Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Discovering Spammers in Social Networks

Yin Zhu[†], Xiao Wang[⋆], Erheng Zhong[†], Nanthan N. Liu[†], He Li[⋆], Qiang Yang[†]
[†]Hong Kong University of Science and Technology, Hong Kong
[⋆]Renren Inc., China
{yinz,ezhong,nliu,qyang}@cse.ust.hk, {xiao.wang, he.li}@renren-inc.com

Abstract

As the popularity of the social media increases, as evidenced in Twitter, Facebook and China's Renren, spamming activities also picked up in numbers and variety. On social network sites, spammers often disguise themselves by creating fake accounts and hijacking normal users' accounts for personal gains. Different from the spammers in traditional systems such as SMS and email, spammers in social media behave like normal users and they continue to change their spamming strategies to fool anti-spamming systems. However, due to the privacy and resource concerns, many social media websites cannot fully monitor all the contents of users, making many of the previous approaches, such as topology-based and content-classification-based methods, infeasible to use. In this paper, we propose a Supervised Matrix Factorization method with Social Regularization (**SMFSR**) for spammer detection in social networks that exploits both social activities as well as users' social relations in an innovative and highly scalable manner. The proposed method detects spammers collectively based on users' social actions and social relations. We have empirically tested our method on data from Renren.com, which is one of the largest social networks in China, and demonstrated that our new method can improve the detection performance significantly.

Introduction

Users are fed up with spam. Our email boxes are filled with spam messages. Our Facebook pages are approached by fake accounts trying to seize our privacy, and to send us unwanted information. Due to its seriousness, spammer detection has attracted a lot of attention in research ever since the advent of the Web. Major research topics in spamming detection include spamming email detection (Blanzieri and Bryl 2008), spamming Web page detection (Gyöngyi and Garcia-Molina 2005), and spamming instant message detection (Xu et al. 2012; Liu et al. 2006). Recently, the success of social media such as Facebook, Twitter and Renren also attracted a new way of spamming: Social Networking Spam (Brown et al. 2008). In this new type of spamming, spammers create fake accounts to seize private information or to promote commercial advertisements in a social network for personal gains, which threaten the quality of social networking. A main challenge in detecting social spamming accounts is that the possible social activities and behavior have more varieties than before, and they constitute a much larger feature space. As a result, they are more difficult to detect. Added to this issue is the concern for user privacy; users do not like an anti-spammer system to constantly scan the content of their social media. Another challenge in detecting social spammers is that their behaviors change too fast to be detected by a traditional anti-spamming system that is based on extensive offline model building. A system that is capable of capturing most of the spamming accounts this month may fail to do so next month, because the spammers get smarter and can create new, more evasive accounts to avoid being detected. Thus, social spamming detection is a new challenge for researchers.

In the past, researchers have tried different techniques to detect spammers: link analysis (Li and Hsieh 2006) and content analysis (Blanzieri and Bryl 2008). Some of these methods have very high precision – when they predict that an account is fake, it has a very high probability of being truly fake. However, we have also found that many of these existing methods suffer from low recall in the real applications; that is, a lot of spamming accounts are not considered as spammer candidates in the first place, leaving these accounts at large. Typically, a social networking site will identify these evasive accounts only when normal users tip them off via customer support; however, by which time, users are already annoyed and unhappy. The low-recall issue is partly due to the quick behavior change ability of spammers. Once the spammers see that their fake accounts are caught by the system, and they come up with a new strategy to deceive the system. Thus, we believe that a more sustainable way to detecting spammers in a social network is to detect them based on their social activities. We do not want to rely on content analysis because if content analysis were conducted for each of the myriad activity types in Social Network Services (SNS), it would result in an enormous number of features, which made learning extremely hard due to the very limited amount of labeled data. In this paper, we try to encode the users' social activity with a user-activity count matrix. Each row of the matrix is a high dimensional sparse activity count vector for a particular user, and the columns correspond to different types of social activities.

