detection method yields an AUC of 0.54 with precision@100 as 0.12 and performs significantly better with 7.77% increase in AUC and a 2.4 times improvement on precision over the best performing baseline.

CCS CONCEPTS

• **Information systems** → **Data mining**; **Social networks**;

KEYWORDS

Spam detection, Biased embedding, Biased Random walks

ACM Reference Format:
†Suman Kalyan Maity, ‡Santosh K C, ‡Arjun Mukherjee. 2018. Spam2Vec: Learning Biased Embeddings for Spam Detection in Twitter. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3184558.3186930

1 INTRODUCTION

Social spammers are sophisticated and adaptable. Reflexive reciprocity makes it easier for social spammers to establish social influence and pretend to be normal users by accumulating a large number of friends and thereby easily bypass the spam detection systems. In this paper, we present *Spam2Vec*, a framework to collectively use both content and network information for social spammer detection. *Spam2Vec* learns a biased spam embeddings in the network by leveraging biased random walks. We first calculate follow-spam scores of the nodes in the network and try to bias the random walks with follow-spam scores of the nodes together with spam related features of the nodes. The biasing methodology maximizes the likelihood of obtaining spammer nodes in local few hop neighborhood instead of just concentrating on the immediate neighbor.

In Twitter, there are limited attempts been made to tackle the spam detection problem [1, 3, 6, 10, 12]. Lee et al. [6] leveraged profile-based features and deployed social honeypots to detect new social spammers. Ghosh et al. [3] studied link farming in Twitter. Zhu et al. [12] propose a Supervised Matrix Factorization method with Social Regularization for spammer detection.

2 MODEL DESCRIPTION

We use the Twitter dataset collected by Yang et al. [11] consisting of posts from 17 million users from June 2009 to December 2009. We extracted the follower-following topology of Twitter from [5]. We further prune the network and we are left with 4,405,698 users and separately crawled the status of users to identify if they were suspended or not. In total, we have 100,758 spammer accounts.

Our entire framework is composed of 3 components: a) Follow-spam, b) Biasing in the network, c) Learning Spam Representation in the network.

2.1 Follow-spam

One of the prominent ways of spamming in Twitter is follow-spam where a Twitter user follows large number of unknown other users hoping that these pretend-friends will follow him back in exchange. In this module, we design a pagerank-like model inspired by [3] to rank the nodes in the who-follows-whom network based on their spamicity (see Algorithm 1).

Algorithm 1 Follow-spam scores

```
Input: Who-follow-whom network $G(V, E)$
Output: follow-spam scores $f$
  for all nodes $v \in V$ do
    $f^0(v) \leftarrow \frac{1}{|V|}$ /*Initialize the follow-spam scores of nodes*/
  end for
  $t \leftarrow 1$
  while not converged do
    for all nodes $v \in V$ do
      $f^t(v) \leftarrow (1-\alpha)f^0(v)$
      $+\alpha * \sum_{p \in followings(v)} \frac{f^{t-1}(v)*nrf(r)}{\sum_{r \in follower(p)} nrf(r)}$
      /*nrf($r$) means non-reciprocal followings of node $r$*/
    end for
    $t \leftarrow t+1$
  end while
```

2.2 Biasing in the network

We want to combine the follow-spam scores and also consider the spammer's properties into a single framework that will at the same time consider rich node and edge features as well as the structure of the network. From a given source node s and set of spammer nodes (S) in the network, we aim to bias the random walk originating from s (irrespective of whether it is a spammer node or not) so that it visits other spammer nodes more often than the non-spammer nodes (H) in the network. For edge (u, v) in the network, we compute the edge strength $a_{uv} = \psi_w(\phi_{uv})$ where ϕ_{uv} denotes the corresponding feature vector that describes the nodes u, v and their interaction. Function ψ_w parameterized by w takes the edge feature vector ϕ_{uv} as input and computes the corresponding edge strength a_{uv} which models the biased random walk transition probability. Therefore, we need to set the parameters w of function

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186930

$\psi_w(\phi_{uv})$ so that it will assign edge weights a_{uv} in such a way that a random walker will more likely visit spammer nodes S than non-spammer nodes H. Towards this objective, we formulate the optimization problem to find the optimal set of parameters w of edge-weight objective function $\psi_w(\phi_{uv})$ as follows:

$$\begin{aligned} &\min_{w} \Omega(w) = ||w||^2 \quad \text{such that} \\ &\forall s \in S \& t \in \Gamma(s),\ \psi_{s,t \in H} < \beta_1 \psi_{s,t \in S} \\ &\text{and} \quad \forall h \in H \& t \in \Gamma(h),\ \psi_{h,t \in H} < \beta_2 \psi_{h,t \in S} \end{aligned} \tag{1}$$

Note that $\psi_{u,v}^{OPT} = \phi_{uv}.w^{OPT}.(f_u + f_v)^{\gamma}$ where f_u and f_v are the follow-spam scores of node u and v respectively and w^{OPT} is the optimal w vector. We further bias the random walk with neighborhood spam. The idea behind this is that when a walker lands up on a node, he will choose a node that increases the likelihood of finding a spammer in his local neighborhood.

