Catch the Black Sheep: Unified Framework for Shilling Attack Detection Based on Fraudulent Action Propagation

Yongfeng Zhang[†], Yunzhi Tan[†], Min Zhang[†], Yiqun Liu[†], Chua Tat-Seng[‡], Shaoping Ma[†]
[†]State Key Laboratory of Intelligent Technology and Systems
[†]Department of Computer Science, Tsinghua University, Beijing, 100084, China
[‡]School of Computing, National University of Singapore, 117417, Singapore
{yongfeng14,tyz13}@mails.thu.edu.cn, {z-m,yiqunliu,msp}@thu.edu.cn, dcscts@nus.edu.sg

Abstract

Many e-commerce systems allow users to express their opinions towards products through user reviews systems. The user generated reviews not only help other users to gain a more insightful view of the products, but also help online businesses to make targeted improvements on the products or services. Besides, they compose the key component of various personalized recommender systems. However, the existence of spam user accounts in the review systems introduce unfavourable disturbances into personalized recommendation by promoting or degrading targeted items intentionally through fraudulent reviews. Previous shilling attack detection algorithms usually deal with a specific kind of attacking strategy, and are exhausted to handle with the continuously emerging new cheating methods. In this work, we propose to conduct shilling attack detection for more informed recommendation by fraudulent action propagation on the reviews themselves, without caring about the specific underlying cheating strategy, which allows us a unified and flexible framework to detect the spam users.

1 Introduction

With the ability to help recommend items of potential interests to users, and thus to benefit both online users and businesses, Personalized Recommender Systems (PRS) [Ricci *et al.*, 2011] have become an essential part of various online applications, including e-commerce, social networks, video websites, and news portals. The widely adopted Collaborative Filtering (CF) [Su and Khoshgoftaar, 2009] approaches has the advantage of making recommendations with the wisdom of crowds, which contributed to the great success of personalized recommendation in practical systems.

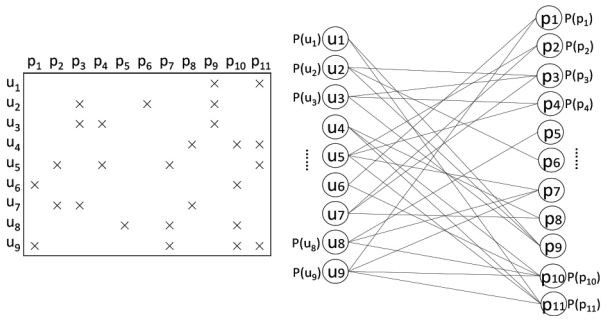
However, the CF-based approaches inherently relies on the historical behaviours of a user as well as those of the others for model estimation and personalized recommendation, which leaves space for attackers to affect the recommendation results received by normal users through attack profile injection into a system [Gunes *et al.*, 2012]. For example, an attacker who aims at promoting a targeted item may register a sufficient number of fake accounts in the system and rate relatively higher scores towards the targeted item, or relatively lower scores vise versa. According to the experimental observations in practical systems, a 3% fake profile attack would results in a prediction shift of around 1.5 points on a five-point scale, which poses a severe problem on the usability of recommender systems [Jannach *et al.*, 2010].

To reconstruct a pure land for recommendation algorithms, researchers have conducted various studies to scrutinize different shilling attack strategies, profile injection attack types, attack detection schemes, robust algorithms to overcome such attacks, and evaluate them with respect to accuracy, cost, benefit, and overall performance [Gunes *et al.*, 2012].

However, previous methods usually have to examine the details of an attack strategy so as to find ways out to extract the injected profiles out of millions of normal ones. As a result, each attack detection method can only tackle with a limited kind of attack strategies, and their application are usually restricted to specific recommendation algorithms. With the continuous emerging of new attack strategies, researchers have been exhausted to construct new targeted detection algorithms. The investigation on robust recommendation algorithms [Mehta *et al.*, 2007] partly alleviates the problem, but the explicit detection of attack profiles remains an important problem because personalized recommendation is not the only application of shilling attack detection techniques.

In this work, we propose to conduct shilling attack detection in an eye for an eye manner without being bothered by the details of the attack strategies. This is achieved by taking advantage of the inherent motivation and goal of the attackers, regardless of the specific attack method they used. To do so, we construct the user to item bipartite graph (Figure 1(b)) from the user-item rating matrix (Figure 1(a)) according to their relationship of equivalence introduced in [Zhang *et al.*, 2013a; 2013b], and develop a recursive bipartite propagation approach based on the bipartite graph, so as to estimate the probability of each user being a spam user account. Each edge from a user to an item in the graph indicates the degree that the user aims to manipulate the rating of the targeted item intentionally.