To address these challenges, we propose a novel framework that seamlessly integrates feature extraction from social activities with classification model for spammer prediction. We conduct our study using a large scale data set from Renren (www.renren.com), one of the largest SNSs in China, and demonstrate significant success with the proposed method. Our major contributions are as follows:

1. We propose a joint optimization model that simultaneously does feature extraction and classifier learning. We use a matrix factorization model to collaboratively induce a succinct set of latent features for different users, and this latent feature learning process is guided by the social relationship graph and the label information. We call our proposed method Supervised Matrix Factorization with Social Regularization (**SMFSR**). The induced features are then used as the input representation for a spammer classification model. These two steps are solved jointly, which enables the latent features to be induced by taking into account its effectiveness in classifying the labeled spammers.
2. We conduct extensive analysis of a large scale data set from a realworld SNS, and find several important insights about the behaviors of social spammers. In particular, we find that the neighbors of a spammer in a SNS actually consist mostly of other normal users. In other words, the spammers do not form a cluster themselves and they are well integrated into a larger social network. This observation is in contrast to previous studies, e.g. (Li and Hsieh 2006), which found spammers to be densely connected to each other.
3. We evaluate our method on a large scale realworld social network data set based on human assessed spammers and demonstrate promising performances in detecting social spammers. To the best of our knowledge, our study in this paper is the first systematic analysis on spammer detection in a complex SNS where user activity space is very large.

Related Work

Spammer detection in message systems Spammer detection has been studied in message systems for many years, from email systems (Blanzieri and Bryl 2008), to SMS systems (Liu et al. 2006; Xu et al. 2012), and most recently to microblogging websites such as Twitter (Benevenuto et al. 2010; Lee, Eoff, and Caverlee 2011). A common characteristic of message systems is that the number of activities is very limited. The network structure is also quite different from Facebook or Renren, e.g. the difference between Facebook and Twitter is described in (Kwak et al. 2010). These two reasons make existing approaches developed in message systems not work well for Facebook or Renren.

Spammer detection in complex SNS Due to the data availability issue, the spammer detection on complex SNS such as Facebook and Renren has not been studied carefully before. In (Lee, Caverlee, and Webb 2010), the authors proposed a honeypot based method to attract spammers to add honeypots as friends. But a spammer study on Renren social network (Yang et al. 2011) shows that spammers add their friends in a snowball-effect style, i.e. most spammers add popular users in the social network and gradually add more normal users and finally they are integrated into the social network as normal users. It also requires the honeypots to be popular in the social network to make this method work; otherwise the honey-spots cannot attract enough spammers. Recently, Facebook (Stein, Chen, and Mangla 2011) publishes a report on their immune system, in which a classification based method is used for spammer detection. However, no technical detail is provided on, e.g. what feature extraction method is used. To the best of our knowledge, our study in this paper is the first systematic analysis on spammer detection in a complex SNS where user activity space is very large.

Matrix Factorization Our main technique in this paper is matrix factorization. Matrix factorization is a family of algorithms that are well suitable for a lot of machine learning applications, e.g. Latent Semantic Indexing (LSI) (Landauer et al. 2007), and Collaborative Filtering (CF) (Koren 2008). One advantage of factorization is to represent the knowledge in a compact form. After the factorization, the the original matrix, which is usually very sparse, is approximated as the product of several compact matrices. In our classification setting, the compact matrices could be viewed as a latent feature set of users or activities. Another advantage is that it can incorporate other information sources, e.g. labeling information (Zhu et al. 2007) and social relationship (Ma et al. 2011). Specifically, (Ma et al. 2011) utilizes the social relations to generate a manifold regularization term for collaborative filtering. But the way to encode social relation in this paper is different from the one in (Ma et al. 2011), which requires all users should have similar latent features to their neighbors regardless of spammers or normal users. In addition, these previous methods are applied on collaborative filtering, which cannot be adapted to social spammer detection ideally.

Problem Formulation

We formulate the problem in this section. We aim to detect spammers based on three kinds of heterogeneous knowledge, small amount of labeled data (whether one user is spammer or not), users' social activities and users' social relations. Let $\mathcal{U} = \{u_i\}_{i=1}^{n}$ denote the user set, where n is the number of users. Let $Y = (u_i, y_i)_{i=1}^{\ell}$ denote the labeled data, where $y_i \in \{-1, +1\}$. If one user u_i is spammer, $y_i = +1$, otherwise, $y_i = -1$. We represent users' social activities as a matrix $A \in \mathbb{R}^{n \times m}$, where m is the number of different user activities. A_{ik} denotes the number of activity k that user u_i has performed. There are two main kinds of activities users can perform in the social network: activities performed on his/her own social page, and interactive activities with other users. The second kind is the majority. We also consider users' social relations. Formally, let R denote users' relation network, where $R_{ij} = 1$ denotes user u_i and user u_j are friends. The friendship graph is undirected. Generally, our