2.3 Learning Spam Representation

We now want to learn the spam representations (network embeddings) of the nodes in the network. For each node u in the network $G = (V, E)$, we define a proximity or neighborhood $N(u)$ which can be obtained by simulating a biased random walk (defined above) on the network starting at node u. We optimize the following objective function that predicts which nodes belong to the neighborhood $N(u)$ based on the learned node attributes y, by adopting the Skip-gram architecture [4, 7, 9].

$$\max_{y} \sum_{u \in V} log(\prod_{n \in N(u)} P(n|y(u))) \tag{2}$$

We also make the assumption that a source node and neighborhood node have a symmetric effect over each other in representation space. So, we have

$$P(n|y(u)) = \exp(y(n).y(u)) / \sum_{v \in V} \exp(y(v).y(u)) \tag{3}$$

Since the per-node function $\sum_{v \in V} \exp(y(v).y(u))$ is computationally expensive, we approximate it using negative sampling [8]. We then optimize the objective function mentioned in Eq.2 using stochastic gradient descent over the model parameters. The detailed methodology is presented as Algorithm 2.

Algorithm 2 Spam representation model

```
Input: Graph $G = (V, E, W)$, Dimensions $d$, Biased Walks per node $b$, Walk length $l$, context size $k$
  /*The edge strengths $a_{uv}^{final}$ are provided as weights $W^*$*/
Output: Spam representation $y$
  $biased_{walks} = \{\}$ /*Initialize the $biased_{walks}$ set to empty*/
  for iter = 1 to b do
    for all nodes $v \in V$ do
      $walk = BiasedWalk(G, v, l)$
      Append walk to $biased_{walks}$
    end for
  end for
  $y = SGD(k, d, biased_{walks})$
```

3 EXPERIMENTS

We use the node representations learnt earlier as features for spammer detection task. We learn a SGD regressor giving spammer nodes and non-spammer nodes two distinct values and then rank the nodes in test set according to the regression values. We evaluate our model with several baseline models to see how they are performing against various evaluation metrics.

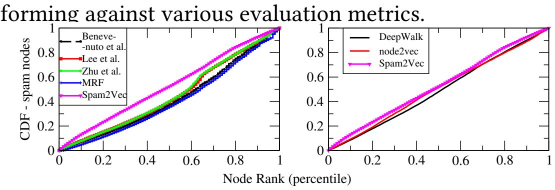

Figure 1: Cumulative Distribution Function of spam nodes

In fig 1, we present the cumulative distribution function for presence of spam nodes in the rank percentile of nodes. We can observe that *Spam2Vec* performs best followed by node2vec. We also calculated area under the above curve (see table 1). Our model performs significantly better both in terms of Area under the CDF curve as well as precision@n (n=100). *Spam2Vec* yields an AUC of 0.541 with precision@100 as 0.12 which outperforms the best performing baseline model (*node2vec*) by 7.77% and 2.4 times increase in AUC and precision respectively. For other values of n also, *Spam2Vec* performs consistently and better than the other baseline models.

Table 1: Evaluation Results: AUC and Precision @100

Models	AUC	P @100	Models	AUC	P @100
Markov Random Field[2]	0.38	0.02	DeepWalk[9]	0.488	0.05
Benevenuto et al. [1]	0.42	0.06	node2vec[4]	0.502	0.05
Lee et al. [6]	0.452	0.07	**Our Model** *Spam2Vec*	**0.541**	**0.12**
Zhu et al. [12]	0.454	0.03			

4 CONCLUSIONS

We propose a network-cum-content based spam-represented embedding learning framework *Spam2Vec*. We boost our spam representation learning of the node in the network by leveraging biased random walks. We compare our method *Spam2Vec* with already existing baselines. *Spam2Vec* yields an AUC of 0.54 with precision@100 as 0.12 and performs significantly better with 7.77% increase in AUC and a 2.4 times improvement on precision over the best performing baseline.

5 ACKNOWLEDGMENTS

This research work is supported in part by the National Science Foundation under grant no. 1527364.

REFERENCES

[1] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *CEAS*, volume 6, page 12, 2010.
[2] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. *ICWSM*, 13:175–184, 2013.
[3] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi. Understanding and combating link farming in the twitter social network. In *WWW*, pages 61–70, 2012.
[4] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *ACM SIGKDD*, pages 855–864. ACM, 2016.
[5] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, pages 591–600, 2010.
[6] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *SIGIR*, pages 435–442, 2010.
[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ICLR*, 2013.
[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
[9] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *ACM SIGKDD*, pages 701–710. ACM, 2014.
[10] K. Santosh, S. K. Maity, and A. Mukherjee. Enwalk: Learning network features for spam detection in twitter. In *SBP-BRiMS*, pages 90–101, 2017.
[11] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186. ACM, 2011.
[12] Y. Zhu, X. Wang, E. Zhong, N. N. Liu, H. Li, and Q. Yang. Discovering spammers in social networks. In *AAAI*, pages 171–177, 2012.