This framework is based on the observation that, the ultimate goal of a spam user account will eventually lead its way to affect the rating of the targeted items. As a result, we only have to consider the final ratings that a user cast on



(a) Rating Matrix (b) Bipartite Graph

Figure 1: An example of user-item rating dataset (matrix) and the corresponding bipartite graph. Each of the edges in bipartite graph corresponds to a rating in the user-item rating matrix, and the weight of each edge denotes the degree that the corresponding rating is a spam rating to promote or degrade the corresponding item/product. Recursive propagation is conducted on the graph to estimate the probability of each user being a spam user account.

the targeted items, without being burdened by the various and frequently changing attack strategies.

The main contributions of the paper can be summarized as follows:

- We propose to conduct shilling attack detection in review systems for personalized recommendation with a unified framework, based on propagating the degree that a user promotes or degrades a target item, without considering the specific attack strategy used.
- We develop an efficient algorithm for propagation and prove the convergence of the propagation process.
- With extensive experiments on real-world Amazon movie review datasets, we verify the effectiveness of our proposed unified framework.

The rest of the paper will be organized as follows: In Section 2, we review some of the related work, and present the problem formalizations in Section 3. We introduce our unified shilling attack detection framework, the algorithms, as well as the corresponding theoretical analysis in Section 4, and report the extensive experimental results on real-world datasets in Section 5. We finally conclude this work and point out some of the future research directions in Section 6.

2 Related work

Online businesses employ Collaborative Filtering (CF) [Su and Khoshgoftaar, 2009; Ricci *et al.*, 2011] algorithms to provide personalized recommendations to their customers so as to increase the sales and profits. And personalized recommendation also benefits online users to discover things of potential interests so as to get rid of the problem of information overwhelm.

However, although the personalized recommender systems are successful in e-commerce websites, they are vulnerable to shilling attacks [Gunes *et al.*, 2012]. On one hand, online businesses utilize CF algorithms to enhance their competitive edge over other businesses. On the other hand, malicious users and/or competing vendors might insert shilling reviews into the systems intentionally in such a way that they can affect the predicted ratings, so as to promote the items that they have an interest in, or to degrade the competing items.

Researchers proposed statistical analysis methods to detect anomalies in databases caused by suspicious ratings. Statistical anomaly detection [Bhaumik *et al.*, 2006] is one such approach relying on item average values, where outlier items are determined based on statistical process. Similarly, [Hurley *et al.*, 2009] utilized Neyman-Pearson statistical detection theory in which a binary hypothesis testing is performed to discriminate between genuine and attacker profiles. Besides, [Zhang *et al.*, 2006] propose a probabilistic approach using SVD-based data reduction method, where a compacted model of observed ratings (including real and biased ones) is generated by maximizing the log-likelihood of all ratings.

Supervised classification techniques are also utilized in attack detection schemes. [Burke *et al.*, 2006a; 2006b] utilize a classification approach for detecting malicious users based on attributes derived from each individual profile. These are called generic attributes in literature. Two of such derived attributes are Rating Deviation from Mean Agreement (RDMA) and Degree of Similarity with top neighbors (DegSim) proposed by [Chirita *et al.*, 2005] as a metric for detecting malicious profiles. In addition to RDMA based attributes, one more generic attribute is also proposed in [Burke *et al.*, 2006a], called Length Variance (LengthVar), which measures how much the length of a given user profile varies from the average length in the database, where length is the number of ratings of a user.

Besides, clustering approach in attack detection is utilized by [O'Mahony *et al.*, 2003] as neighbourhood selection to eliminate suspicious users. [Mehta, 2007] and [Mehta and Nejdl, 2009] introduced a PLSA-based clustering method to determine the spam users in recommendation generation process instead of using traditional nearest neighbor methods. [Bhaumik *et al.*, 2011] applied an unsupervised clustering algorithm based on several classification attributes for attack detection relying on statistical characteristics of dataset and accordingly produce user profiles relying on those attributes.

However, an important problem of current approaches is the extensive efforts required to construct different attack detection methods for different specific attack strategies, which is time consuming. In this work, we attempt to avoid this expensive process by introducing a unified framework for spam account detection, based on the final ratings of the users themselves, which relieves practical recommender systems from the extensive work of continuously fighting against emerging attack strategies.

3 Problem Formalization

Before presenting our framework, we introduce some definitions that will be adopted in this work.