Table 1: Definition of notations

Notation	Notation Description
$\mathcal{U} = \{u_i\}_{i=1}^{n}$	User set
$\mathcal{V} = \{v_i\}_{i=1}^{m}$	Activity set
n	Number of users
m	Number of social activities
$Y = (u_i, y_i)_{i=1}^{\ell}$	Labeled data
ℓ	Number of labeled data
A	User activity matrix
A_i	i-th row in A, user u_i's activity count vector
R	User relation matrix
$N(u_i)$	User u_i's friends
U	User activity latent matrix
V	Activity latent matrix
$\Sigma, \mathbf{w}$	Classification model coefficients
K	Number of latent dimensions

task is to build a binary classifier which predicts whether a unlabeled user is a spammer or not. The notations are summarized in Table 1.

Our Proposed Approach

In this section, we present the proposed algorithm for spammer detection. We first introduce the basic matrix factorization technique and the corresponding optimization algorithm. After that, we propose a socially regularized matrix factorization method for spammer detection using users' social interactions (John sent a greetings to Mary) and social relations (John and Steve are friends). Finally, we incorporate supervised knowledge together to build a more comprehensive model.

Matrix Factorization (MF)

Since the social activities are sparse, building models using the original activities may fail to predict the label of one user precisely. We propose to factorize the activity matrix A into two latent matrices U and V, which represent the latent factors of users and activities respectively.

$$\begin{aligned} \min \mathcal{J}(U,V) &= \sum_{a_{ij} \in A} I_{ij}(a_{ij} - U_i V_j)^2 \\ &+ \frac{\lambda_f}{2}(||U||_F^2 + ||V||_F^2) \end{aligned} \tag{1}$$

where I_{ij} is an indicator which equals 1 if the corresponding element in the matrix is not empty, otherwise 0. Note that only nonempty entities are considered in the optimization process, since we aim to detect spammers based on observed activities. The first term is called the approximating loss. The three squares of the Frobenius Norm terms are the regularization part to avoid overfitting the factorization. The Frobenius Norm of a matrix M is defined as

$$||M||_F = \sqrt{\sum_{ij} M_{ij}^2}.$$

The regularization coefficient, λ_f, is to tradeoff between the approximating loss and the regularization terms. After the decomposition, the latent matrix U can be exploited to represent users. In addition, we can apply SVM or other classifiers on U to build models. One advantage of building models based on U instead of A is that matrix factorization can effectively reduce the dimensionality of feature space. Furthermore, unlabeled users' activity information can also be utilized and then make the built models more robust.

MF with Social Regularization

The basic MF only considers the social activity information, which cannot robustly handle cold start users with few observed activities. However, users are connected in the social network and the links between users reflect users' closeness. Thus, one intuitive solution is to exploit users' social relations to regularize the decomposition of the activity matrix A. This motivation is based on two observations. First, for the normal users, they perform similarly with their neighbors. Second, for the spammer, they perform differently from their neighbors and most of their neighbors are normal users. These two observations motivate us to design a social regularization term as follows.

$$\mathcal{R}_s = \sum_{u_i} \sum_{u_j \in N(u_i)} \hat{y}_i (U_i - U_j)^2 \tag{2}$$

where $\hat{y}_i = -1$ if u_i is spammer and $\hat{y}_i = 1$ otherwise. R_s has the following intuitive meaning: for each spammer, its latent factors should be different from its neighbors in the social network, most of whom are normal users. But for the normal users, their factors should be similar to their neighbors since they share similar interests and may perform similar social activities. Then the new objective with the social regularization becomes

$$\begin{aligned} \mathcal{J}_s(U,V) &= \sum_{a_{ij} \in A} I_{ij}(a_{ij} - \sum_{f=1}^{K} U_{if} V_{jf})^2 \\ &+ \frac{\lambda_f}{2}(||U||_F^2 + ||V||_F^2) + \frac{\lambda_s}{2}\mathcal{R}_s \end{aligned} \tag{3}$$

From machine learning aspect, The importance of $\mathcal{R}_s$ is that it can help avoid the overfitting on U and V, since the observations of the social activities and the labeled data on spammers may be sparse. We call this algorithm as Matrix Factorization with Social Regularization (MFSR).

SMFSR: MFSR with Supervised Information

After we get the latent factors for each user, we can use these vectors as the features and train a supervised model using these features. However, one problem is that, the supervised information has not been considered in the factorization process, which may lead the latent factors fail to capture the key knowledge to detect spammers. A more coherent method is to combine the factorization and the classification processes into a unified framework so that the factorization is not only guided by the approximating loss but also by the classification loss. Inspired by the Collective Matrix Factorization (Singh and Gordon 2008), we plug a classification loss term to the basic matrix factorization in Eq.(3). We choose the popular hinge loss used in Support Vector Machines (SVM).

However, the original hinge loss, $h_{svm}(z) = \max(0, 1-z)$, is not smooth at $z = 1$. To make the gradient computation and the subsequent optimization more tractable, we use the smoothed hinge loss (Rennie 2004):

$$h(z) = \begin{cases} \frac{1}{2} - z & z \leq 0 \\ \frac{1}{2}(1-z)^2 & 0 < z < 1. \\ 0 & z \geq 1 \end{cases} \quad (4)$$

We then define a classifier based on the activity factors as $f(u_i) = \text{sign}(\mathbf{w}_u^T U_i)$ and the new optimization objective becomes

$$\begin{aligned} \mathcal{J}_s(U,V,\mathbf{w}) = & \sum_{a_{ij} \in A} I_{ij}(a_{ij} - \sum_{f=1}^{K} U_{if} V_{jf})^2 \quad (5) \\ & + \frac{\alpha}{2} \sum_{i=1}^{\ell} h(y_i(\mathbf{w}^T U_i)) + \frac{\lambda_w}{2} ||\mathbf{w}||_2^2 + \frac{\lambda_s}{2} \mathcal{R}_s \\ & + \frac{\lambda_f}{2} (||U||_F^2 + ||V||_F^2 + ||W||_F^2) \end{aligned}$$

where α is the tradeoff coefficient between the factorization loss and the classification loss, and the $\mathbf{w}$ is the coefficient vector for user latent factors. We refer to this algorithm as Supervised Matrix Factorization with Social Regularization (**SMFSR**). We can update the latent matrices U and V, and the classification model parameters $\mathbf{w}$ using gradient based methods. Note that, although we apply a linear classifier $\mathbf{w}^T U_i$ to the latent user features, non-linear classifiers can be applied using kernel trick. For example, we can define $f(u_i)$ as

$$f(u_i) = \sum_{j=1}^{\ell} \alpha_j \kappa(U_i, U_j) y_j \quad (6)$$

where $\kappa(\cdot,\cdot)$ is a kernel, e.g. Gaussian kernel is defined as $\kappa(\mathbf{x},\mathbf{y}) = \exp(-\sigma(\mathbf{x}-\mathbf{y})^2)$.

Optimization

We optimize above objective functions using a stochastic gradient descent algorithm (SGD) (Zinkevich, Smola, and Langford 2009), which belongs to a class of hill-climbing optimization techniques that seeks a stationary point of a function. To utilize the SGD, we derive the gradients of each variables in the Eq. 5 as follows.

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial U_i} &= \sum_{a_{ij} \in A} (a_{ij} - \sum_f U_{if} V_{jf}) \cdot V_j + \alpha h'(y_i(\mathbf{w}^T U_i)) \mathbf{w}^T y_i \\ &+ \lambda_s \sum_{u_j \in N(u_i)} \hat{y}_i (U_i - U_j) + \lambda_f U_i \quad (7) \\ \frac{\partial \mathcal{J}}{\partial V_i} &= \sum_{a_{ij} \in A} (a_{ij} - \sum_f U_{if} V_{jf}) \cdot U_j + \lambda_f V_i \\ \frac{\partial \mathcal{J}}{\partial \mathbf{w}} &= \alpha \sum_i h'(y_i(\mathbf{w}^T U_i)) U_i y_i + \lambda_w \mathbf{w}, \end{aligned}$$

where the gradient of the smoothed hinge loss $h(z)$ is

$$h'(z) = \begin{cases} -1 & z \leq 0 \\ z - 1 & 0 < z < 1. \\ 0 & z \geq 1 \end{cases} \quad (8)$$