Definition 1 *The rating dataset R is a set of triples $\langle u_i, p_j, r_{ij} \rangle$, where u_i is a user, p_j is a product/item, and r_{ij} is the rating given by user u_i towards product p_j. Let $\mathcal{U}$ and $\mathcal{P}$ denote the set of all users and products, respectively.*

Definition 2 *The rating dataset R can be equivalently represented by its corresponding bipartite graph $G = (\mathcal{U}, \mathcal{P}, \mathcal{E})$, where $\mathcal{U}$ and $\mathcal{P}$ are the same as above, and $\mathcal{E}$ is the set of edges $\langle u_i, p_j \rangle$. An edge in $\mathcal{G}$ connects a user and a product, which corresponds to a numerical rating made by the user towards the product in the rating dataset R.*

There exist two types of nodes in the graph, namely, the user nodes and the product/item nodes. Each of the user or item nodes is assigned a score $P(u_i)$ or $P(p_j)$, which denotes the probability of a user being a spam user, or the probability of a product being a spam product. Figure 1 gives an intuitional example of the rating dataset as a matrix, as well as its corresponding bipartite graph.

Definition 3 *The labeled seed spam user set $\mathcal{U}_s \subset \mathcal{U}$ is a small set of users that are manually detected and labeled as spam users. These users are used as the initial seed users that drive the propagation process on a bipartite graph. We will present this process and discuss the selection of seed users in the following parts.*

Given $G = (\mathcal{U}, \mathcal{P}, \mathcal{E})$ and a set of seed spam users $\mathcal{U}_s$, the goal of our framework is to estimate the spam probability $P(u_i)$ for each user $u_i \in \mathcal{U}$. After the propagation algorithm terminates, each user and product will receive a score that denotes its possibility of being spam.

4 The Framework

We propose a label propagation algorithm for the detection of spam user accounts. Specifically, we calculate the spam probability $P(u_i)$ of a user u_i by incorporating all of the spam probability of its adjacent products, as well as the connections (edges) between the user and the corresponding products. Similarly we calculate the spam probability $P(p_j)$ for each product p_j by considering all its adjacent users. This procedure is conducted back and forth and recursively. We will formally describe this process in this section.

4.1 Construction of the Bipartite Graph

To construct the bipartite graph from a rating matrix, we need to determine the weight of each edge connecting, for example, a user u_i and a product p_j in the bipartite graph G. In this work, we consider the rating count, user rating bias, product rating bias, as well as the global rating bias to estimate the edge weight w_{ij}:

$$w_{ij} = 1 + \left|\frac{r_{ij} - \bar{r}_i}{\bar{r}_i}\right| + \left|\frac{r_{ij} - \bar{r}_j}{\bar{r}_j}\right| + \left|\frac{r_{ij} - \bar{r}}{\bar{r}}\right| \quad (1)$$

In this formulation, the first additive component "1" is adopted to stand for the count that a user rates towards an item, where the intuition is that the spam user accounts usually tend to make a larger amount of ratings towards items compared with normal users, in order to manipulate the rating of the targeted items. By taking into consideration the number of ratings made during the propagation process on the bipartite graph, we are able to put more potentials of being spam on those highly active accounts.

The remaining additive components represent the relative biases of the rating against the average rating of the user, the average rating of the product, and the global average. The intuition here is that when a user attempts to rate a product far beyond the previous averaged ratings of himself, or the averaged rating of the product, or even the global averaged rating, we tend to treat such a rating as more "intentional" to manipulate the rating of a product [Gunes *et al.*, 2012; Su and Khoshgoftaar, 2009].

It is important to note that we can well incorporate other available information, such as user profiles and item contents, to construct more comprehensive edge weights for propagation, and our framework is flexible to integrate any weight definition. However, as we will show in the experiments, such a simple one as Eq.(1) based solely on ratings gives us competitive performance in spam account detection.

4.2 Transition Probability

In order to propagate the spam probability between users and products, we need to estimate the user to product transition probability $t_{u_i p_j}$, as well as the product to user transition probability $t_{p_j u_i}$ based on the original bipartite graph. To do so, we determine the transition probability for an edge in the graph by calculating the percentage of weights adopted by this edge against the total edge weight of a user/product, namely:

$$t_{u_i p_j} = \frac{w_{ij}}{\sum_{j':\langle u_i, p_{j'} \rangle \in \mathcal{E}} w_{ij'}} \quad (2)$$

and,

$$t_{p_j u_i} = \frac{w_{ij}}{\sum_{i':\langle u_{i'}, p_j \rangle \in \mathcal{E}} w_{i'j}} \quad (3)$$

and $t_{u_i p_j} = t_{p_j u_i} = 0$ if these is no edge connecting the user u_i and product p_j.