Algorithm 1 SGD for **SMFSR**

1: **Input**: social relation matrix: R, social activity matrix A, labeled data Y, number of latent features K, learning rate η and maximal number of iterations I
2: **Output**: User latent matrix U, activity latent matrix V
3: Generate U and V randomly
4: **for** $i = 1$ to I **do**
5: $\quad U \leftarrow U + \eta \frac{\partial \mathcal{J}}{\partial U}$
6: $\quad V \leftarrow V + \eta \frac{\partial \mathcal{J}}{\partial V}$
7: $\quad \mathbf{w} \leftarrow \mathbf{w} + \eta \frac{\partial \mathcal{J}}{\partial \mathbf{w}}$
8: $\quad$ **IF** convergence **break**
9: **end for**
10: **Return** U, V and $\mathbf{w}$

The optimization process can be found in Algorithm 1. The complexity of one iteration (lines 5-7) is $O(nPK + nAK)$, where n, P, A, K are the number of users, the average number of friends of a user, the average number of non-empty activities of a user, and the number of latent factors respectively. Usually, P is less than 500, A is less than 50, and K is set to 10 to 30 in our experiments. If we consider P, A and K as constants, one iteration costs only linear time proportional to the number of users. Generally, MF-based algorithm optimized by SGD is very scalable (Koren 2008; Rendle 2012).

Experiments

Dataset and Summary

To empirically study the effectiveness of our social-activity based spammer detection framework, we use a community detection algorithm (Duch and Arenas 2005) to extract a dataset from a realworld social network, `Renren.com` (NASDAQ: RENN). Renren is one fo the largest SNSs in China and has over 150 million registered users with over 15 million daily active users. Similar to Facebook, Renren is an undirected social network with a mature application platform to support various social applications, most of which are developed by third-party companies. More than Facebook, Renren has its own vertical interest groups and is embedded with multiple kinds of living services. Therefore the user activity in Renren is more complicated than Twitter-like social networks.

We extract a large community in Renren, and select 30,000 active normal accounts during one week, from 2 December 2011 to 8 December 2011. We then select 700 spammer accounts, which have social interactions with the selected normal users. Part of the spammers are caught by Renren's immune system, which is similar to the one reported by Facebook (Stein, Chen, and Mangla 2011). The rest of the spammers, which are considered to be more cunning, are tipped by normal Renren users at the customer support. For normal users, we use their activities during this week. Because spammers are caught at different time periods, we extract one-week of their activities right before they were caught. Each account has at least five actions. In this paper, we use 1,680 different activities in total, including all common activities and a few popular social ap-

Table 2: Dataset Summary

Dataset	*# of spammers*	*# of normal users*
Training	400	1,200
Testing	300	28,800

plications' usage. The top five activities are: *Visit-Album* (17.9%), *Show-Visit-Bulletin* (11.9%), *Visit-Blog* (9.9%), *Share/Retweet* (9.7%), and *Friend-Apply* (2.1%). Each ratio indicates the portion of the corresponding activities over all performed counts The top five activities account for 51.5% of all the activities. We find that the activity counts depict a power law distribution, i.e. some activities have large counts and most others have small counts. For each activity, we sort the different counts among the 30,700 users, make 10 bins, and normalize the numbers to 1 to 10. This normalization method is reminiscent of the median filter in signal processing, and is more robust to extreme values than mean filters.

To validate the social regularization assumption we made in the previous sections, we first count the number of normal friends and the number of spammer friends of each spammer and draw the CDF of these two numbers over all spammer accounts. As shown in Figure 1, over 70% of spammer accounts do not have any spammer friends while the average number of normal users a spammer account connects to is about 20. It is safe to make the assumption that a spammer's friends are normal users. We also perform Kernel Density Estimation [1] of the distance between the activity count vector of a user u_i (A_i) and the mean activity vector of his friends ($\frac{1}{|N(u_i)|}\sum_{u_j \in N(u_i)} A_j$):

$$\|A_i - \frac{1}{|N(u_i)|}\sum_{A_j \in N(u_i)} A_j\|. \quad (9)$$

We then compare the distribution of this distance for spammers and normal users respectively in Figure 2. We can see that the density function for normal users peaks at around 9, while the density function for spammers peaks at around 13. Therefore this observation validates our social regularization assumption that a spammer behaves more or less differently to its friends than a normal user.