Suppose there exist m users and n products in the rating matrix (and also in the bipartite graph), we further define the user to product transition probability matrix $T_{up} = (t_{u_i p_j})_{m \times n}$, and symmetrically, the product to user transition probability matrix $T_{pu} = (t_{p_j u_i})_{n \times m}$. These matrices indicate the transition probabilities in a unified matrix format, and can be efficiently used for matrix-formed propagation process in the following.

4.3 Spam Probability Estimation

We consider the spam probability of each user and product, and construct the user spam probability vector $\mathbf{P}_u$, as well as the product spam probability vector $\mathbf{P}_p$ by organizing the spam probabilities as follows:

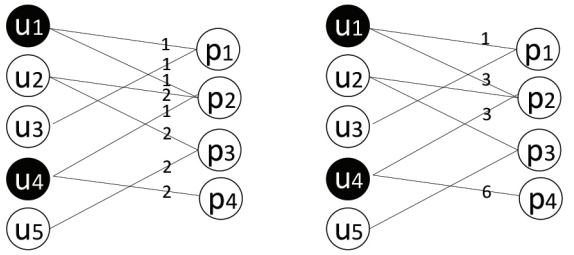
$$\mathbf{P}_u = [P(u_1), P(u_2), P(u_3), \cdots, P(u_m)]^T \quad (4)$$

$$\mathbf{P}_p = [P(p_1), P(p_2), P(p_3), \cdots, P(p_n)]^T \quad (5)$$

Based on the transition probability matrices, we further estimate the probability vectors in a propagation manner. In the i-th iteration, the following two rounds of propagation is conducted:

$$\mathbf{P}_p^i = T_{pu} \mathbf{P}_u^{i-1}, \text{ and } \mathbf{P}_u^i = T_{up} \mathbf{P}_p^i \quad (6)$$

It should be noted that the spam probability of the labeled users should be clamped before each round of iteration, which



(a) Example for the problem of positive feedback

(b) Example for the problem of reliability

Figure 2: Toy examples that exposit the problems of positive feedback and reliability, respectively, where the black user nodes are those spam user accounts in the seed set.

means that all of the users in the seed set $\mathcal{U}_s$ should be reassigned their initial spam probabilities (i.e., $P(u_i) = 1$ for $u_i \in \mathcal{U}_s$). In this way, the algorithm converges. We present the proof of convergence of this propagation process in the following subsections.

4.4 Polishing the Edge Weights

Propagation-based algorithms in practical systems usually suffer from the problem of positive feedback and the problem of the reliability of transition probabilities derived from trivially designed edge weights. In this section, we investigate the positive feedback problem and reliability problem in detail, and further polish the edge weights in order to alleviate the problems to achieve better shilling attack detection performances.

The Problem of Positive Feedback

Label propagation algorithms or random walks on bipartite graphs usually face positive feedback problems, and this is exampled in Figure 2(a). In this example, the user u_3 rated only the product p_1, and the spam probability will be $P(u_3) = 0.5$ for this user after the first round of iteration. Before the second iteration begins, we will reset $P(u_1) = 1$ because u_1 is a seed user account. It is easy to see that $P(u_3) = 0.75$ after the second iteration, and it finally converges to 1 as the iteration process proceeds on. The inherent reason here is that u_3 is a 1-degree node, which means that the probability score of u_3 will flow back to node p_1, which leads to the positive feedback problem, and this problem would magnify the noise in the bipartite graph and distort the final results.

Suppose that a user rated a product with suspicious ratings, while most of the other users made ordinary and consistent ratings towards the specific product, then all of the spam probabilities of the users will converge to 1 when the propagation procedure terminates. Even if the 1-degree nodes are removed from the results, the existence of the positive feedback problem will most likely misinterpret the explanations of the ratings of other users.

The Problem of Reliability

The spam probability of a specific node comes from its adjacent nodes during the propagation process, and the weight of each edge is related only to the specific rating from the corresponding user to the corresponding product. However, the correlation between a user and a product should be better determined by the rating distribution of both nodes jointly, and the current construction of bipartite graph fails to take this factor into account.

Consider the sampled portion of nodes and edges from a bipartite graph in Figure 2(b), where the edge from u_1 and u_4 are equally weighted for the product node p_2. We see that the spam probability of u_1 are mainly contributed from p_2, while the spam probability of u_4 is equally contributed by p_2 but with more contributed by p_4. In such a case, the spam probability contributed from u_1 is more convincing than that from u_4 for product p_2, although the corresponding edges are equally weighted, because the probability from u_4 is highly absorbed by the single node p_4 due to the relatively higher edge weight between u_4 and p_4.

Algorithm 1: PROPAGATE(G,$\mathcal{U}_s$)

Input: G: Bipartite graph with polished weights
$\mathcal{U}_s$: Selected seed spam user set
Output: $P(u_i)$ for all users in G

```
repeat
    for u_i ∈ U_s set P(u_i) = 1;
    P_p ← T_pu P_u;
    P_u ← T_up P_p;
until Convergence;
return P_u;
```

Based on the observations, we take the rating distributions of both the users and the products into consideration, and adopt the following definition of bidirectional edge weight, where for each edge $\langle u_i, p_j \rangle \in \mathcal{E}$, the new weight w'_{ij} is:

$$w'_{ij} = \frac{w_{ij}}{\left(\sum_{j':\langle u_i, p_{j'}\rangle \in \mathcal{E}} w_{ij'}\right)\left(\sum_{i':\langle u_{i'}, p_j\rangle \in \mathcal{E}} w_{i'j}\right)} \tag{7}$$

Intuitionally, this definition of bidirectional edge weight helps to magnify the influence from those nodes with a close relationship during the iterative process, and to minimize the effect of noisy nodes. We replace w_{ij} with this polished weight w'_{ij} in Eq.(2) and Eq.(3) to calculate the transition probability, and the algorithm of the propagation process for spam user account detection is described in Algorithm 1.

4.5 Proof of Convergence

We can see that both the user to product transition matrix T_{up} and the product to user transition matrix T_{pu} are right stochastic matrices, where each row of them consists of nonnegative real values summing to 1.

Consider the underlying user to user transition matrix $T_{uu} = T_{up}T_{pu}$ in each iteration, where each element $t_{ij} \in T_{uu}$ is $t_{ij} = \sum_k (t_{u_i p_k} t_{p_k u_j})$, and we have:

$$\begin{aligned} \sum_j t_{ij} &= \sum_j \sum_k (t_{u_i p_k} t_{p_k u_j}) = \sum_k \sum_j (t_{u_i p_k} t_{p_k u_j}) \\ &= \sum_k t_{u_i p_k} \sum_j t_{p_k u_j} = \sum_k t_{u_i p_k} = 1 \end{aligned} \tag{8}$$

As a result, the user to user transition matrix T_{uu} in each iteration is also a right stochastic matrix, and the user spam probability matrix $\mathbf{P}_u$ after an iteration process will be:

$$\mathbf{P}_u^i = T_{up}T_{pu}\mathbf{P}_u^{i-1} = T_{uu}\mathbf{P}_u^{i-1} \tag{9}$$

Without loss of generality, let the top l rows/columns in T_{uu} correspond to the selected seed users, and the remaining r rows/columns therein correspond to the unable users. Then the transition matrix T_{uu} can be split into a block matrix in the following manner:

$$T_{uu} = \begin{bmatrix} T_{ll} & T_{lr} \\ T_{rl} & T_{rr} \end{bmatrix} \tag{10}$$

Similarly, let the user spam probability vector $\mathbf{P}_u = [\mathbf{P}_l\ \mathbf{P}_r]^T$, where $\mathbf{P}_l$ are the top l elements of $\mathbf{P}_u$ that correspond to the seed users, and $\mathbf{P}_r$ are the remaining unlabelled users.

It can be noted that $\mathbf{P}_l$ never really changes because we reassign the spam probability of the seed users to one in the beginning of each iteration. It is proved by [Zhu and Ghahramani, 2002] that $\mathbf{P}_r$ converges to $(\mathbf{I}-T_{rr})^{-1}T_{rl}\mathbf{P}_l$ if T_{uu} is a right stochastic matrix. As a result, the user spam probability matrix $\mathbf{P}_u$ converges in the propagation process. Leveraging the same technique, we can prove that the product spam probability matrix $\mathbf{P}_p$ also converges.

5 Experiments

In this section, we conduct extensive experiments to evaluate the propagation-based framework for shilling attack user detection. We begin by introducing the experimental setup and evaluation measures, and then we report and analyze the experimental results of our framework.

5.1 Dataset Description

Previous work on shilling attack detection mostly experiment on simulated datasets. Typically, they insert simulated spam users with suspicious actions (e.g., large amount of ratings) into an original dataset, and attempt to detect these inserted users. We argue that this paradigm is infeasible for the evaluation of shilling attack detection, and we should better leverage complete real-world datasets for the detection of spam accounts.