Evaluation Protocol

We split the dataset into a training set and a testing set and the statistics of which are shown in Table 2. Following the traditional evaluation setting in spam detection, we use precision, recall and F1-score as our evaluation measures. By treating spammer accounts as positive samples in the binary classification, precision is defined as

$$\text{precision} = \frac{tp}{tp+fp},$$

and recall is defined as

$$\text{recall} = \frac{tp}{tp+fn},$$

[1] We use R's `density` function for Kernel Density Estimation.

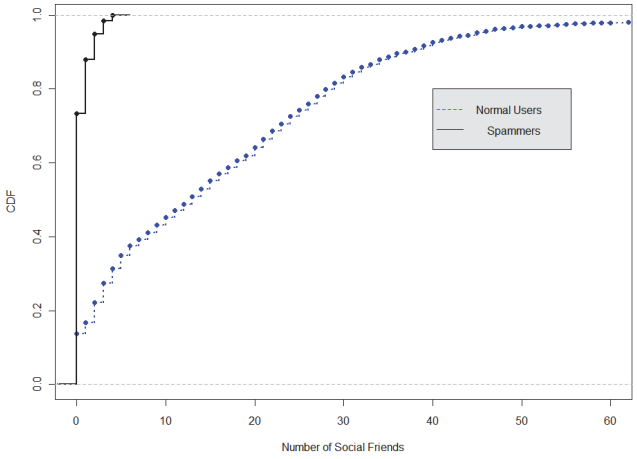

Figure 1: Spammer degree distribution over normal users nodes and spammer nodes. Most spammers ($> 70\%$) have no connections to other spammers.

where tp, fp and fn are numbers of true positive, true false positive and false negative respectively. F1-score is defined as:

$$\text{F1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

Baseline Methods

1. **SVM**. As the simplest method, we use activity counts as feature vector for each user and train a SVM model for spammer prediction.
2. **SF+SVM**. We designed two social features, the number of friends, and the neighbor-distance defined in Eq. 9. We concatenate the two features into the activity count vector as the new feature vector and train the SVM model.
3. **MF+SVM**. We perform the matrix factorization on the user-activity matrix, and then use the latent factors of users as the feature vectors to build the SVM model.
4. **MFSR+SVM**. We first perform the matrix factorization with social regularization, and then use the latent factors of users as the feature vectors to build the SVM model. The difference between this method and our proposed method (**SMFSR**) is that this method does not incorporate the classification loss in the model.

We use LibSVM (Chang and Lin 2011) as our classifier for all baseline methods. We use linear kernel and set its tradeoff parameter $C = 1$ in all the results reported in this paper. We have also tried other values from 0.001 to 100.0 and found the result changes very little within this range.

Results

To do the comparison, we train all the methods on the training dataset and apply the models to the testing data, where we obtain the precision, recall and F1-score. We repeat the

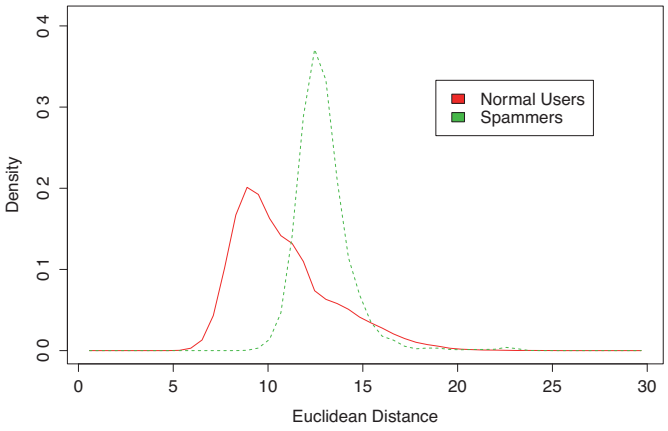

Figure 2: The distribution of Eq. 9, the Euclidean distances of activity vectors between a user u_i and the mean of his friends $N(u_i)$. The distance of normal users is generally smaller than that of spammers.