To do so, we choose the Amazon Movie dataset[1] for experimentations. The Amazon Movie dataset consists of 8,567,727 reviews from 1,318,175 users towards 235,042 movies. To remove the cold users, we selected those users with at least 100 historical reviews, and this results in 297,073 reviews from 5,519 users towards 70,633 movies.

An important feature of the Amazon Movie dataset is that, each review is accompanied with the number of times that it is rated as helpfulness by users in Amazon, as well as the times that it is rated as unhelpfulness. Based on these ratings, we calculate the percentage of times that a user's reviews are rated as helpfulness by other online users, and empirically selected those users whose percentages of helpfulness ratings are less than 0.3 as candidate spam users accounts. These accounts are further labeled by three human annotators with careful observation into the user accounts in Amazon.com, and those users that are labeled as spam by more than two annotators are finally selected as spam accounts. The inter-annotator agreement is 78.2%. This gives us 583 spam user accounts, and we randomly select 300 users therefrom to construct the seed user set, and reserve the remaining 283 users as testing set.

[1] http://snap.stanford.edu/data/web-Amazon.html

5.2 Evaluation Protocol

We adopt the frequently used evaluation measures of Precision@k ($P@k$), Recall@k ($R@k$), and F-measure@k ($F_1@k$) for performance evaluation. Each user will be assigned a score of spam probability when the propagation process terminates, we thus rank the users in descending order of spam probability, and select the top-k users as the set of spam user accounts $\mathcal{U}_{spam}$. Suppose the set of spam users reserved for testing is $\mathcal{U}_{test}$, then the evaluation measures are calculated as follows:

$$P@k = \frac{\mathcal{U}_{spam} \cap \mathcal{U}_{test}}{\mathcal{U}_{spam}},\ R@k = \frac{\mathcal{U}_{spam} \cap \mathcal{U}_{test}}{\mathcal{U}_{test}} \tag{11}$$

and $F_1@k$ is thus calculated based on $P@k$ and $R@k$.

5.3 Spam User Account Detection

We evaluate the performance of spam user account detection under different choices of prediction length k, and the performance v.s. different choices of k are shown in Table 1.

k	100	200	300	400	500	600
$P@k$	**0.983**	0.952	0.927	0.718	0.588	0.497
$R@k$	0.261	0.522	0.726	0.749	0.768	0.778
$F_1@k$	0.413	0.674	**0.814**	0.733	0.666	0.606
k	700	800	900	1000	1500	2000
$P@k$	0.431	0.385	0.343	0.310	0.215	0.169
$R@k$	0.789	0.804	0.807	0.809	0.843	**0.880**
$F_1@k$	0.558	0.521	0.482	0.448	0.343	0.283

Table 1: Performance of spam user account detection under different choices of prediction length k, and the best performance on Precision, Recall, and F-measure are bolded.

The results show that we can achieve superior performance on the Precision@k when the prediction length k is relatively low. This means that our algorithm tends to give reliable spam accounts for those users that are predicted with high spam probabilities. Besides, the performance of Recall@k increases with the increase of prediction length, which is not surprising. Our algorithm achieves the best performance on F_1-measure of 0.814 with a prediction length of $k = 300$, where the corresponding precision $P@k = 0.927$ and recall $R@k = 0.726$, which is a well deserving performance in practical applications.

5.4 Comparison with Baseline Methods

We experiment with the most commonly used spam detection approaches of Support Vector Machine (SVM), Logistic Regression (LR), and Radial Basis Function Networks (RBFN)

[Orr, 1996] as baseline methods to conduct binary classification between normal users and spam users.

To do so, we first generate feature vectors to be adopted as inputs by the baseline methods. We conduct Matrix Factorization (MF) on the original user-item rating matrix by the widely adopted MF method of Singular Value Decomposition (SVD) [Abdi, 2007], with the number of latent factors set as 50. This process gives us a 50-dimensional feature vector for each user and each item in the dataset.

For each baseline method, we use the 50-dimensional user vector as features to classify spam users and normal ones. We conducted five-fold cross-validation for each experimental setting. As our testing set in the previous experiment of spam account detection includes 283 ($\sim$300) users, we adopt the average Precision@k for $k = 100, 200$, and 300 to calculate the comparable accuracy of our Fraudulent Action Propagation (FAP) method, and the results are shown in Table 2.

	SVM	LR	RBFN	FAP
Accuracy	92.98%	92.76%	93.02%	95.45%

Table 2: Accuracy of baseline methods, p-value< 0.01.

We see that our proposed propagation-based framework outperforms the commonly used shilling attack detection approaches. Besides, by comparing the performance on spam user detection of our framework (reported in the previous subsection) with the baseline methods (in this section), our propagation-based approach also gains superior performance in a large range of the selection of prediction length k.

5.5 Boost Existing Methods with Spam Probability

Except for conducting spam user account detection as a sole method, the spam probability produced by our framework can be further adopted as a discriminative feature to boost existing (baseline) spam detection approaches to achieve superior shilling attack detection performances.

Based on the spam probability estimated by our propagation framework, we further conduct experiments to boost the performance of LR, SVM, and RBFN methods, so as to verify the discriminative ability of our spam probability as features in the process of classification.

To do so, we append the corresponding spam probability of each user produced by our propagation process to the corresponding 50-dimensional feature vector given by the SVD method in the previous subsection, which results in a 51-dimensional feature vector representation. To keep the same model complexity, we append a trivial value of 1 to the original 50-dimensional feature vector for each user, and thus also extend its dimension to 51 for fair performance comparison. The results on classification accuracy are shown in Table 3, where "+SP" indicates the performance of integrating our Spam Probability as a feature.

	SVM	SVM+SP	LR	LR+SP	RBFN	RBFN+SP
Accuracy	93.06%	94.08%	92.95%	97.64%	93.06%	94.26%

Table 3: Classification accuracy by integrating spam probability as a feature, p-value< 0.01.

We can see that the spam probability given by our propagation framework helps to boost the performance of other spam detection approaches. This indicates that the spam probability given by our framework is a discriminate and promising feature to distinguish spam users from normal ones. This observation also grants the ability of our framework to generally help boost the performance in other related scenarios such as social networks or forums.

5.6 Algorithm Robustness Analysis

An important factor in our propagation-based framework is the choice of spam users as seeds, because the spam probability of each user is inherently propagated from the selected seed users. To investigate the robustness of our framework, we further experiment with the influence of the number of seed users adopted.

Of the 583 spam users in total, we randomly select N users as seed users, and reserve the remaining for evaluation, where N ranges from 30 to 360 with a step size of 30. For each choice of N, we evaluate the $P@k$, $R@k$, and $F_1@k$ with choices of $k = 100, 200, \cdots, 500$, and average the results to obtain an aggregated view. The results of aggregated precision, recall, and F-measure is shown in Figure 3.

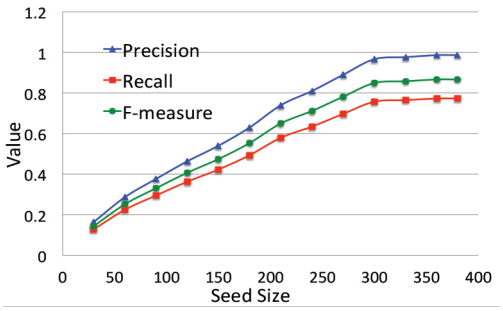


Figure 3: Precision, recall, and F-measure v.s. the number of seed users in the propagation process.

We see that in the beginning, the performance of our framework tends to rise along with the increase of the number of seed users selected, but it then tends to keep stable although we added more seed spam users into the framework. Besides, the performance on the measures of precision, recall, and F-measure exhibit similar trends.

This experimental result indicates that our framework does not require linearly more seed spam users in order to gain better performance. More precisely, when an adequately enough number of seed users (e.g., about 300 in this experiment) are incorporated to obtain satisfactory performance, there would be no need to add further more seed users to the propagation process, and this is the reason that we selected 300 spam users as seeds in the previous experiments.

This feature of our framework is favourable because it grants us the ability to achieve superior performance with a limited number of (perhaps manually selected and thus time consuming) seed spam users in practice.

6 Conclusions and Future Work

Shilling attack and spam user detection is crucial to personalized recommender systems to provide accurate recommendation results. In this paper, we investigated the problem

of shilling attack detection by proposing a unified framework based on spam probability propagation on user to product bipartite graphs, which alleviates researchers and practitioners from the exhausting work of fighting against continuously emerging new attack strategies. We further investigated the problems of positive feedback and reliability in propagation-based frameworks, and constructed meticulously designed edge weights for more accurate user to product relationship estimation in the bipartite graphs. Extensive experimental results on real-world Amazon movie review dataset verified the effectiveness of our framework, and the possibility to boost other shilling attack detection approaches with the spam probability features produced by our framework.

This work is a first attempt towards the approach of unified shilling attack detection in review systems, and there is much room for further improvements. Besides the numerical ratings that we used for edge weight estimation, we can well incorporate other information available in review systems to construct more discriminative edge weights for propagation, for example, the demographic information of users and the content information of products. We will also investigate automatic unified shilling attack detection in other online systems such as social networks or forums. The automatic propagation framework can even be leveraged in other online tasks like rumour detection in twitter and microblogging systems.