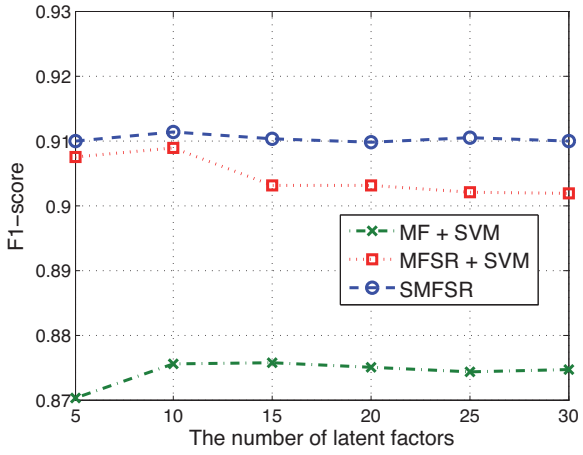

Figure 3: Varying the number of latent factors, K, in matrix factorization based models.

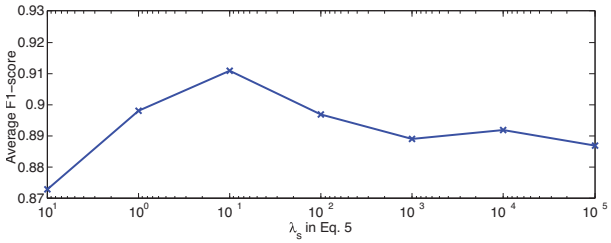

Figure 4: Varying the social regularization coefficient λ_s.

experiments 10 times and report the average performance of all methods in Table 3. The **MF+SVM** method performs better than **SVM** on raw activity counts, which shows the advantage of the matrix factorization as a method to get stable and compact representation. **MFSR+SVM** method improves all the three evaluation criteria over the other three baseline methods. The simple use of social information (**SF+SVM**) does not improve the precision and recall as much as using social information as regularization. We believe that adding more features extracted from social structures could further improve the precision and recall of **SF+SVM**, e.g. rather than using Euclidian distance in Eq. 9, we can use other distance metrics. However, this kind of method needs more human interference while the social regularized method is an automatic method to extract better features. Finally, our proposed method, **SMFSR**, performs better than **MFSR+SVM** because the social regularized latent factors are guided by the classification loss.

We also report the sensitivity of two important parameters, the number of latent factors K used in all MF based methods, and the tradeoff parameter for social regularization λ_s in **SMFSR**. As shown in Figure 3, the F1-score in all methods is best at small K values (10 or 15). **SMFSR** method does not decrease the performance when K is large, , while other MF based methods do not perform well when K is large. The supervised learning term in Eq. 5 makes the dimensionality reduction more stable than other MF based methods. Figure 4 shows the average F1-score when λ_s changes from 10^{-6} to 10. When λ_s is small, the performance is close to that of **MFSR+SVM**. However, when λ_s is relatively large, the optimization in Eq. 5 may be dominated by the social regularization term, therefore the supervised learning loss term is not properly optimized.

Discussion

Our recall is comparable or better than published results for spammer detection in Twitter and MySpace, e.g. in (Wang 2010; Lee, Caverlee, and Webb 2010). But our precision is slightly lower than them. This is because the binary classification in our experiment is a very skew classification (700 vs 30,000), while above work uses balanced classification for evaluation. We believe that our setting is more realistic since spammers are the minority group compared to majority normal users. Different from traditional email spam detection, where precision is extreme important, in social spammer detection, high recall is more desirable. Most users will prefer typing one reCAPTCHA (Von Ahn et al. 2008) to being annoyed by spammers repeatedly.

Conclusion and Future Work

In this paper, we have studied the problem of detecting spammers in a very large scale social network. Our proposed method unifies dimensionality reduction, social regularization and spammer classification into a single framework (**SMFSR**) using Collective Matrix Factorization (CMF). The experiments conducted on the Renren social network validate the effectiveness of our method. Besides gaining better performance over traditional methods, we have also analyzed statistics on the spammers structure in Renren social network, which gives us insight on the model design. In particular, we have found that social regularization – a spammer is different from its friends, is the most effective component in our method.

Table 3: Comparison of different methods: In all MF based methods, the number of latent factors K is set to 10.