Acknowledgement

This work was supported by National Key Basic Research Program (2015CB358700), Natural Science Foundation (61472206) of China, Tsinghua University Initiative Scientific Research Program (2014Z21032), and Tsinghua-Samsung Joint Lab. Part of this work was conducted at the Tsinghua-NUS NExT Search Centre, which is supported by the Singapore National Research Foundation & Interactive Digital Media R&D Program Office, MDA under research grant (WBS:R-252-300-001-490).

References

[Abdi, 2007] H. Abdi. Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). *Ency. of Measu. and Stat.*, pages 907–912, 2007.

[Bhaumik *et al.*, 2006] Runa Bhaumik, Chad Williams, Bamshad Mobasher, and Robin Burke. Securing Collaborative Filtering Against Malicious Attacks Through Anomaly Detection. *The 4th workshop on intelligent techniques for web personalization (ITWP)*, 2006.

[Bhaumik *et al.*, 2011] Runa Bhaumik, Bamshad Mobasher, and Robin Burke. A Clustering Approach to Unsupervised Attack Detection in Collaborative Recommender Systems. *ICDM*, pages 181–187, 2011.

[Burke *et al.*, 2006a] Robin Burke, Bamshad Mobasher, Chad Williams, and Runa Bhaumik. Classification Features for Attack Detection in Collaborative Recommender Systems. *KDD*, pages 542–547, 2006.

[Burke *et al.*, 2006b] Robin Burke, Bamshad Mobasher, Chad Williams, and Runa Bhaumik. Detecting Profile Injection Attacks in Collaborative Recommender Systems. *E-Commerce Technology*, 2006.

[Chirita *et al.*, 2005] Paul-Alexandru Chirita, Wolfgang Nejdl, and Cristian Zamfir. Preventing shilling attacks in online recommender systems. *WIDM*, pages 67–74, 2005.

[Gunes *et al.*, 2012] Ihsan Gunes, Cihan Kaleli, Alper Bilge, and Huseyin Polat. Shilling Attacks Against Recommender Systems: A Comprehensive Survey. *Artificial Intelligence Review*, pages 1–33, 2012.

[Hurley *et al.*, 2009] Neil Hurley, Zunping Cheng, and Mi Zhang. Statistical Attack Detection. *RecSys*, pages 149–156, 2009.

[Jannach *et al.*, 2010] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. Recommender Systems: An Introduction. *Cambridge University Press*, 2010.

[Mehta and Nejdl, 2009] Bhaskar Mehta and Wolfgang Nejdl. Unsupervised Strategies for Shilling Detection and Robust Collaborative Filtering. *User Modeling and User-Adapted Interaction*, 19:65–97, 2009.

[Mehta *et al.*, 2007] Bhaskar Mehta, Thomas Hofmann, and Wolfgang Nejdl. Robust Collaborative Filtering. *RecSys*, pages 49–56, 2007.

[Mehta, 2007] Bhaskar Mehta. Unsupervised Shilling Detection for Collaborative Filtering. *AAAI*, pages 1402–1407, 2007.

[O'Mahony *et al.*, 2003] Michael P. O'Mahony, Neil J. Hurley, and Guenole C. M. Silvestre. Collaborative Filtering: Safe and Sound? *Foundations of Intelligent Systems*, 2871:506–510, 2003.

[Orr, 1996] Mark J. L. Orr. Introduction to Radial Basis Function Networks. *University of Edinburgh*, 1996.

[Ricci *et al.*, 2011] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to Recommender Systems Handbook. *Springer US*, pages 1–35, 2011.

[Su and Khoshgoftaar, 2009] Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 4, 2009.

[Zhang *et al.*, 2006] Sheng Zhang, Yi Ouyang, James Ford, and Fillia Makedon. Analysis of a Low-dimensional Linear Model under Recommendation Attacks. *SIGIR*, pages 517–524, 2006.

[Zhang *et al.*, 2013a] Yongfeng Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. Improve Collaborative Filtering Through Bordered Block Diagonal Form Matrices. *SIGIR*, pages 313–322, 2013.

[Zhang *et al.*, 2013b] Yongfeng Zhang, Min Zhang, Yiqun Liu, Shaoping Ma, and Shi Feng. Localized Matrix Factorization for Recommendation based on Matrix Block Diagonal Forms. *WWW*, pages 1511–1520, 2013.

[Zhu and Ghahramani, 2002] Xiaojin Zhu and Zoubin Ghahramani. Learning from Labeled and Unlabeled Data with Label Propagation. *Technical Report CMU-CALD-02-107*, 2002.