Method	*Precision*	*Recall*	*F1-score*
SVM	0.781 ± 0.027	0.952 ± 0.030	0.857 ± 0.020
SF + SVM	0.810 ± 0.029	0.949 ± 0.021	0.874 ± 0.020
MF + SVM	0.807 ± 0.036	0.966 ± 0.038	0.879 ± 0.029
MFSR + SVM	0.839 ± 0.026	0.978 ± 0.027	0.903 ± 0.013
SMFSR	0.851 ± 0.032	0.980 ± 0.037	0.911 ± 0.031

We plan to extend our work in the following directions. Firstly, we wish to use other matrix factorization methods to explore additional hidden factors to represent users. In particular, recent advances in tensor factorization could be used to factorize the performer-activity-receiver interaction tensor, which contains richer information than the user-activity matrix used in this work. However, the great sparseness in such representation poses challenges. Secondly, we will integrate our method as a new component into the existing immune system in Renren. Thirdly, we are particularly interested in identifying normal users whose accounts are hijacked by spammers to send spam to his social friends.

Acknowledgment

We thank the support of Hong Kong RGC GRF projects 621010 and 621211.

References

Benevenuto, F.; Magno, G.; Rodrigues, T.; and Almeida, V. 2010. Detecting spammers on twitter. In *Anti-Abuse and Spam Conference (CEAS)*.

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

Brown, G.; Howe, T.; Ihbe, M.; Prakash, A.; and Borders, K. 2008. Social networks and context-aware spam. In *ACM Conference on Computer Supported Cooperative Work (CSCW)*, 403–412.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: A library for support vector machines. *ACM TIST* 2(3):27.

Duch, J., and Arenas, A. 2005. Community detection in complex networks using extremal optimization. *Physical Review E* 72(2):027104.

Gyöngyi, Z., and Garcia-Molina, H. 2005. Web spam taxonomy. In *Adversarial Information Retrieval on the Web (AIRWeb)* , 39–47.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 426–434.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. B. 2010. What is twitter, a social network or a news media? In *WWW*, 591–600.

Landauer, T.; McNamara, D.; Dennis, S.; and Kintsch, W. 2007. *Handbook of latent semantic analysis.* Lawrence Erlbaum Associates Publishers.

Lee, K.; Caverlee, J.; and Webb, S. 2010. Uncovering social spammers: social honeypots + machine learning. In *SIGIR*, 435–442.

Lee, K.; Eoff, B. D.; and Caverlee, J. 2011. Seven months with the devils: A long-term study of content polluters on twitter. In *The International AAAI Conference on Weblogs and Social Media (ICWSM)*, 185–192.

Li, F., and Hsieh, M.-H. 2006. An empirical study of clustering behavior of spammers and group-based anti-spam strategies. In *Anti-Abuse and Spam Conference (CEAS)*.

Liu, Z.; Shu, G.; Li, N.; and Lee, D. 2006. Defending against instant messaging worms. In *In Proceedings of IEEE GLOBECOM*, 1–6.

Ma, H.; Liu, C.; King, I.; and Lyu, M. R. 2011. Probabilistic factor models for web site recommendation. In *SIGIR*, 265–274.

Rendle, S. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST 2012)* 3(3).

Rennie, J. D. M. 2004. Smooth hinge classification. http://people.csail.mit.edu/jrennie/writing.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD*, 650–658.

Stein, T.; Chen, E.; and Mangla, K. 2011. Facebook immune system. In *Proceedings of the EuroSys Social Network Systems (SNS)*, 8:1–8:8.

Von Ahn, L.; Maurer, B.; McMillen, C.; Abraham, D.; and Blum, M. 2008. recaptcha: Human-based character recognition via web security measures. *Science* 321(5895):1465.

Wang, A. H. 2010. Don't follow me - spam detection in twitter. In *Intl. Conf. on Sec. and Crypto.*, 142–151.

Xu, Q.; Xiang, E.; Du, J.; Zhong, J.; and Yang, Q. 2012. Sms spam detection using content-less features. *IEEE Intelligent Systems* http://doi.ieeecomputersociety.org/10.1109/MIS.2012.3.

Yang, Z.; Wilson, C.; Wang, X.; Gao, T.; Zhao, B. Y.; and Dai, Y. 2011. Uncovering social network sybils in the wild. In *Internet Measurement Conference*, 259–268.

Zhu, S.; Yu, K.; Chi, Y.; and Gong, Y. 2007. Combining content and link for classification using matrix factorization. In *SIGIR*, 487–494.

Zinkevich, M.; Smola, A. J.; and Langford, J. 2009. Slow learners are fast. In *NIPS*, 2331–2339.